

STUDIENARBEIT

Thema: Vergleichende Untersuchungen zur
Repräsentanz des L_m -Wertes

Betreuer: Prof. Dr.-Ing. R. Azzam
Dipl.-Geol. D. Tondera

Bearbeiter: Jörg Meier

Abgabetermin: Juli 2001

Stichworte: Gebirgsklassifikation, Kernauswerteverfahren,
RQD-Index, Klüftigkeitsziffer k , Mittlere
Kernlänge k , RMR-System, Gebirgsfaktor C ,
 L_m -Wert, Rock Mass Quality Q und Q_{TBM} ,
Zerrüttungsgrad Z_G , SMR-System

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer, als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Freiberg, 27. Juni 2001

Vorwort

Die vorliegende Arbeit entstand im Rahmen einer Studienarbeit an der TU Bergakademie Freiberg. Für die Themenstellung und ständige fachliche Unterstützung und Förderung möchte ich mich ganz besonders bei meinen Betreuen, Prof. Dr.-Ing. R. Azzam und Dipl.-Geol. D. Tondera, bedanken.

Im Zusammenhang mit der Realisierung dieser Arbeit wurden zahlreiche Gespräche und Diskussionen mit einer Vielzahl von Professoren, Doktoren und Mitarbeitern der TU Bergakademie Freiberg geführt, die mir wertvolle Hinweise und Anregungen gaben.

Besonderen Dank gebührt auch dem Ingenieurbüro Dr. G. Meier für die Bereitstellung von Daten und für Konsultationen.

Inhaltsverzeichnis

1	Aufgabenstellung.....	6
2	Analyse verschiedener Auswerteverfahren und Kenntnisstandsanalyse.....	7
2.1	Übersicht und Klassifizierung.....	7
2.2	Darstellung der Auswerteverfahren.....	12
2.2.1	RQD-Index	15
2.2.2	Klüftigkeitsziffer k	17
2.2.3	Mittlere Kernlänge \bar{k}	18
2.2.4	RMR-System	19
2.2.5	Gebirgsfaktor C	23
2.2.6	L_m -Wert	24
2.2.7	Rock Mass Quality Q und Q_{TBM}	25
2.2.8	Zerrüttungsgrad Z_G	30
2.2.9	SMR-System	31
2.3	Vergleich und Diskussion der Auswerteverfahren.....	33
2.3.1	Theoretische Betrachtungen ausgewählter Verfahren.....	35
2.3.2	Darstellung der Datenbasis für den Vergleich der Auswerteverfahren.....	43
2.3.3	Gegenüberstellung der berechneten Werte.....	44
3	Versuch einer Gebirgsklassifikation nach dem L_m -Wert anhand von ausgewählten Anwendungsbeispielen.....	47
3.1	Gegenüberstellung des L_m -Wertes mit anderen Klassifikationsverfahren.....	47
3.1.1	Vergleich mit dem RQD-Index	48
3.1.2	Vergleich mit dem Zerrüttungsgrad Z_G	52
3.2	Vergleich der Klassen	53
3.3	Substitution des RQD-Indexes mit dem L_m -Wert.....	55
4	EDV-gerechte Aufarbeitung der Auswerteverfahren und deren Darstellung	57
4.1	Systemvoraussetzungen.....	57
4.2	Beschreibung des Programms „Gebirgsklassifikation“	57
4.2.1	Installation des Programms	58
4.2.2	Bedienoberfläche.....	58
4.2.3	Erstellen, Öffnen und Speichern von Dokumenten.....	59
4.2.4	Erstellen, Bearbeiten und Löschen von Bohrungen	60

4.2.5	Eingeben und Bearbeiten der Bohrkerndaten.....	61
4.2.6	Ausgabe der Ergebnisse	63
4.2.7	Beschreibung der zugrunde liegenden Datenbank	64
4.3	Beschreibung des ActiveX-Projektes „RockClassX“	66
4.3.1	Erläuterung des Konzeptes	66
4.3.2	Installation der Komponente	67
4.3.3	Einbinden der ActiveX-DLL in Anwendungsprogramme am Beispiel von MS Excel	67
4.3.4	Beschreibung der Schnittstelle clsClassification.....	70
5	Zusammenfassung und Ausblick.....	72
6	Verwendete Variablen.....	73
7	Literaturverzeichnis.....	74
8	Anlagenverzeichnis	77
9	Abbildungs-, Tabellen und Gleichungsverzeichnis.....	78
10	Stichwortverzeichnis	81
Anlagen	I bis XIII

1 Aufgabenstellung

Die geotechnische Beschreibung und Bewertung insbesondere von oberflächennahen Felsbereichen bezüglich der Beschaffenheit und Standfestigkeit ist bei vielen Baumaßnahmen der unterschiedlichsten Art ein offenes Problem. Mit Hilfe von Kernbohrungen werden zwar die geologischen Verhältnisse geklärt, aber repräsentative ingenieurgeologische Parameter damit nicht oder nur unzureichend bestimmt. Verschiedene Kernauswerteverfahren und komplexe Analysen verweisen hierbei auf vielversprechende Lösungsansätze.

Da bis heute kein kostengünstiges, zerstörungsfreies und zuverlässiges Erkundungsverfahren existiert, mit dem die Untergrundverhältnisse zweifelsfrei geklärt und geotechnisch bewertet werden können, sind Kernbohrungen nach wie vor die effektivste direkte Erkundungsmethode. Aufgrund des relativ hohen technischen Aufwandes sind Kernbohrungen kostenintensiv. Deshalb wird eine möglichst hohe Informationsausbeute aus dem Kernmaterial angestrebt. Zudem bestehen nach der Bohrkernaufnahme noch die Möglichkeiten der Entnahme von Laborproben zur Bestimmung weiterer geotechnischer Parameter und der Durchführung geomechanischer und geophysikalischer Versuche und Videobefahrungen am bestehenden Bohrloch.

Eine der ersten Gebirgsklassifikationen wurde 1946 von TERZAGHI [40] veröffentlicht. Dieses rein qualitative Verfahren bewertet das Gebirge nach der Belastung und dem eingebrachten Stützausbau bei Tunnelvortrieb. Das erste Verfahren auf quantitativer Basis wurde 1958 von LAUFFER [26] für den Stollenausbau im Fels vorgeschlagen. Im Rahmen der umfangreichen Tunnel- und Talsperrenbaumaßnahmen kam es zur Entwicklung weiterer allgemeiner Verfahren, wie zum Beispiel der Rock Mass Quality Q oder des RMR-Systems. In der letzten Zeit gehen die Tendenzen jedoch zu „projektbezogenen Klassifizierungen“. Die Zielstellung ist dabei stets die möglichst genaue Vorhersage und Kontrolle des Aufwandes für die Erstellung und die nötigen Maßnahmen für die Sicherung sowie der damit verbundenen Kosten.

In der vorliegenden Studienarbeit sollen die wesentlichen Kernauswerteverfahren analysiert, verglichen und dem neu entwickelten L_m -Wert gegenübergestellt werden. Außerdem soll der Versuch einer Gebirgsklassifikation anhand des L_m -Wertes unternommen werden und die Auswerteverfahren EDV-gerecht aufgearbeitet werden.

2 Analyse verschiedener Auswerteverfahren und Kenntnisstandsanalyse

2.1 Übersicht und Klassifizierung

Heute werden Kernauswerteverfahren in vielen Bereichen eingesetzt. Sowohl im über-tägigen als auch im untertägigen Bereich wird mit Hilfe von gewonnenen Bohrkernen versucht, komplexe Informationen über die Gebirgseigenschaften zu erhalten. Die Abbildung 1 gibt eine Übersicht über bekannte Einsatzgebiete und nennt Beispiele der in diesen Bereichen eingesetzten Auswerteverfahren.

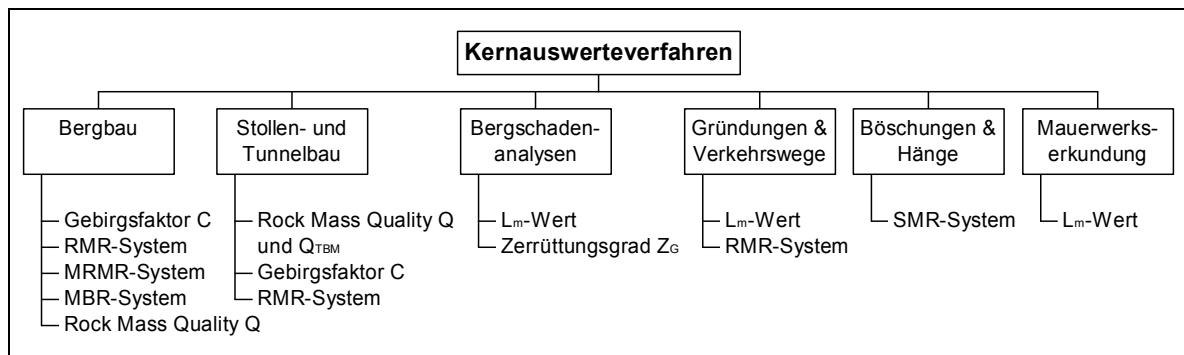


Abbildung 1: Einsatzgebiete von Kernauswerteverfahren

In der Regel teilen diese Auswerteverfahren das anstehende Gebirge in Bereiche ein. Ziel dieser Klassifizierung ist es, Bereiche im Gebirge zu identifizieren, die sich bereichsbezogen auf ein Merkmal bzw. eine Merkmalsgruppe annähernd homogen verhalten. Nach JOHN [20] sind folgende Faktoren für die Beschreibung des Gebirgsverhaltens bestimmend: die Baumethode, die Primärspannungsverhältnisse, der Bauvorgang und die Orientierung des Bauwerkes zum Trennflächengefüge. Aufbauend auf den durch die Kernauswerteverfahren gewonnenen Bereichen werden für einige Auswertesysteme die für die Vorrichtung und Erhaltung notwendigen Maßnahmen (z.B. Art und Stärke des nötigen Ausbaus) vorgeschlagen.

Auswerteverfahren lassen sich nach verschiedenen Gesichtspunkten klassifizieren. Eine Einteilung kann zum einen nach qualitativen bzw. quantitativen Verfahren (Abbildung 2, Seite 8) und zum anderen nach technologiebezogenen bzw. gebirgsbezogenen Verfahren (Abbildung 3, Seite 9) erfolgen.

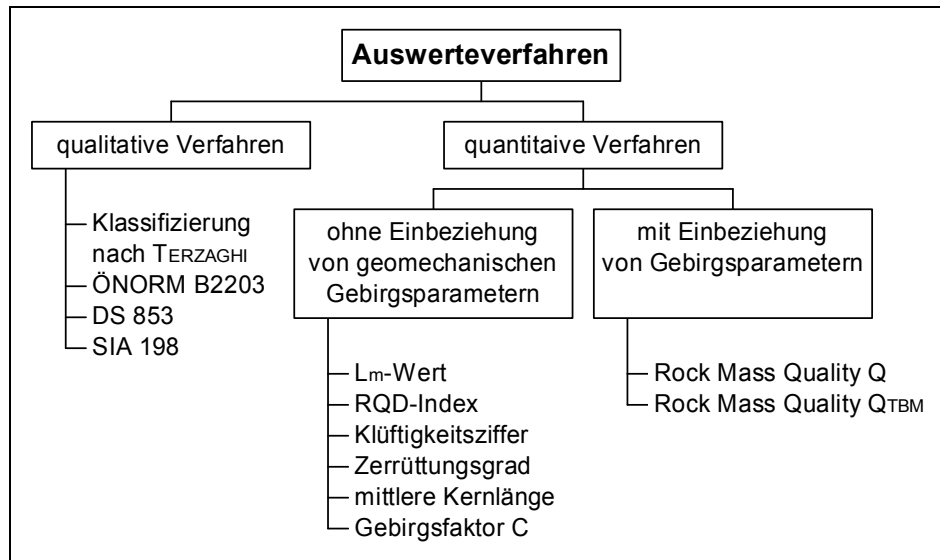


Abbildung 2: Klassifikation der Auswerteverfahren nach qualitativen und quantitativen Aspekten

Als qualitative Verfahren seien hier nur stellvertretend die österreichische ÖNORM B2203, DS 853 und die SIA 198 genannt. Im Rahmen dieser Arbeit sollen vor allem die quantitativen Verfahren untersucht werden. Qualitative Verfahren versuchen, unabhängig von den jeweiligen geologischen Gegebenheiten das Verhalten des Gebirges als Grundlage einer Einteilung in Bereiche zu nehmen. Dabei werden die Art des Ausbruches, das Gebirgsverhalten, mögliche Angriffstiefen, geometrische Ausbildungen, Art der Sicherungsarbeiten, der Zeitpunkt ihrer Durchführung und andere betriebliche Faktoren in die Auswertung und Klassenfindung mit einbezogen. Verfahren wie die ÖNORM B2203, DS 853 und die SIA 198 sind nach diesem Prinzip als reine Abrechnungsgrundlage entstanden [14].

Die quantitativen Verfahren lassen sich ihrerseits wiederum in Verfahren mit und ohne Berücksichtigung von geomechanischen Gebirgsparametern aufteilen.

Technologiebezogene Verfahren, wie die von BARTON im Jahr 2000 entwickelte „Rock Mass Quality Q_{TBM} “, wurden für eine bestimmte Technologie optimiert und lassen sich nicht ohne weiteres auf andere Verfahren übertragen. Bei der Entwicklung neuer Technologien oder einer signifikanten Weiterentwicklung muss wieder ein bestimmtes Maß an Grundlagenforschung betrieben werden, um zu determinieren, ob ein technologiebezogenes Verfahren erweitert werden kann und es dann gegebenenfalls anzupassen.

Gebirgsbezogene Verfahren, wie z.B. der RQD-Index oder der L_m -Wert, unterliegen solchen Einschränkungen nicht. In ihnen sind keine technologiebezogenen Parameter enthalten. Es lassen sich allerdings oft aus ihren Ergebnissen Hinweise auf die einzu-

setzenden Bauverfahren ableiten. Die gebirgsbezogenen Methoden bilden in der Regel die Grundlage für technologiebezogene Verfahren und gehen in diese als Parameter ein (wie z.B. der RQD-Index in die Rock Mass Quality Q_{TBM}) (Abbildung 3).

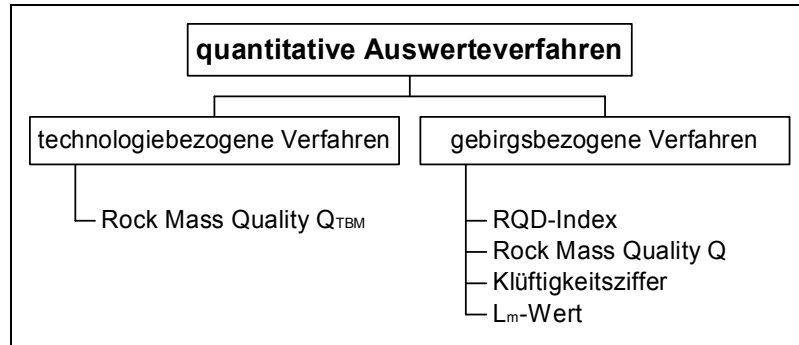


Abbildung 3: Klassifikation nach technologie- und gebirgsbezogenen Aspekten

In der Tabelle 1 wurden alle hier behandelten quantitativen Gebirgsklassifikationsverfahren in chronologischer Reihenfolge (Jahr der ersten Publikation) aufgelistet. Eine detaillierte Beschreibung der einzelnen Verfahren erfolgt in Abschnitt 2.2.

Tabelle 1: Zusammenfassung der Gebirgsklassifikationsverfahren

Parameter	Definition	Eingangswerte	Bewertung
RQD-Index (DEERE 1963)	(Seite 15) $RQD = \frac{\sum l}{S} * 100\%$	$\sum l$ Summe der Kernstücklängen mit $l > 10$ cm S Länge der Kernstrecke	RQD-Index [%] Felsqualität 0 - 25 sehr gering 25 - 50 gering 50 - 75 mittel 75 - 90 gut 90 - 100 ausgezeichnet
Klüftigkeitsziffer (MÜLLER 1963, MÖBUS 1982)	(Seite 17) $k = \frac{n}{L}$ [m ⁻¹]	n Anzahl anschnittskorrigierter Kluftschnitte entlang einer Messstrecke L Länge der Kernstrecke parallel zur Bohrloch-längsachse (z.B. Kernstreckenlänge in [m])	k [m ⁻¹] Klüftigkeit 0,1 - 1 weitmaschig 1 - 10 mitteldicht 10 - 100 engständig 100 - 1000 zerdrückt bis mylonisiert
mittlere Kernlänge \bar{k} (HANSÁGI 1967)	(Seite 18) $\bar{k} = \frac{k_1 * x_1 + k_2 * x_2 + \dots + k_n * x_n}{\sum x_i}$ [cm]	x_n Länge der Kernbruchstücke mit $l > 10$ cm k_n Mittellängen der Klassen (15, 25, 35 cm usw.)	
RMR-System (BIENIAWSKI 1973)	(Seite 19) $RMR = R_{q_n} + R_{RQD} + R_a + R_b + R_c + R_d$	R_{q_n} Wertung für einaxiale Gesteinsdruckfestigkeit R_{RQD} Wertung für RQD-Index nach DEERE (vgl. Seite 13) R_a Wertung für Kluftabstand R_b Wertung für Kluftbeschaffenheit R_c Wertung für Raumstellung der Klüfte R_d Wertung für Zutritt von Grundwasser	RMR Klasse 81 - 100 I sehr gut 61 - 80 II gut 41 - 60 III normal 21 - 40 IV schlecht 0 - 20 V sehr schlecht
Gebirgsfaktor C (HANSÁGI 1974)	(Seite 23) $C = \frac{1}{2 * S} * (p * H + \frac{K}{n})$	S Länge der Berechnungseinheit in [m] p Anzahl zylindrischer Probenkörper, die aus intakten Bohrkernen herausgesägt werden können H Höhe der zylindrischen Probenkörper in [m] K Summe der Kernstücklängen in der Berechnungseinheit in [m] n Anzahl der Kernstücke in der Berechnungseinheit	C-Faktor Felsqualität 0,00 - 0,15 sehr schwach 0,15 - 0,30 schwach 0,30 - 0,45 mittel 0,45 - 0,65 gut 0,65 - 1,00 ausgezeichnet
L_m-Wert (MEIER 1974)	(Seite 24) $L_m = \frac{k_1 * \sum x_1 + k_2 * \sum x_2 + \dots}{S}$ [cm]	$\sum x_i$ Summe der Kernstücke der Klassen k_i Mittellängen der Klassen, d.h. 2,5; 7,5; 12,5 cm usw. S Länge der Berechnungseinheit Der L _m -Wert stellt das gewogene arithmetische Mittel der Kernstückklassen dar. Er ergibt sich aus 5-cm-Klassen der einzelnen Kernstücklängen und bezieht sich auf eine Auswerteeinheit von 1 m Länge. Die 5-cm-Klasse enthält die Differenz zwischen der Summe aller Kernstücke >5 cm und der Auswerteeinheit.	L _m [cm] Felsqualität < 5,0 sehr gering 5,1 - 10,0 gering - mittel 10,1 - 15,0 mittel - gut 15,1 < gut - sehr gut
Rock Mass Quality Q (BARTON, LIEN, LUND 1974)	(Seite 25) $Q = \frac{RQD}{J_n} * \frac{J_r}{J_a} * \frac{J_w}{SRF}$	RQD RQD-Index nach DEERE J_n Anzahl der Klufscharen J_r Rauigkeit für die schwächste Spaltebene J_a Umwandlungsgrad (Charakter der Klüfte oder Füllung längs der schwächsten Spaltebene) J_w Kluftwasserführung SRF Spannungsniveau vgl. Anlage I: Eingangswerte Rock Mass Quality Q	Q Klasse 400 - 1000 1 ausgezeichnet 100 - 400 2 extrem gut 40 - 100 3 sehr gut 10 - 40 4 gut 4 - 10 5 mittel 1 - 4 6 gering 0,1 - 1 7 sehr gering 0,01 - 0,1 8 extrem gering 0,001 - 0,01 9 äußerst gering

Tabelle 1: Zusammenfassung der Gebirgsklassifikationsverfahren (Fortsetzung)

Parameter	Definition	Eingangswerte	Bewertung												
Zerrüttungsgrad Z_G (MOLEK 1983, 1984)	(Seite 30)	f Abminderungsfaktor durch nicht einmessbare Kernbrocken und Kernverluste bei fehlenden Kernverlusten und intakten, einmessbaren Kernstücken wird $f=1,0$ $f = \frac{K_S - (K_V + K_B)}{K_S}$ K_S Kernstrecke in [m] K_V Kernverlust in [m] K_B Länge nicht einmessbarer Kernbrocken (Schutt) in [m] $\sum l$ Summe der einmessbaren Kernstücke ($l \geq 1$ cm) je Berechnungslänge in [m] S Länge der Berechnungseinheit in [m] n Anzahl der einmessbaren Kernstücke je Berechnungslänge $Z_G = f * \frac{\sum l}{n * S}$	<table border="1"> <thead> <tr> <th>Z_G</th> <th>Zerrüttungsgrad</th> </tr> </thead> <tbody> <tr> <td>0,08 - 1,00</td> <td>schwach</td> </tr> <tr> <td>0,05 - 0,08</td> <td>mäßig</td> </tr> <tr> <td>0,02 - 0,05</td> <td>stark</td> </tr> <tr> <td>0,00 - 0,02</td> <td>sehr stark</td> </tr> </tbody> </table>	Z_G	Zerrüttungsgrad	0,08 - 1,00	schwach	0,05 - 0,08	mäßig	0,02 - 0,05	stark	0,00 - 0,02	sehr stark		
Z_G	Zerrüttungsgrad														
0,08 - 1,00	schwach														
0,05 - 0,08	mäßig														
0,02 - 0,05	stark														
0,00 - 0,02	sehr stark														
SMR-System (ROMANA 1985)	(Seite 31)	RMR RMR nach BIENIAWSKI, 1979 F_1 Winkel zwischen Hang und Kluft $F_1 = (1 - \sin \alpha)^2$ α Verschnittungswinkel zwischen Hangfläche und Kluft F_2 Einfallwinkel für den ebenen Versagensfall $F_2 = \begin{cases} 0,15 & \text{für } \beta_j \leq 21^\circ \\ \tan^2 \beta_j & \text{für } 21^\circ < \beta_j < 45^\circ \\ 1,00 & \text{für } \beta_j \geq 45^\circ \end{cases}$ β_j Kufteinfallwinkel F_3 Relation zwischen Hangoberfläche und dem Klufteneinfällen sehr günstig ± 0 günstig - 6 gut - 25 ungünstig - 50 sehr ungünstig - 60 F_4 Parameter für die Vorrichtungsmethode natürlicher Hang + 15 berauben + 10 schonendes Sprengen + 8 Sprengen oder mechan. Vorrichtung ± 0 mangelhaftes Sprengen - 8 $SMR = RMR + (F_1 * F_2 * F_3) + F_4$	<table border="1"> <thead> <tr> <th>SMR</th> <th>Klasse</th> </tr> </thead> <tbody> <tr> <td>81 - 100</td> <td>I sehr gut</td> </tr> <tr> <td>61 - 80</td> <td>II gut</td> </tr> <tr> <td>41 - 60</td> <td>III normal</td> </tr> <tr> <td>21 - 40</td> <td>IV schlecht</td> </tr> <tr> <td>0 - 20</td> <td>V sehr schlecht</td> </tr> </tbody> </table>	SMR	Klasse	81 - 100	I sehr gut	61 - 80	II gut	41 - 60	III normal	21 - 40	IV schlecht	0 - 20	V sehr schlecht
SMR	Klasse														
81 - 100	I sehr gut														
61 - 80	II gut														
41 - 60	III normal														
21 - 40	IV schlecht														
0 - 20	V sehr schlecht														
Rock Mass Quality Q_{TBM} (BARTON 2000)	(Seite 29)	Die Eingangswerte J_n , J_r , J_a , J_w und SRF wurden unverändert übernommen. J_r und J_a sollten sich auf die Klüfte beziehen, die das Bohren am stärksten begünstigen oder behindern. F Durchschnittliche Schneidmenge (in [tnf], „tons fealty“ - Tonnen fester Stoff) in der gleichen Zone, genormt auf 20 tnf σ geschätzte Gesteinsfestigkeit (in [MPa]) in der betrachteten Zone CLI cutter life index (z.B. 4 für Quarzit, 90 für Kalkstein) aus Johannessen et al. NTH, 1994.) q Quarzgehalt in Prozent σ_θ induzierte biaxiale Spannung in der betrachteten Zone, genormt auf eine Tiefe von ca. 100 m $Q_{TBM} = Q * \frac{\sigma}{F^{10} / 20^9} * \frac{20}{CLI} * \frac{q}{20} * \frac{\sigma_\theta}{5}$ $Q = \frac{RQD_0 * J_r * J_w}{J_n * J_a * SRF}$													

2.2 Darstellung der Auswerteverfahren

Jedes Auswerteverfahren ist nur so gut wie seine Eingangsparameter. Bei dem Bohren ist ein möglichst vollständiger Kerngewinn anzustreben. Werden bei der Aufnahme der verschiedenen Parameter Werte übersehen, weggelassen oder falsch ermittelt, so können diese Messfehler von keinem noch so guten Verfahren ausgeglichen werden. Gleiches gilt auch für die Bohrmannschaft, von deren Geschick der Zustand des Bohrkernes stark abhängt. Je schonender gebohrt wird, desto repräsentativer sind die Kennwerte für das untersuchte Gebirge.

Unabhängig von diesen eher zufälligen Fehlern sind systembedingte Einflussfaktoren und Fehler zu beachten. Die Qualität des erhaltenen Bohrkernes ist sehr stark von dem Kernrohrdurchmesser und von dem verwendeten Rohrtyp (z.B. Einfach- oder Doppelkernrohr, vgl. PRINZ [36]) abhängig. Die Wahl eines falschen Kernrohrtyps oder eines zu kleinen Durchmessers kann den Bohrkern zu stark belasten und ihn übermäßig deformieren oder gar den Verlust des Kernes zur Folge haben („Zerbohren des Kernes“), obwohl das anstehende Gebirge relativ gute geomechanische Eigenschaften besitzt. Aus diesem Ergebnis würde das Gebirge schlechter eingestuft werden, als es in der Realität der Fall ist [36]. So wurden z.B. von JOHN [20] unterschiedliche Bohrverfahren und deren Auswirkung auf die erhaltenen Bruchstücke gegenübergestellt. Innerhalb eines Projektes oder bei dem direkten Vergleich von verschiedenen Projekten sollten gleiche Kernrohrdurchmesser Verwendung finden, da dieser die Qualität des Kernes beeinflusst.

Die Wahl des optimalen Kernrohrtyps, Durchmessers und Bohrverfahrens kann nicht pauschal getroffen werden, da hier neben technischen auch wirtschaftliche Faktoren Beachtung finden müssen. Die Festlegung sollte unter Beachtung der Gegebenheiten und der Zielstellung von dem zuständigen Bearbeiter getroffen werden. Pauschal werden von verschiedenen Quellen Mindestdurchmesser vorgeschlagen. So werden z.B. von der International Society for Rock Mechanics (ISRM) für die Ermittlung von RQD-Werten Doppelkernrohre mit einem Durchmesser von mindestens 54,7 mm empfohlen [6] und von PRINZ [36] ein Enddurchmesser des Kernes von über 10 cm vorgeschlagen.

Einen ähnlich starken Einfluss auf die Güte des Kernmaterials besitzt die Vorbelastung des zu untersuchenden Gebirgsbereiches. Kernausswertungen in einem z.B. durch Sprengen stark vorbelasteten Gebirge sollten nach Möglichkeit vermieden werden (z.B. Untersuchung vor dem Sprengen) oder zumindest unter diesem Gesichtspunkt kritisch bewertet werden [35].

Die erbohrten Kerne sollten direkt nach dem Bohren aufgenommen werden, da sich verschiedene Gesteine unter den veränderten Umweltbedingungen (Änderung des Spannungszustandes, Wasser- und Sauerstoffangebotes usw.) verändern können. Tonige Gesteine neigen z.B. zum Quellen bzw. Schrumpfen.

Die Aufnahme des Bohrkerns sollte die Petrographie, Trennflächen mit ihren Eigenschaften, ggf. Füllung, Wasserführung, Verwitterung und evtl. Raumlage einschließen. Bei der Aufnahme der einzelnen Kernbruchstücke sollten alle Bruchstücke ab ca. 1 - 2 cm Länge aufgenommen werden, auch wenn das angestrebte Auswerteverfahren nur größere Bruchstücklängen berücksichtigt (z.B. RQD-Index: > 10 cm; L_m -Wert: ≥ 5 cm). Sollte während der Bearbeitung ein zusätzliches Auswerteverfahren genutzt werden, fehlen so keine Eingangsdaten. Wie die Wahl des Kernrohrdurchmessers sollte auch die Festlegung der Untergrenze der kleinsten aufzunehmenden Kernbruchstücke vom zuständigen Bearbeiter vor Ort festgelegt werden und für ein Objekt einheitlich sein.

In der Abbildung 4 ist ein Aufnahmebeispiel dargestellt. Es sollten der Reihe nach mit größer werdender Teufe immer die dominanteren Bruchstücke, zerdrückten Bereiche bzw. Kernverluste/Hohlräume von Mitte zu Mitte der Begrenzungen gemessen werden. Bei dem Auftreten von technisch bedingten Brüchen, werden diese bei der Bruchstückaufnahme ausgelassen. Achsenparallel zur Bohrung durchtrennte Bruchstücke werden ebenfalls nicht in die Bruchstückaufnahme aufgenommen, bei diesen Fällen wird wieder der dominanter Teil des Kerns gemessen¹.

Um den Informationsgehalt der Kernaufnahmen zu erhöhen wird vorgeschlagen, hinter jeden Längenwert der einzelnen Bruchstücke bzw. Bereiche einen zusätzlichen Parameter anzuhängen, der den Zustand des Kernes bzw. sein Fehlen wiedergibt. Dazu wurde aus der DIN 4022 Teil 2 [12] die Tabelle des Abschnittes 4.2.7 um das Kurzzeichen „E“ erweitert, um die Angabe von Kernverlusten bzw. von Hohlräumen zu ermöglichen (vgl. Tabelle 2). Mit diesem „Erhaltungsgrad“ können nicht nur intakte Kernbruchstücke, sondern auch Störungszonen, mylonitisierte Bereiche, Hohlräume und Kernverluste quantitativ aufgenommen werden und die Kernbruchstücke differenzierter wiedergegeben werden.

¹ nach mündlicher Information von Dr. G. MEIER

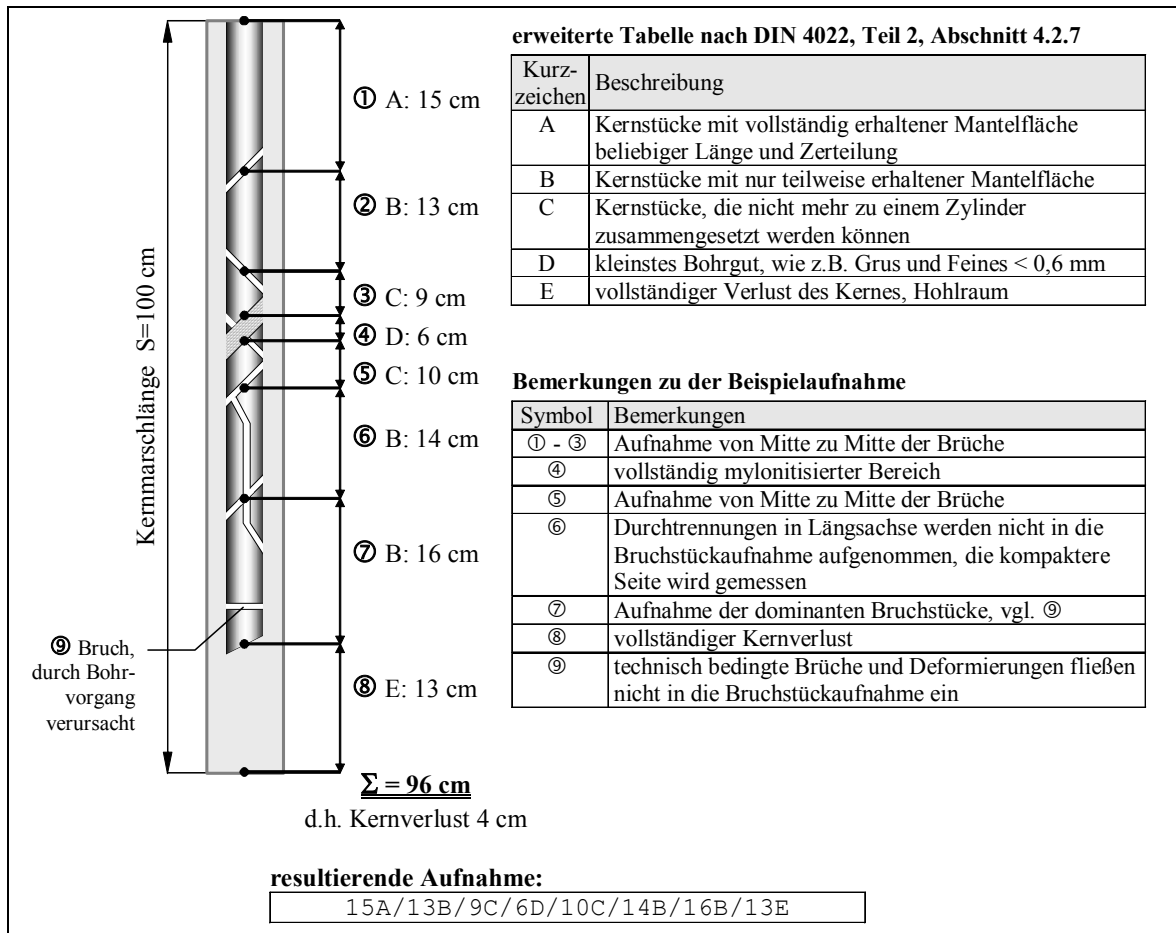


Abbildung 4: Beispielaufnahme eines Bohrkernes

Tabelle 2: erweiterte Tabelle nach DIN 4022 Teil 2 [12]

Kurzzeichen	Beschreibung	
A	Kernstücke mit vollständig erhaltener Mantelfläche beliebiger Länge und Zerteilung	
B	Kernstücke mit nur teilweise erhaltener Mantelfläche	
C	Kernstücke, die nicht mehr zu einem Zylinder zusammengesetzt werden können	
D	kleinstes Bohrgut, wie z.B. Grus und Feines < 0,6 mm	
E	vollständiger Verlust des Kernes, Hohlraum	

Unabhängig von der schriftlichen Aufnahme der Bohrkerndaten sollte von jedem Kernmarsch mindestens ein Foto angefertigt werden, auf dem der Name der Bohrung, Nummer des Kernmarsches und die Teufenangaben eindeutig zuzuordnen sind. Wenn Störungen oder andere Besonderheiten angetroffen werden, die auf dem Foto des gesamten Kernmarsches nicht ausreichend erkennbar sind, sollte gegebenenfalls ein Detailfoto angefertigt werden.

2.2.1 RQD-Index

Der RQD-Index wurde 1963 von D. U. DEERE entwickelt. Die Abkürzung RQD steht hierbei für „Rock Quality Designation“. Ursprünglich wurde dieser Parameter als vollständige Felsklassifikation konzipiert, heute wird er allerdings nur noch als Quantifizierung für den Zerbrechungsgrad von Bohrkernen verwendet [10], [14].

Der RQD-Index geht als Basiswert in eine Vielzahl von Klassifikationssystemen ein:

- RMR- System (vgl. Seite 19)
- RMS-System (vgl. Seite 31)
- Rock Mass Quality Q (vgl. Seite 25)
- Rock Mass Quality Q_{TBM} (vgl. Seite 29)

Die Berechnung des RQD erfolgt nach folgender Formel:

$$RQD = \frac{\sum l}{S} * 100\% \quad [\%] \quad \text{Gleichung 1: RQD-Index}$$

Die Eingangswerte sind folgendermaßen definiert:

- $\sum l$ Summe der Kernstücklängen mit $l > 10$ cm
- S Länge der Kernstrecke

Als Charakteristik der Gebirgsgüte gibt der RQD-Index den prozentualen Anteil der Kernstücke mit einer Länge von über 10 cm an, bezogen auf eine Kernstrecke von 1,0 m oder die Länge des Kernmarsches. Der erhaltene Zahlenwert weist einen Wertebereich zwischen 0 und 100 % auf, wobei 100 % ein sehr gutes Gebirge mit hoher Standfestigkeit und 0 % ein Gebirge ohne Standfestigkeit mit sehr ungünstigen Eigenschaften darstellt [14]. (Tabelle 3)

Tabelle 3: Einteilung in Klassen - RQD-Index

RQD-Index [%]	Felsqualität
0 - 25	V sehr gering (very poor)
25 - 50	IV gering (poor)
50 - 75	III mittel (fair)
75 - 90	II gut (good)
90 - 100	I ausgezeichnet (excellent)

Mitunter werden die in Tabelle 3 angegebenen Klassen noch in Unterklassen „A“ und „B“ eingeteilt (Tabelle 4).

Tabelle 4: Einteilung des RQD-Indexes in Unterklassen

Klasse	ausgezeichnet		gut		mittel		gering		sehr gering	
	A	B	A	B	A	B	A	B	A	B
RQD-Index [%]	100-91	90-76	75-66	65-56	55-46	45-36	35-26	25-16	15-6	5-0

Bei Kernbruchstücken unter 10 cm Länge weist der RQD-Index definitionsgemäß immer den Wert 0 auf. Damit ist eine Unterscheidung zwischen Bruchstücken an der Untergrenze (engständig) und total zermahlenem bzw. mylonitisierendem Gebirge trotz der großen geomechanischen Unterschiede nicht möglich [14]. Des weiteren unterscheidet der RQD-Index bessere Gebirgsbereiche nur sehr unzureichend: falls kein Kernverlust auftritt und alle Bruchstücke über 10 cm Länge liegen, ist der RQD-Index 100 % (ein Kernmarsch von 1,0 m Länge, der als ein Bruchstück der Länge 1,0 m vorliegt, wird einem Kernmarsch mit fünf 20 cm-Bruchstücken gleichgesetzt).

Kernverlust wird von diesem Verfahren ebenfalls nicht berücksichtigt. Somit können mechanisch bedeutsame Bereiche nicht in eine Auswertung einfließen. Um diesen Mangel auszugleichen, wird von DITTRICH [13] vorgeschlagen, für diese Bereiche eine Anzahl stellvertretender Klüfte anzunehmen bzw. durch eine Anzahl von 1 cm langen Kernstücken zu ersetzen.

Die International Society for Rock Mechanics (ISRM) empfiehlt für die Ermittlung von RQD-Werten Doppelkernrohre mit einem Durchmesser von mindestens 54,7 mm [6] (NX diameter, ursprünglich wurde der RQD-Index für diesen Durchmesser entwickelt [10]). LAUBSCHER [24] empfiehlt für die Bestimmung des RQD-Indexes einen Mindestdurchmesser des Bohrkernes von 42 mm (BXM diameter).

Einen Zusammenhang zwischen dem RQD-Index und der Anzahl der Einzelklüfte pro Kubikmeter (J_v) schlägt PALMSTRÖM [4], [21] für tonfreie Gesteine vor:

$$RQD = \begin{cases} 115 - 3,3 * J_v & \text{für } J_v > 4,5 \\ 100 & \text{für } J_v \leq 4,5 \end{cases} \quad [\%] \quad \text{Gleichung 2: Zusammenhang zwischen dem RQD-Index und } J_v$$

mit:

$$J_v \quad \text{Anzahl der Einzelklüfte pro Kubikmeter [m}^{-3}\text{]}$$

2.2.2 Klüftigkeitsziffer k

Auf der Basis der Klüftigkeitsziffer von STINI schlug 1974 MÖBUS [32] ein mathematisch einfaches Verfahren für die Charakterisierung der Intensität der Klüftung vor. Die Berechnung der Klüftigkeitsziffer k erfolgt dabei nach folgender Formel:

$$k = \frac{n}{L} \quad [\text{m}^{-1}] \quad \text{Gleichung 3: Klüftigkeitsziffer k}$$

Die Eingangswerte sind wie folgt definiert:

- n* Anzahl anschnittskorrigierter Kluftschnitte entlang einer Messstrecke
- L* Länge der Kernstrecke parallel zur Bohrlochlängsachse (in [m])

Die manchmal auch als „Trennflächenziffer“ bezeichnete Kennzahl ist ein pauschaler Wert, der über Gruppierung, Orientierung und Abstand der einzelnen Trennflächen nichts aussagt [13]. Der Wertebereich der Klüftigkeitsziffer k reicht theoretisch von 0 bis ∞ . In der Praxis variieren die Werte zwischen $0,1 \text{ m}^{-1}$ und 1000 m^{-1} . Diese Einschränkung des Wertebereiches in beide Richtungen liegt darin begründet, dass in der Natur zum einen keine echten Monolithe in größerem Ausmaß vorkommen (ein Klüftigkeitsziffer von $0,1 \text{ m}^{-1}$ bedeutet, dass alle 10 m eine Kluft vorkommen darf) und zum anderen zu stark zerdrückte Bereiche nicht mehr als Festgestein, sondern als Lockergestein angesprochen werden (eine Klüftigkeitsziffer von 1000 m^{-1} ließe auf einen mittleren Kluftabstand von 1 mm schließen). Die Einteilung in Klassen kann gemäß Tabelle 5 entnommen werden.

Tabelle 5: Einteilung in Klassen - Klüftigkeitsziffer k

k [m^{-1}]	Klüftigkeit
0,1 - 1	weitmaschig
1 - 10	mitteldicht
10 - 100	engständig
100 - 1000	zerdrückt bis mylonitisiert

Die in der Abbildung 5 veranschaulichte Korrektur des Schnitteffektes ist für die Betrachtung der Anzahl der Trennflächen nötig, da annähernd horizontal verlaufende öfter von einer senkrechten Bohrung getroffen werden, als steiler einfallende Klüfte [32]. Die Korrektur erfolgt nach folgender Gleichung: [31]

$$d = m \cdot \sin \alpha \quad [\text{m}] \quad \text{Gleichung 4: Korrektur des Schnitteffektes}$$

mit:

- d anschnittskorrigierter Kluftabstand in [m]
- m nicht anschnittskorrigierter (im Bohrloch gemessener) Kluftabstand in [m]
- α Einfallwinkel der Trennfläche in [°] oder [°]

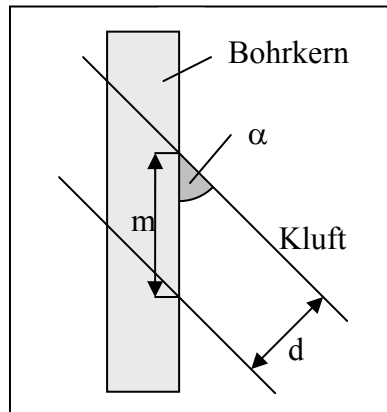


Abbildung 5: Korrektur des Schnitteffektes

MÖBUS [32] schlägt vor, die Korrektur der Klüftigkeitsziffer für jeden Kernmarsch gesondert vorzunehmen. Er schreibt: „Hierzu werden die bei der Kluftaufnahme statistisch registrierten Schnittwinkel verschiedenen Größenklassen zwischen 0° und 90° (bzw. 0° und 100°) zugeordnet, für die sich eine Spannweite von 10° als zweckmäßig erwiesen hat.“ DITTRICH [13] beschreibt eine generelle Abnahme der Klüftigkeitsziffer mit der Teufe, was mit der in der Praxis beobachteten Verbesserung des Gebirges mit zunehmender Teufe in tagesnahem Bereich übereinstimmt.

2.2.3 Mittlere Kernlänge \bar{k}

HANSÁGI [16] stellte 1967 das Verfahren der mittleren Kernlänge \bar{k} vor. Er entwickelte diese Berechnungsvorschrift für die Eisenerzmine von Kiruna, also für den aus mitteleuropäischer Sicht mit hohen Festigkeiten ausgestatteten Syenitporphyr. Die Berechnung der mittleren Kernlänge \bar{k} erfolgt nach folgender Formel:

$$\bar{k} = \frac{k_1 \cdot x_1 + k_2 \cdot x_2 + \dots + k_n \cdot x_n}{\sum x_i} \quad [\text{m}] \quad \text{Gleichung 5: mittlere Kernlänge } \bar{k}$$

Die Eingangswerte sind folgendermaßen definiert:

- x_n Länge der Kernbruchstücke mit $l > 10$ cm
- k_n Mittellängen der Klassen bei einer Klassenlänge von 10 cm (15, 25, 35 cm usw.)

Wie bereits aus der Gleichung 5 und deren Definition der Eingangswerte ersichtlich, beachtet HANSÁGI ausschließlich Kernbruchstücke mit einer Länge von über 10 cm. Die Kernbruchstücke über 10 cm Länge werden mit dem Mittelwert der nächstgelegenen Klasse multipliziert (Klassengröße: 10 cm; Mittellängen 15, 25, 35 cm usw.) [16]

2.2.4 RMR-System

Nach BIENIAWSKI [6] wird das RMR-System fälschlich in einigen Quellen auch als „CSIR Geomechanisches System“ bezeichnet. Die offizielle Bezeichnung ist „RMR-System“ (Rock Mass Rating) oder „Geomechanische Klassifikation“. Dieses System wurde in den Jahren 1972 und 1973 entwickelt, 1976 in allen Details publiziert [5], [17]. Es wurde seit dem intensiv genutzt und mehrfach weiterentwickelt. 1989 wurden von BIENIAWSKI [6] signifikante Änderungen in den Parametern des RMR vorgenommen. Er listet in seinem Buch „Engineering Rock Mass Classification“ 351 Fälle auf, in denen das RMR-System genutzt wurde.

Durch die langjährige Nutzung und aufgrund der Tatsache, dass das RMR-System vielfach weiterentwickelt wurde, geht es als Basiswert in eine Vielzahl von Klassifikationssystemen ein, so auch unter anderem in folgende Klassifikationen:

- MRMR-System (vgl. Seite 21)
- MBR-System (vgl. Seite 22)
- SMR- System (vgl. Seite 31)

Die Berechnung des RMR erfolgt nach folgender Formel:

$$RMR = R_{RQD} + R_{q_u} + R_a + R_b + R_c + R_d \quad \text{Gleichung 6: RMR-System}$$

Die Eingangswerte sind folgendermaßen definiert:

R_{q_u} Wertung für die einaxial ermittelte Gesteinsdruckfestigkeit

R_{RQD} Wertung für RQD-Index nach DEERE (vgl. Seite 15)

R_a Wertung für den Kluftabstand

R_b Wertung für die Kluftbeschaffenheit

R_c Wertung für die Raumstellung der Klüfte

R_d Wertung für den Zutritt von Grundwasser

Die Eingangswerte für die Parameter können aus Anlage II: Rock Mass Rating System (nach BIENIAWSKI [6]) entnommen werden.

Der Wertebereich des RMR liegt zwischen 0 und 100. Er setzt sich aus sechs Summanden zusammen, die die einaxiale Gesteinsfestigkeit, den RQD-Index nach DEERE, den Kluftabstand, Kluftbeschaffenheit, Raumstellung der Klüfte und den Zutritt von Grundwasser repräsentieren. Diese Summanden werden auf der Basis der zuvor genannten Parameter anhand einer von BIENIAWSKI zusammengestellten Tabelle (vgl. Anlage II: „Rock Mass Rating System (nach BIENIAWSKI [6])“) ermittelt. Als alternative Bestimmungsmöglichkeit für die Parameter R_{qu} , R_{RQD} und R_a gibt BIENIAWSKI in seinem Buch „Engineering Rock Mass Classification“ [6] auf den Seiten 56 und 57 zusätzlich noch Diagramme zu einer kontinuierlichen Bestimmung der Eingangswerte an.

Die daraus resultierende Einteilung in Klassen kann der Tabelle 6 entnommen werden:

Tabelle 6: Einteilung in Klassen - RMR-System

RMR	Klasse
81 - 100	I sehr gut (very good rock)
61 - 80	II gut (good rock)
41 - 60	III normal (fair rock)
21 - 40	IV schlecht (poor rock)
0 - 20	V sehr schlecht (very poor rock)

Eine sehr günstige Darstellung der in der Tabelle 6 aufgelisteten Klassen bei der Verwendung dieses Systems im Hohlraumbau stellt die Auftragung in dem von LAUFFER [26] erstmals vorgestellten und nach ihm benannten „LAUFFER-Diagramm“ dar. Wie aus Abbildung 6 ersichtlich, wird in dem angesprochenen Diagramm die Standzeit und die freie Stützweite logarithmisch gegeneinander aufgetragen. Über die Lage des jeweiligen Wertepaares Standzeit und Stützweite kann auf die Klasse (und umgekehrt) geschlossen werden.

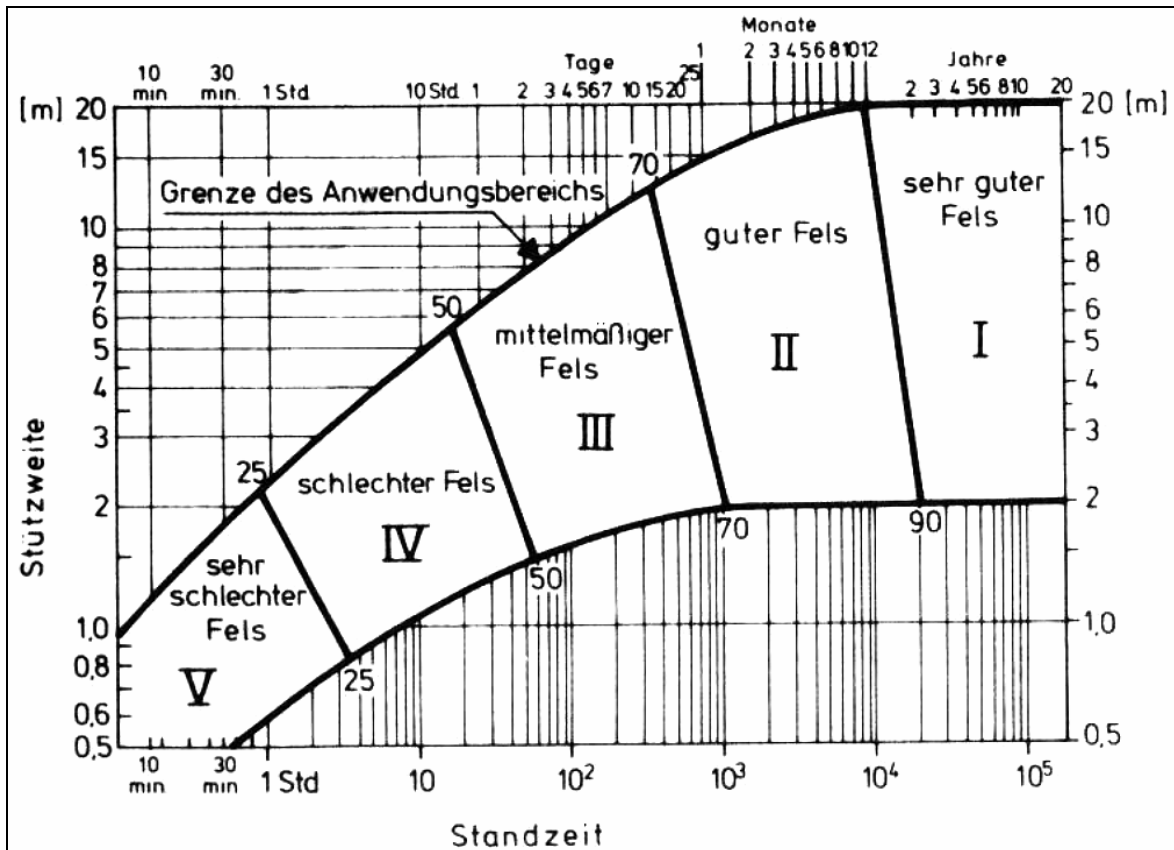


Abbildung 6: Gebirgsklassen nach BIENIAWSKI im LAUFFER-Diagramm [6]

Für hochqualitative Gebirge ($GSI > 25$), kann der GSI^2 direkt aus der von 1976 stammenden Version des RMR-Systems bestimmt werden. Dazu sollte der Parameter R_d (Zutritt von Grundwasser) auf den Wert 10 (trocken) und der Parameter für die Raumstellung der Klüfte auf Null (sehr bevorzugt) angepasst werden [7]. Bei qualitativ schlechten Gesteinen ist eine Schätzung des RMR-Wertes sehr schwierig, da das Gleichgewicht zwischen den Wertungen keine zuverlässige Basis für die Einschätzung der Gesteinsfestigkeit mehr ergibt. Als Konsequenz daraus sollte bei qualitativ schlechten Gesteinen der GSI nicht aus RMR-Werten abgeleitet werden [17].

Falls die 1989 publizierte Version des RMR-Systems [17] genutzt wird (hier als RMR^{89} bezeichnet), so kann der GSI nach der Formel $GSI = RMR^{89} - 5$, wobei bei dem Parameter RMR^{89} der Wert R_d auf 15 und R_c auf Null gesetzt werden sollte.

MRMR-System

BIENIAWSKI's „Rock Mass Rating System“ wurde ursprünglich für zivile Zwecke entwickelt und aufgrund dessen von der Bergbauindustrie als relativ konservativ

² GSI: „Geological Strength Index“ (HOEK 1995; HOECK, KAISER und BAWDEN 1995)

eingestuft. Um das System für den Bergbau interessanter zu gestalten, wurden verschiedene Modifikationen vorgenommen [17].

LAUBSCHER ([23], [22]), LAUBSCHER & TAYLOR ([24]) und LAUBSCHER & PAGE ([25]) beschreiben ein abgewandeltes RMR-System für den Einsatz im Bergbau. Dieses MRMR (Modified Rock Mass Rating) verwendet den „einfachen“ RMR-Wert und justiert diesen unter Beachtung der in situ - Belastung, eingeleiteten Belastung, der Effekte des Sprengens und der Verwitterung (vgl. Abbildung 7). Mit dem errechneten MRMR-Wert sind Ausbauempfehlungen verbunden. Bei dem Gebrauch von LAUBSCHERS MRMR sollte beachtet werden, dass ein Grossteil der Datenbasis des MRMR-Systems durch Blockbau in den Asbestbergwerken Afrikas gewonnen wurde, allerdings später noch weitere Fälle auch aus anderen Erdteilen hinzugefügt wurden [6], [17].

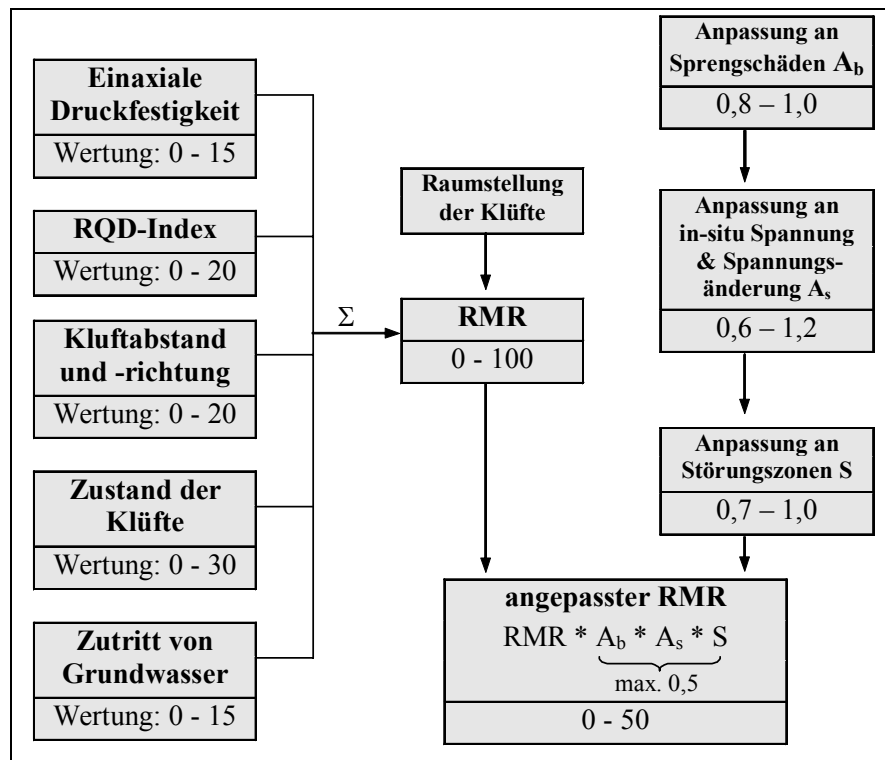


Abbildung 7: Schema für die Bestimmung des MRMR [6]

MBR-System

CUMMINGS et al und KENDORSKI et al [8] modifizierten den RMR ebenfalls und entwickelten das MBR-System („modified basic RMR“) für den Bergbau. Dieses System wurde für den Blockbau in den USA entwickelt und beinhaltet verschiedene Wertungen für die ursprünglichen Parameter des RMR-Systems und die anschließende Anpassung des resultierenden MBR-Wertes an Sprengungen, eingeleitete Spannung, Störungszonen, Abstand zur Abbaufont und der Größe des abgebauten Blockes [6], [17].

2.2.5 Gebirgsfaktor C

HANSÁGI [15] entwickelte 1974 den Gebirgsfaktor C für den Einsatz im Bergbau, Tunnel- und Stollenbau. Dieses Verfahren basiert nicht nur auf den Längen der intakten Kernbruchstücke und deren Mittelwert, sondern auch auf der Anzahl der aus dem Kernmaterial gewinnbaren zylindrischen Probenkörper.

Die Berechnung des Gebirgsfaktors C erfolgt nach Gleichung 7 bzw. Gleichung 10: [15]

$$C = \frac{C_p + C_k}{2} \quad \text{Gleichung 7: Gebirgsfaktor C}$$

mit:

$$\text{Probenfaktor:} \quad C_p = p * \frac{H}{S} \quad \text{Gleichung 8: Gebirgsfaktor C; Probenfaktor}$$

$$\text{Klüftigkeitsfaktor:} \quad C_k = \frac{K}{n * S} \quad \text{Gleichung 9: Gebirgsfaktor C; Klüftigkeitsfaktor}$$

durch einsetzen ergibt sich:

$$C = \frac{1}{2 * S} * (p * H + \frac{K}{n}) \quad \text{Gleichung 10: Gebirgsfaktor C (zusammengefügte Gleichung)}$$

Die Eingangswerte sind folgendermaßen definiert:

- S Länge der Berechnungseinheit in [m]
- p Anzahl zylindrischer Probenkörper, die aus intakten Bohrkernbruchstücken herausgesägt werden können
- H Höhe der zylindrischen Probenkörper in [m]
- K Summe der Kernstücklängen in S , die länger als der Kerndurchmesser sind ([m])
- n Anzahl der Kernstücke mit einer Länge größer als der Kerndurchmesser in S

Als einheitenloser Kennwert besitzt der Gebirgsfaktor C einen Wertebereich von 0,00 (total zerdrückten Fels) bis 1,00 („gesunder Fels“), wobei in die Berechnung nur Kernstücke eingehen, deren Länge größer als der Kerndurchmesser ist. Die Grenzen der einzelnen Klassen können der Tabelle 7 entnommen werden:

Tabelle 7: Einteilung in Klassen - Gebirgsfaktor C

C-Faktor	Felsqualität
0,00 - 0,15	sehr schwach (very poor)
0,15 - 0,30	schwach (poor)
0,30 - 0,45	mittel (fair)
0,45 - 0,65	gut (good)
0,65 - 1,00	ausgezeichnet (excellent)

Die Berechnungslänge S ist nicht streng mit der Kernmarschlänge gleich zu setzen. Die exakte Berechnungslänge hängt von dem erhaltenen Kernmaterial ab, da das Ende des Intervalls entweder mit einem Ende eines Kernbruchstückes abschließen oder in eine Zone mit einem Erhaltungsgrad³ C, D bzw. E fallen muss.

Bei der Berechnung und Auswertung des Gebirgsfaktors C legt HASÁGI keine feste Größe für die Höhe der zylindrischen Probenkörper fest⁴. Bei dem Vergleich von Werten des Gebirgsfaktors C sollten möglichst nur Werte mit gleicher Probenhöhe gegenübergestellt werden.

2.2.6 L_m -Wert

Der L_m -Wert stellt das gewogene arithmetische Mittel von festgelegten Kernstückklassen dar. Er wurde 1974 von MEIER [29] entwickelt und ist in der Lage, auch total zerbohrte Bereiche beziehungsweise Kernverluste (Erhaltungsgrade³ C, D bzw. E) zu bewerten. Neben den „klassischen“ Einsatzgebieten, wie z.B. die Erkundung von Wiederlagern und Gründungen wurde der L_m -Wert bereits bei der Erkundung und Vorhersage der nötigen Verpressmengen von historischen kalkgebundenen Natursteinfundamenten erfolgreich angewendet [28].

Die Berechnung des L_m -Wertes erfolgt nach folgender Formel:

$$L_m = \frac{k_1 * \sum x_1 + k_2 * \sum x_2 + \dots + k_n * \sum x_n}{S} \quad [\text{m}] \quad \text{Gleichung 11: } L_m\text{-Wert}$$

Die Eingangswerte sind folgendermaßen definiert:

$\sum x_i$ Summe der gemessenen Kernstücke der Klassen in [m]

k_i Mittellängen der Klassen in m, d.h. 0,025; 0,075; 0,125 m usw.

S Länge der Berechnungseinheit in [m]

$$S = \sum x_i$$

Der L_m -Wert ergibt sich aus 5-cm-Klassen der einzelnen Kernstücklängen und bezieht sich auf eine Auswerteeinheit mit einer Länge von 1 m. Die Länge der Auswerteeinheit kann je nach tektonischer oder petrografischer Situation variiert werden, sollte aber nicht zu groß gewählt werden, um Details nicht durch Mittelwertbildung zu unterdrücken [28]. Die

³ vgl. Tabelle 2, Seite 14

⁴ In dem 1974 von HANSÁGI verfassten Artikel [15] finden als Probenhöhe H die Werte $H = D$ und $H = 0,5 * D$ Verwendung (wobei D den Bohrkerndurchmesser symbolisiert).

5-cm-Klasse (k_1) enthält die Differenz zwischen der Summe aller Kernstücke > 5 cm und der Auswerteeinheit und ermöglicht somit die Bewertung der Bereiche mit den Erhaltungsgraden C, D oder E [28], [30].

Der Wertebereich des L_m -Wertes hängt von der gewählten Länge der Berechnungseinheit (S) und der Klassenlänge (k , in der Regel 0,05 m) ab. Die Untergrenze entspricht der Mittellänge der ersten Klasse (k_1) und die Obergrenze der Mittellänge der größten Klasse (k_n). Bei einer 1 m langen Auswerteeinheit und einer Klassenlänge von 0,05 m beträgt die Untergrenze 0,025 m und die Obergrenze 0,975 m.

Der Zahlenwert der größten Klasse k_n lässt sich nach Gleichung 12 berechnen: [30]

$$k_n = \left(\text{Fix} \left(\frac{S}{k} + 0,5 \right) - 0,5 \right) * k \quad [\text{m}] \quad \text{Gleichung 12: } L_m\text{-Wert; größte Klasse}$$

mit:

k Länge der Klassen in [m] (in der Regel 0,05 m)

$\text{Fix}(x)$ Die Funktion „Fix“ schneidet alle Nachkommastellen ab und gibt den ganzzahligen Anteil zurück.

Die Verknüpfung zwischen der quantitativen und einer qualitativen Gebirgsklassifizierung ist in der Tabelle 8 zusammengestellt [28].

Tabelle 8: Einteilung in Klassen - L_m -Wert

L_m -Wert	Gebirgseigenschaften	
$< 5,0$	sehr gering	mürbe, teils zersetzt, sehr stark verwittert und aufgelockert, sehr kleinklüftig
5,1 - 10,0	gering - mittel	stark geklüftet und aufgelockert, teils mürbe, angewittert, lagig
10,1 - 15,0	mittel - gut	teils kompakt, frisch, quarzreich, auf Klüften wenig angewittert, gering entfestigt
15,1 <	gut - sehr gut	kompakt, fest bis sehr fest, frisch, quarzreich (Beton B25 $> 17,5$ cm)

2.2.7 Rock Mass Quality Q und Q_{TBM}

Rock Mass Quality Q

1974 wurde von BARTON, LIEN und LUNDE [4] diese Gebirgsklassifikation für den Entwurf und die Bemessung von Tunnelausbau vorgestellt. Bei der Entwicklung dieses - manchmal

auch als „NGI-System“ bezeichneten - Verfahrens wurden von den drei Autoren ca. 200 fertig gestellte Tunnelbauten untersucht, deren Gebirgsqualität (Q-Wert) mit dem verwendeten permanenten Ausbau in Beziehung gesetzt und Gleichungen für den zu erwartenden Druck auf den Ausbau angegeben [4]; 1993 wurde nach GRIMSTAD und BARTON [4] die Rock Mass Quality Q bereits in mehr als 1250 Fällen angewandt. Die Berechnung erfolgt nach folgender Formel:

$$Q = \frac{RQD}{J_n} * \frac{J_r}{J_a} * \frac{J_w}{SRF}$$

Gleichung 13: Rock Mass Quality Q

Dieser Term kann in folgende Teilterme zerlegt werden:

$\frac{RQD}{J_n}$ Kluftkörpergröße

Scherspannungen zwischen den Kluftkörpern

$$\frac{J_r}{J_a} \quad \frac{J_r}{J_a} \cong \tan \phi \quad [1]$$

Gleichung 14: Annäherungsgleichung für die Scherspannungen zwischen Kluftkörpern

$\frac{J_w}{SRF}$ wirksame Spannung

Die Eingangswerte sind folgendermaßen definiert:

RQD RQD-Index nach DEERE (vgl. Seite 15), für Beträge unter 10 wird als stellvertretender RQD-Index der Wert 10 angenommen.

J_n Anzahl der Kluftscharen, Kluftsystem-Ziffer

J_r Rauigkeit für die schwächste Spaltebene, Kluftreibungsziffer

J_a Umwandlungsgrad (Charakter der Klüfte oder Füllung längs der schwächsten Spaltebene), Kluftverwitterungsziffer

J_w Kluftwasserführung, Kluftwasserreduktionsziffer

SRF Spannungsreduktionsfaktor

Die Eingangswerte für die Parameter können aus Anlage I: Eingangswerte Rock Mass Quality Q entnommen werden.

Die aus den sechs oben aufgelisteten Parametern gebildete Rock Mass Quality Q stellt eine numerische, dimensionslose Leitziffer dar. Sie umfasst einen Wertebereich von 0,001 für äußerst schlechten Fels bis 1000 für hochwertigen, fast bruchfreien Fels. Die Einteilung in die einzelnen Klassen kann der Tabelle 9 entnommen werden:

Tabelle 9: Einteilung in Klassen - Rock Mass Quality Q

Q	Klasse
400 - 1000	1 ausgezeichnet
100 - 400	2 extrem gut
40 - 100	3 sehr gut
10 - 40	4 gut
4 - 10	5 mittel
1 - 4	6 gering
0,1 - 1	7 sehr gering
0,01 - 0,1	8 extrem gering
0,001 - 0,01	9 äußerst gering

Bei der Festlegung der Eingangswerte für die Berechnung sollten folgende Punkte Beachtung finden:

- Der erste Parameter - der RQD-Index - wird unverändert von DEERE [9] übernommen (vgl. Seite 15). Durch die für diesen Parameter geltenden Definitionen wird die minimale in die Bewertungsmethode einfließende Kernbruchstücklänge auf 10 cm beschränkt, wodurch wiederum weder eine Unterscheidung zwischen engständiger Zerklüftung und zermahlenem Gebirge, noch das Einkalkulieren von Kernverlusten möglich ist (vgl. Seite 16).

Sollte der berechnete RQD-Index unter 10 liegen, wird als „stellvertretender RQD-Index“ der Wert 10 angenommen.

- Bei der Bestimmung des Parameters für die Anzahl der Klufscharen (J_n) sollte darauf geachtet werden, dass eine Schieferung oder Schichtung die „echten“ Klüfte überprägen kann. Wenn diese extrem parallelen „Klüfte“ verstärkt auftreten, sollten diese als eine Klufschar gezählt werden oder - falls nur wenige „Klüfte“ sichtbar sind - diese als „ungerichtete Klüfte“ gezählt werden.
- Die Faktoren für die Scherspannungen zwischen den Klufkörpern (J_r und J_a) sollten in Bezug auf die schwächste Klufschar oder tongefüllte Diskontinuität in dem betrachteten Bereich festgelegt werden. Sollte die schwächste Klufschar besonders günstig in Bezug auf die Stabilität orientiert sein, kann die Berechnung von Q auf der Basis der nächst schwächsten Klufschar repräsentativer sein.

Nach BARTON [1] kann der Bruch J_r/J_a ungefähr mit dem Tangens des Reibungswinkels ϕ gleichgesetzt werden.

- Der fünfte Faktor J_w ist ein Maß für die Kluftwasserführung und des durch dieses Wasser aufgebauten hydrostatischen Druckes. Dieser Druck setzt zum einen die wirksamen Spannungen herab und zum anderen kann das eventuell vorhandene Wasser Kluftfüllungen aufweichen bzw. auswaschen, was ein weiteres Herabsetzen der Stabilität nach sich ziehen kann.
- Der letzte Faktor SRF ist als Wert für das Spannungsniveau erstens ein Ausdruck für die Abnahme der Belastung im Fall des Vortriebes durch Scherzonen bzw. tonhaltigen Gesteinen, zweitens für den Gesteinsdruck in festem Gestein und drittens für Schwind- oder Quelldrücke in plastischem, geringfesterem Gestein.

BARTON et al. [4] zeigt auf, dass das Einfallen der Klüfte in ihrer Bedeutung hinter der Parameter J_n , J_r und J_a zurücksteht. Weiterhin führt er aus, dass bei Versuchen, den Einfallwinkel der Klüfte einzubeziehen das System an Signifikanz verloren hätte. Implizit ist das Einfallen der Klüfte jedoch in den Parametern J_r und J_a enthalten, da diese sich per Definition auf die am ungünstigsten orientierten Klüfte beziehen [6].

Um die bautechnische Durchführung und Sicherung einfacher und ökonomischer zu gestalten, weist BARTON [4] darauf hin, dass es sinnvoll ist, größere gleich bleibende Q-Bereiche zu erfassen.

Für die Bestimmung des Ausbaudruckes im Firstbereich (p_{Firste} in $[N/mm^2]$) werden folgende Gleichungen angegeben: [4]

$$p_{Firste} = \begin{cases} \frac{0,2}{J_r} * Q^{-\frac{1}{3}} & \text{für mehr als 3 Kluftsysteme} \\ \sqrt{\frac{J_n}{J_r} * \frac{Q^{-\frac{1}{3}}}{15}} & \text{für weniger als 3 Kluftsysteme} \end{cases}$$

Gleichung 15: Ausbaudruck im Firstbereich

Für die Ulmenbereiche wird keine Berechnungsformel angegeben, da davon ausgegangen wird, dass die Ausbaudrücke dort geringer sind und im Bereich von einem Drittel der Firstdrücke liegt. Unberücksichtigt bleiben dabei jedoch tektonische Restspannungen, die das Verhältnis von First- und Ulmendruck verändern können [27].

1993 wurden von SHEOREY [19] einige Vorschläge für Änderungen der Rock Mass Quality Q bei der Anwendung des Systems in Kohlebergwerken unterbreitet, da einige geologische Merkmale im Originalsystem nicht enthalten bzw. ungünstig bewertet waren. Aufgrund von aufgetretenen Versagensfällen wurden im Jahr 1994 Änderungen an den Parametern der Rock Mass Quality Q veröffentlicht [2], [11]. Dabei wurden signifikante Änderungen an dem Spannungsreduktionsfaktor (SRF) und dem „excavation support

ratio“ - Faktor (ESR), welcher für die Bestimmung des erforderlichen Ausbaus benötigt wird, vorgenommen. Die Tabelle 10 stellt die Änderungen am *SRF*-Faktor zusammen [2].

Tabelle 10: Änderungen am System der Rock Mass Quality Q

Festes Gestein, Probleme aus Gebirgsspannungen	σ_c/σ_1	σ_t/σ_1	SRF
L mäßige Gebirgsschlaggefahr (Rutschungen) nach über einer Stunde in massivem Gestein	5 - 3	0,5 - 0,65	5 - 50
M Rutschungen und Gebirgsschlaggefahr nach wenigen Minuten in massivem Gestein	3 - 2	0,65 - 1	50 - 200
N hohe Gebirgsschlaggefahr und unmittelbare dynamische Deformationen in massivem Gestein	< 2	> 1	200 - 400

EICHLER [14] merkt an, dass dieses System leicht zu Meinungsverschiedenheiten führen kann, da alle einfließenden Parameter dieses Systems geschätzt werden müssen und so z.B. der Spannungsreduktionsfaktor je nach geologischen Gegebenheiten scheinbar willkürlich bis auf den Wert 200 gesteigert werden kann.

Rock Mass Quality Q_{TBM}

Von BARTON und JHUN ITOH [3] wurde 2000 das System der Rock Mass Quality Q zu „ Q_{TBM} “ weiterentwickelt und auf den Einsatz von Tunnelbohrmaschinen (TBM) optimiert. Die Hauptaufgabe der Q_{TBM} liegt in der Voraussage des Bohrfortschritts („penetration rate“ bzw. „PR“) und der Vortriebsrate („advance rate“ bzw. „AR“), welche für die Planung und den Betrieb von Tunnelbohrmaschinen eine große technische als auch wirtschaftliche Bedeutung besitzen.

Die Berechnung der Rock Mass Quality Q_{TBM} erfolgt nach folgender Formel: [3]

$$Q_{TBM} = Q * \frac{\text{Sigma}}{F^{10} / 20^9} * \frac{20}{CLI} * \frac{q}{20} * \frac{\sigma_\theta}{5} \quad \text{Gleichung 16: Rock Mass Quality } Q_{TBM}$$

Die Eingangswerte sind folgendermaßen definiert:

- Q Rock Mass Quality Q nach BARTON, LIEN und LUNDE [4] (vgl. Seite 25)
- F Durchschnittliche Schneidemenge (tnf; „tons fealty“ - Tonnen fester Stoff, ohne Auflockerungsfaktor⁵) in der gleichen Zone, genormt auf 20 tnf
- Sigma geschätzte Gesteinsfestigkeit (MPa) in der betrachteten Zone
- CLI Cutter Life Index⁶ (z.B. 4 für Quarzit, 90 für Kalkstein)
- q Quarzgehalt in Prozent
- σ_θ induzierte biaxiale Spannung in der betrachteten Zone, genormt auf eine Tiefe von ca. 100 m

⁵ nach mündlicher Information mit Herrn Dipl.-Ing. L. SCHUMACHER

⁶ JOHANNESSEN et al. NTH, 1994

Durch die Spezialisierung auf Tunnelbohrmaschinen ist eine Übertragung der Ergebnisse auf andere Auswerteverfahren - wie z.B. die Erkundung von Baugrund - nicht möglich.

2.2.8 Zerrüttungsgrad Z_G

Die Entwicklung des Zerrüttungsgrades Z_G in den Jahren 1983 und 1984 durch MOLEK [33] zielte auf eine geotechnische Beurteilung des bruchmechanischen Verhaltens des Deckgebirges in Senkungsgebieten des bedeckten Karstes ab.

Der Begriff „Zerrüttung“ bezieht sich hierbei eindeutig auf die Zerlegung von Festgestein durch tektonische und subrosionsbedingte Bruchprozesse. Verwitterungsbedingte Entfestigungen des Gesteinsmaterials werden hierbei ausgeklammert. Derartige Bereiche sind bereits bei der Kernbemusterung gesondert auszuhalten [33].

Die Berechnung des Zerrüttungsgrades Z_G erfolgt nach folgender Formel: [34]

$$Z_G = f \cdot \frac{\sum l}{n \cdot S} \quad \text{Gleichung 17: Zerrüttungsgrad } Z_G$$

Die Eingangswerte sind wie folgt definiert:

f Abminderungsfaktor durch nicht einmessbare Kernbrocken und Kernverluste, bei fehlenden Kernverlusten und intakten, einmessbaren Kernstücken = 1,0

$$f = \frac{K_S - (K_V + K_B)}{K_S} \quad \text{Gleichung 18: Abminderungsfaktor}$$

K_S Kernstrecke in [m]

K_V Kernverlust in [m]

K_B Länge nicht einmessbarer Kernbrocken (Schutt) in [m]

$\sum l$ Summe der einmessbaren Kernstücke je Berechnungslänge in [m]

S Länge der Berechnungseinheit in [m]

n Anzahl der einmessbaren Kernstücke je Berechnungslänge

Über den Abminderungsfaktor f ist es dem Zerrüttungsgrad Z_G möglich, in dem Wertebereich von 0,0 („völlig entfestigtes Gebirge“) bis 1,0 („völlig intaktes Gebirge ohne Kernverluste, Kernbrocken und einheitlich zusammenhängendem Kern; in der Natur jedoch nicht gegeben“), auch Kernverluste bzw. nicht einmessbare Kernbereiche zu erfassen [33]. MOLEK bezeichnet Kernbruchstücke als „einmessbar“, wenn ihr Kerndurchmesser noch vollständig erhalten ist und eine Länge des Bruchstückes parallel zur Bohrlochlängsachse von mindestens 1 cm vorliegt.

Die Einteilung in Zerrüttungsgrade kann der Tabelle 11 entnommen werden:

Tabelle 11: Einteilung in Klassen - Zerrüttungsgrad Z_G

Z_G	Zerrüttungsgrad
0,08 - 1,00	schwach
0,05 - 0,08	mäßig
0,02 - 0,05	stark
0,00 - 0,02	sehr stark

2.2.9 SMR-System

Als Weiterentwicklung des RMR-Systems (vgl. Abschnitt 2.2.4, Seite 19) wurde im Jahr 1985 von ROMANA [37] das SMR-System (Slope Mass Rating) vorgestellt.

Die Berechnung des SMR erfolgt nach folgender Formel: [18]

$$SMR = RMR + (F_1 * F_2 * F_3) + F_4 \quad \text{Gleichung 19: SMR-System}$$

Die Eingangswerte sind wie folgt definiert:

RMR RMR-Index nach BIENIAWSKI [5] (vgl. Seite 19)

F_1 Winkel zwischen Hang und Kluft. ROMANA [38] gibt als Wertebereich für den Parameter F_1 $0,15 \leq F_1 \leq 1,00$ an.

$$F_1 = (1 - \sin \alpha)^2$$

α Verschneidungswinkel zwischen Hangfläche und Kluftfläche

F_2 Einfallwinkel für den ebenen Versagensfall. ROMANA [38] gibt auch hier wiederum als Wertebereich $0,15 \leq F_2 \leq 1,00$ an.

$$F_2 = \begin{cases} 0,15 & \text{für } \beta_j \leq 21^\circ \\ \tan^2 \beta_j & \text{für } 21^\circ < \beta_j < 45^\circ \\ 1,00 & \text{für } \beta_j \geq 45^\circ \end{cases} \quad (\text{übersteiler Versagensfall})$$

β_j Klufteinfallwinkel

F_3 Relation zwischen Hangoberfläche und dem Klufteinfallen

sehr günstig	(very favorable)	\pm 0
günstig	(favorable)	- 6
gut	(fair)	- 25
ungünstig	(unfavorable)	- 50
sehr ungünstig	(very unfavorable)	- 60

F_4 Parameter für die Vorrichtungsmethode

natürlicher Hang	+ 15
berauben	+ 10
schonendes Sprengen	+ 8
Sprengen oder mechanische Vorrichtung	\pm 0
mangelhaftes Sprengen	- 8

Der aus den ersten drei Parametern gebildete Faktor ($F_1 * F_2 * F_3$) gibt den Einfluss der Klüfte wider. Der vierte Parameter bezieht sich auf die verwendete Vorrichtungsmethode. Aus diesen Parametern und aus dem Wertebereich des RMR-Indexes ergibt sich ein theoretischer Wertebereich für das SMR-System von -68 bis +115. In der Realität wird der Wertebereich allerdings bis auf einen ungefähren Wertebereich von +10 bis +100 eingeschränkt werden.

Die Einteilung in Klassen kann Tabelle 12 entnommen werden:

Tabelle 12: Einteilung in Klassen - SMR-System

SMR	Klasse
81 - 100	I sehr gut
61 - 80	II gut
41 - 60	III normal
21 - 40	IV schlecht
0 - 20	V sehr schlecht

2.3 Vergleich und Diskussion der Auswerteverfahren

Die im Rahmen dieser Arbeit behandelten Verfahren klassifizieren das Gebirge anhand einer Einteilung ihres Wertebereiches in Unterbereiche (Klassen) und damit verbundenen Aussagen zu der Gebirgsbeschaffenheit. In allen Verfahren wird keine Unterscheidung der Anordnung der Kernbruchstücke innerhalb der betrachteten Berechnungseinheit vorgenommen. Es lassen sich also demzufolge auch keine Aussagen über die Lage von Anomalien innerhalb der Berechnungseinheiten vornehmen. Aus diesem Grund wird für alle Verfahren vorgeschlagen, die Berechnungseinheit nicht zu lang zu wählen, da sonst die Gefahr eines „Verwischens“ und „Vergleichmäßigen“ der tatsächlichen Verhältnisse besteht. Als praktikable Berechnungslänge, bei der diese Effekte nicht übermäßig auftreten, wird von verschiedenen Autoren 1,0 m vorgeschlagen, was in vielen Fällen mit der Kernmarschlänge übereinstimmt [6], [14], [28].

In der Tabelle 13 wurden die behandelten Auswerteverfahren mit ihren wichtigsten Eigenschaften gegenübergestellt. Geomechanische Parameter werden nur von einigen Verfahren (RMR-System, Rock Mass Quality Q und Q_{TBM} und SMR-System) mit in die Berechnung einbezogen. Es fällt auf, dass diese (die geomechanische Parameter beachtenden Verfahren) wiederum von Verfahren abhängen, die nur die Geometrie des Bohrkernes und seiner Bruchstücke in ihre Berechnung einbeziehen (RQD-Index).

Eine direkte Integration aller gemessenen Kernbruchstücke beliebiger Erhaltungsgrade (inklusive von Kernverlusten) findet nur durch den L_m -Wert und den Zerrüttungsgrad Z_G statt. Indirekt werden die Erhaltungsgrade C und D von der Rock Mass Quality Q und Q_{TBM} über den Faktor J_n mit in die Berechnung einbezogen. Die anderen untersuchten Verfahren beachten ausschließlich Erhaltungsgrade A und B. Unterschiede in Bezug auf die Kernbruchstücke liegen nicht nur bei den betrachteten Erhaltungsgraden, sondern auch bei den minimal einbezogenen Kernbruchstücklängen: die Autoren der Verfahren geben unterschiedliche minimale Kernbruchstücklängen bis maximal 10 cm (RQD-Index) an.

Das technologiebezogene Verfahren Q_{TBM} wurde bei den weiterführenden Betrachtungen vernachlässigt, da es eine Spezialisierung der Rock Mass Quality Q auf Tunnelbohrmaschinen darstellt und damit keine Allgemeingültigkeit besitzt.

Um die Verfahren besser untersuchen zu können und Aussagen zu prüfen, wurden für die theoretische Auswertung aus der Gesamtmenge aller möglichen Bruchstückkombinationen mit Kernverlusten für einen minimalen Bruchabstand von 1,0 cm (vgl. Anlage VIII: „Code für Generierung aller Bruchstückkombinationen eines Kernmarsches“ und Anlage IX:

„Anzahl möglicher Kernbruchstückkombinationen“) durch einen Zufallszahlengenerator 4000 Kombinationen ausgewählt und als Datenbasis herangezogen. Alle Werte bezogen sich hierbei auf eine Berechnungslänge von 1,0 m - die Ergebnisse der Untersuchungen lassen sich jedoch in ihren Aussagen auch auf beliebige Kernmarsch- bzw. Berechnungslängen erweitern.

Tabelle 13: Vergleich der Auswerteverfahren

Auswerteverfahren	Einheit	Einbeziehung von geo-mechan. Parametern	Abhängigkeiten von anderen Verfahren	Berücksichtigung von Kernbruchstücken ⁷					minimale Bruchstücklänge [cm]	Wertebereich (Klammerwerte: theoretischer Wertebereich)	Anzahl der abgeleiteten Klassen	detaillierte Beschreibung
				A	B	C	D	E				
RQD-Index	[%]			●	●				10	0 ... 100	5	Seite 15
Klüftigkeitsziffer k	[m ⁻¹]			●	●				-	0,1 ... 1000	4	Seite 17
mittlere Kernlänge \bar{k}	[cm]			●	●				10	0 ... S	-	Seite 18
RMR-System	-	●	RQD	●	●				10**	0 ... 100	5	Seite 19
Gebirgsfaktor C	-			●	●				D	0,0 ... 1,0	5	Seite 23
L _m -Wert	[cm]			●	●	●	●	●	(5)*	0 ... S	4	Seite 24
Rock Mass Quality Q	-	●	RQD	●	●	○	○	***	10**	10 ⁻³ ... 10 ³	9	Seite 25
Zerrüttungsgrad Z _G	-			●	●	●	●	●	1	0,0 ... 1,0	4	Seite 30
SMR-System	-	●	RMR (RQD)	●	●				10**	10 ... 100 (-68 ... 115)	5	Seite 31
Rock Mass Quality Q _{TBM}	-	●	Q (RQD)	●	●	○	○	***	10**	10 ⁻³ ... 10 ³	9	Seite 29

* Die minimale Bruchstücklänge ist gleich der gewählten Klassengröße k . Alle kleineren Bruchstücke, Kernverluste und Hohlraum werden automatisch in die kleinste Klasse einbezogen.

** Diese Einschränkung wird durch die Begrenzung des RQD-Indexes auf eine minimale Kernbruchstücklänge von 10 cm bedingt.

*** Einbeziehung der Erhaltungsklassen C und D durch den Parameter J_n .

Symbolerklärung:

D Kerndurchmesser

S Länge der Berechnungseinheit

Klammerwerte besitzen nur bedingte Gültigkeit oder sind redundant.

⁷ Erhaltungsgrad: Kurzzeichen nach Tabelle 2, Seite 14

2.3.1 Theoretische Betrachtungen ausgewählter Verfahren

Die getroffenen Aussagen für die Verfahren gelten jeweils für den **gesamten** Definitionsbereich. Dies gilt insbesondere bei den Versuchen einer Korrelation. Bei der Betrachtung von Einzelfällen bzw. speziellen Gebirgstypen kann es zu einer Einschränkung des Definitionsbereiches und somit zu einer Erweiterung der getroffenen Aussagen kommen.

Skalierbarkeit der Verfahren

Eine für den Vergleich der Verfahren mit sich selbst und mit anderen Verfahren elementare Eigenschaft ist die geometrische Skalierbarkeit. Ein Verfahren kann als „geometrisch skalierbar“ bezeichnet werden, wenn bei einer Multiplikation aller Geometrieparameter (Länge der Kernstrecke, Länge der Kernbruchstücke) des Verfahrens mit einem konstanten Faktor sich das Ergebnis nicht ändert.

In dem speziellen Fall der Kernauswerteverfahren müssen einige zusätzliche Festlegungen für die Untersuchung der Skalierbarkeit getroffen werden. Es wurden für nahezu alle Verfahren Mindestkernbruchstücklängen angegeben. Bei einer Multiplikation der Kernbruchstücklängen dürfen keine Stücke unter diese Mindestlänge fallen, die ohne die Multiplikation in die Berechnung eingeflossen wären. Gleiches gilt für den umgekehrten Fall. Es dürfen bei der Multiplikation keine Bruchstücke zusätzlich Betrachtung finden. Tritt einer dieser Fälle auf, so äußert sich das in einem „Sprung“ (Unstetigkeit) in den Ergebnissen der Verfahren, sie erscheinen dann als nicht skalierbar.

In Tabelle 14 wurde die Skalierbarkeit der Verfahren mit einem Beispiel aufgelistet.

Tabelle 14: Skalierbarkeit der Verfahren

Verfahren	Skalierbarkeit gegeben	Beispiel	
		$S = 100 \text{ cm}$, Bruchstücke: 25A/20A/35A/20A, $H = 5 \text{ cm}$ Original (Faktor: 1,0)	Faktor: 2,0
RQD-Index	ja	100,00 %	100,00 %
Klüftigkeitsziffer k	nein	$4,00 \text{ m}^{-1}$	$2,00 \text{ m}^{-1}$
mittlere Kernlänge \bar{k}	(ja)	15,00 cm	15,00 cm
RMR-System	nein	RMR_a von mittlerem Kluftabstand abhängig	
Gebirgsfaktor C	(ja) ⁸	0,625	0,625
L_m -Wert	nein	24,00 cm	50,50 cm
Rock Mass Quality Q	ja	durch RQD-Index	
Zerrüttungsgrad Z_G	ja	25,00	25,00
SMR-System	nein	durch RMR-System	

⁸ Der Gebirgsfaktor C ist nicht frei skalierbar, da die Anzahl der Probekörper, die aus intakten Kernstücken herausgesägt werden können, Unstetigkeiten einbringt. Wird der Parameter H auch der Skalierung unterworfen, wird dieser Effekt zu großen Teilen ausgeglichen.

Die Skalierbarkeit eines Verfahrens ist dann von Interesse, wenn unterschiedlich lange Berechnungseinheiten miteinander verglichen werden sollen. Der Sachverhalt soll an dem in der Tabelle 14 gezeigten Beispiel erklärt werden. In dem Beispiel werden für die Länge eines 1 m langen Kernmarsches (Kernbruchstücke: 25A/20A/35A/20A) die Werte der Verfahren berechnet. Dem gegenüber wird ein 2 m langer Kernmarsch mit doppelt so langen Kernbruchstücken (50A/40A/70A/40A) gestellt. Aus geomechanischer Sicht lässt sich die Aussage treffen, dass die Felsqualität im ersten Fall signifikant geringer ist als im zweiten Fall. Eine Gebirgsklassifikation müsste also für den zweiten Fall ein „besseres“ Gebirge attestieren, als für den ersten Fall - ein Verfahren sollte also **nicht skalierbar** sein. Aus der Tabelle 14 ist ersichtlich, dass der RQD-Index, der Zerrüttungsgrad Z_G und teilweise auch die von ihnen abhängigen Verfahren skalierbar sind und damit den aufgezeigten Beispielen - wider die Geomechanik - gleiche Felsqualität attestieren.

Zerrüttungsgrad Z_G

Die Skalierbarkeit des Zerrüttungsgrades Z_G und deren Einschränkung wurden bereits gezeigt. An einem weiteren „pathologischen“ Beispiel soll eine andere Besonderheit dieses Verfahrens aufgeführt werden. Berechnet man den Zerrüttungsgrad für einen 1 m langen Kernmarsch, der ein 98 cm langes Kernbruchstück des Erhaltungsgrades A enthält und 2 cm Bohrklein oder Kernverlust, so erhält man einen Z_G von 0,96. Wird statt der 2 cm Bohrklein ein 2 cm langes Kernbruchstück gemessen und der Z_G berechnet, so erhält man als Wert 0,50.

Eine minimale geomechanische Abwandlung im Gebirge hat demnach eine Änderung des Funktionswertes von fast 50 % zur Folge! Diese Änderung ist aus mathematischer Sicht auf die Tatsache zurückzuführen, dass für die Berechnung des Zerrüttungsgrades Z_G durch die Anzahl der gemessenen Kernbruchstücke geteilt wird. Bei einer kleinen Anzahl von Kernbruchstücken wirkt sich eine Erhöhung dieser Anzahl stärker aus, als bei größeren Bruchstückzahlen.

Gegenüberstellung RQD-Index und Zerrüttungsgrad Z_G

Für den Zerrüttungsgrad Z_G ergeben sich Vereinfachungen, wenn angenommen wird, dass die Kernstrecke (K_S) der Berechnungseinheit (S) entspricht und die Beschränkung der minimalen Kernbruchstücklänge von 1 cm aufgehoben wird. Wird weiterhin angenommen, dass die Kernstrecke abzüglich des Kernverlustes und der nicht einmessbaren Kernbrocken

gleich der Summe der einmessbaren Kernstücke innerhalb der Kernstrecke entspricht, dann verkürzt sich die Gleichung für den Zerrüttungsgrad Z_G wie folgt:

$$Z_G = \frac{f^2}{n} \quad \text{Gleichung 20: Vereinfachte Formel für den Zerrüttungsgrad } Z_G$$

mit:

f Abminderungsfaktor nach MOLEK, vgl. Gleichung 18

$$f = \frac{K_S - (K_V + K_B)}{K_S} = \frac{\sum l}{S} \quad \text{Gleichung 21: verallgemeinerte Formel für Abminderungsfaktor } f$$

Annahmen:

1. $S = K_S$ Gleichung 22: erste Annahme für Gleichung 21
2. $\sum l = K_S - (K_V + K_B)$ Gleichung 23: zweite Annahme für Gleichung 21

Parameter:

- S Länge der Berechnungseinheit in [m]
- K_S Kernstrecke in [m]
- $\sum l$ Summe der einmessbaren Kernstücke je Berechnungslänge in [m]
- n Anzahl der einmessbaren Kernstücke je Berechnungslänge
- K_B Länge nicht einmessbarer Kernbrocken (Schutt) in [m]
- K_V Kernverlust in [m]

Entfernt man weiterhin die Beschränkung der minimalen Kernbruchstücklängen des RQD-Indexes von 10 cm, so kann der Abminderungsfaktor f mit dem RQD-Index gleich gesetzt werden. Damit kann unter Einbeziehung der Anzahl der Kernbruchstücke ein funktionaler Zusammenhang zwischen dem RQD-Index und dem Zerrüttungsgrad Z_G mit folgender Gleichung angegeben werden:

$$Z_G = \frac{f^2}{n} = \frac{RQD^2}{n} \quad \text{Gleichung 24: Vereinfachte Formel für } Z_G \text{ in Abhängigkeit vom RQD-Index}$$

Annahmen:

1. $f = RQD$ Gleichung 25: Annahme für Gleichung 24

Damit zeigt sich - mit den getroffenen Vereinfachungen - eine quadratische Abhängigkeit zwischen dem RQD-Index und dem Zerrüttungsgrad Z_G mit einer zusätzlichen Einbeziehung der Anzahl der einmessbaren Kernstücke je Berechnungslänge. Diese Abhängigkeit ist auch sehr gut aus der Auftragung von Beispieldaten des RQD-Indexes und des Zerrüttungsgrades Z_G ersichtlich (Abbildung 8).

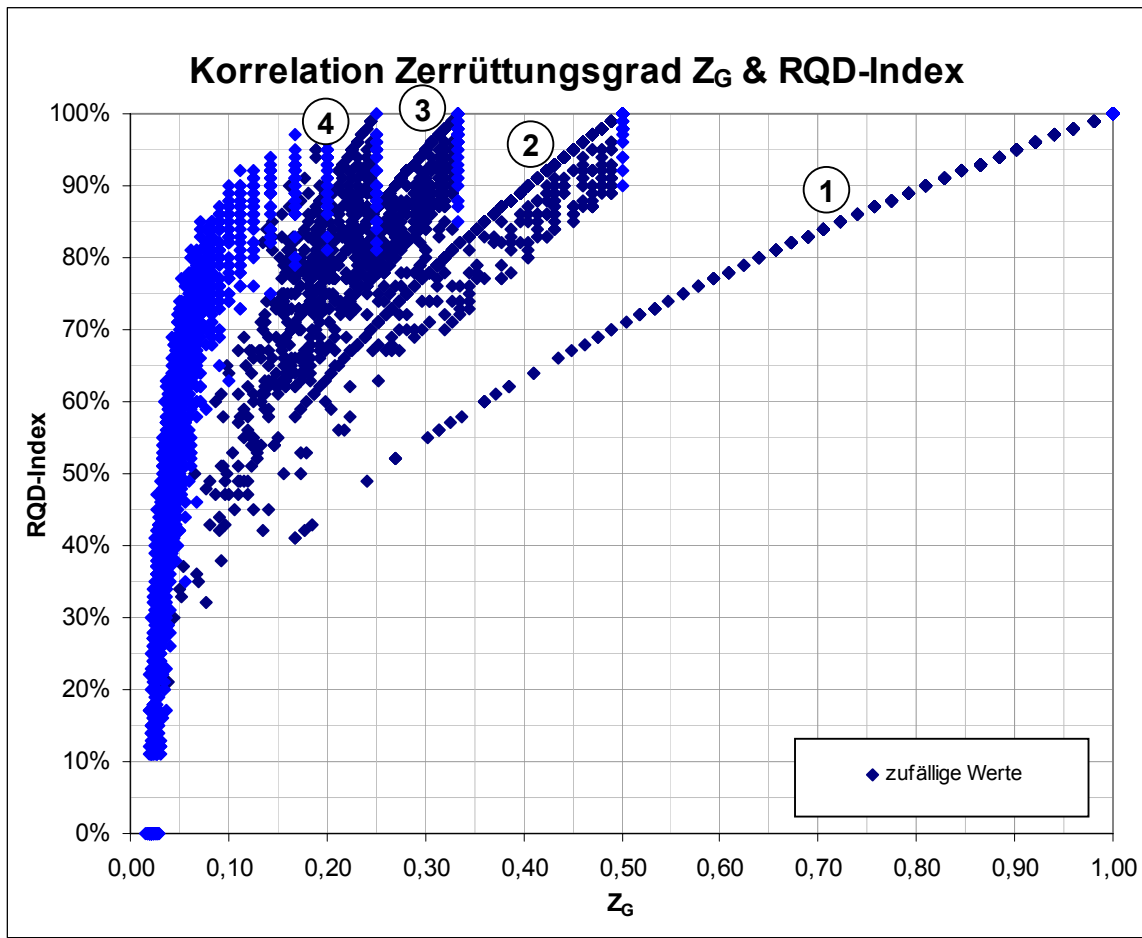


Abbildung 8: Korrelation RQD-Index und Zerrüttungsgrad Z_G

Die in der Abbildung 8 mit ① bis ④ gekennzeichneten unterlinear erscheinenden Häufungen der RQD- Z_G - Wertepaare stellen die (in dieser Darstellung als Wurzelfunktionen in Erscheinung tretenden, vgl. Gleichung 26) Linien gleicher Kernbruchstückzahlen dar. Als Grund für die Streuung der Punkte unterhalb der Linien sind Bruchstücke mit einer Länge unter 10 cm aufzuführen, die in die Berechnung des RQD-Indexes nicht mit einfließen, allerdings bei der Berechnung des Zerrüttungsgrades Z_G Beachtung finden.

Auf den gleichen Grund kann das Fehlen von Wertepaaren zwischen $RQD > 0$ und $RQD \leq 10\%$ zurückgeführt werden. Diese Wertepaare befinden sich auf der Abszisse.

$$RQD = \sqrt{n * Z_G}$$

Gleichung 26: nach dem RQD-Index umgestellte Gleichung 24

L_m -Wert, RQD-Index und Zerrüttungsgrad Z_G

Des weiteren wurde versucht, zwischen den Verfahren L_m -Wert, RQD-Index und Zerrüttungsgrad Z_G eine mehrdimensionale Abhängigkeit abzuleiten (dies ist der Fall, wenn sich ein Verfahren aus den anderen beiden ableiten lässt). Bei einer Darstellung der

drei Verfahren in einem dreidimensionalen Diagramm würde sich eine solche Abhängigkeit in einer Fläche äußern, auf der die Wertetripel liegen. Wie aus Anlage X: „Korrelation L_m -Wert, RQD-Index und Zerrüttungsgrad Z_G “ ersichtlich wird, bilden die Tripel dieser drei Verfahren keine Fläche im 3D aus, womit auch keine eindeutige Abhängigkeit vorliegt. Eine solche Abhängigkeit deutet sich nur für Wertetripel gleicher Kernbruchstückanzahl an, bedarf allerdings weiterer Untersuchung.

RQD-Index und Gebirgsfaktor C

Vergleiche des RQD-Index und des Gebirgsfaktors C wurden bereits mehrfach unternommen ([15], [34]) und auch Korrelationen angegeben (HANSÁGI, 1974, [34]). Aus letztgenannter Publikation (vgl. Abbildung 9) konnte die in Gleichung 27 angegebene Abhängigkeit abgeleitet werden.

$$\text{RQD} = -1650,7 * C^6 + 5029,8 * C^5 - 5432,7 * C^4 + 2379,8 * C^3 - 415,7 * C^2 + 189,5 * C$$

Gleichung 27: nach dem RQD-Index umgestellte Gleichung 24

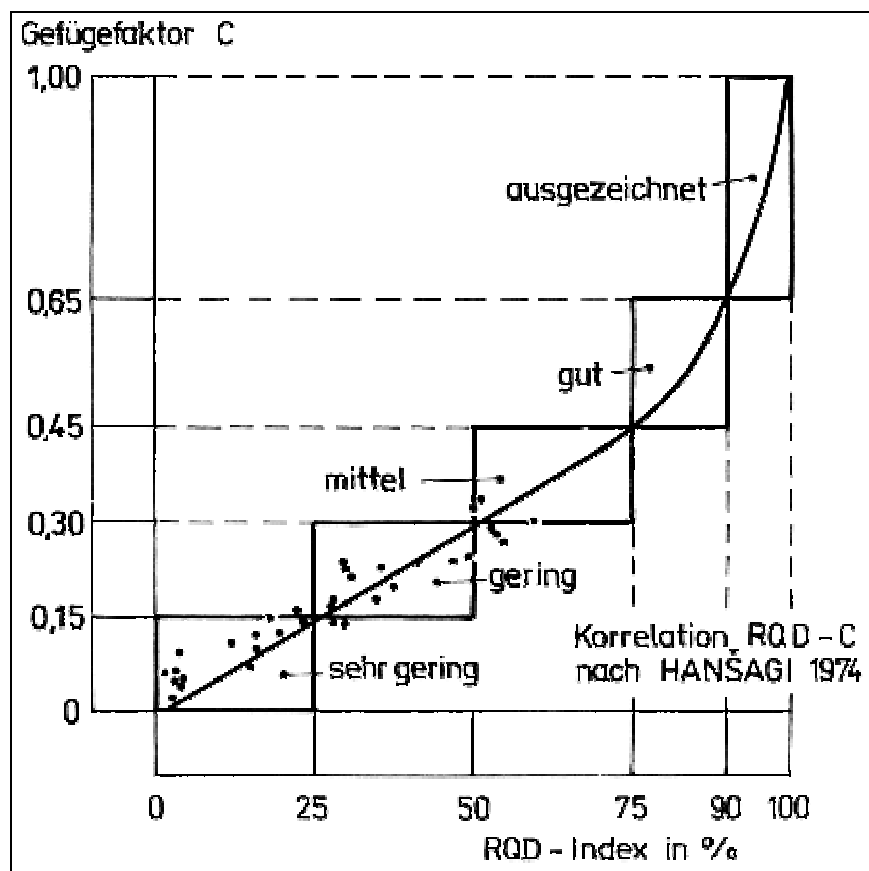


Abbildung 9: Korrelation RQD-Index und Gebirgsfaktor C nach HANSÁGI [34]

Die Abbildung 9 sowie Gleichung 27 wurden an 34 Bohrungen aus dem unteren Buntsandstein abgeleitet. Für eine schnelle Umrechnung der beiden Parameter kann das in Abbildung 10 abgebildete Nomogramm genutzt werden (eine vergrößerte Version des Nomogramms ist nochmals in Anlage III: „Nomogramm der Beziehung zwischen RQD-Index und Gebirgsfaktor C“ abgebildet).

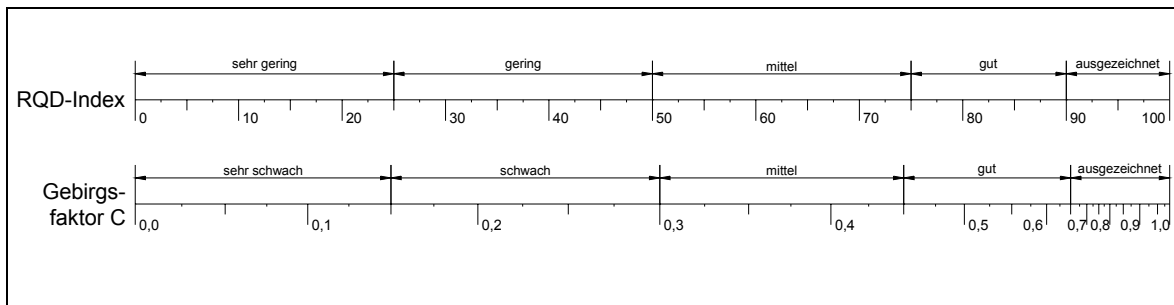


Abbildung 10: Nomogramm der Beziehung zwischen RQD-Index und Gebirgsfaktor C

L_m -Wert und Mittlere Kernlänge \bar{k}

Auf den ersten Blick ähneln sich die Gleichungen der beiden Verfahren L_m -Wert und Mittlere Kernlänge \bar{k} stark. Dennoch korrelieren die Verfahren sehr schlecht, da die Beschränkung der kleinsten Bruchstücke der Mittleren Kernlänge \bar{k} auf 10 cm Länge und die Wahl der Klassengröße auf (relativ hohe) 10 cm dem Verfahren viel Sensibilität nimmt. Der L_m -Wert hingegen beachtet selbst zerdrückte Bereiche und Kernverluste.

RMR-System

Aus den Erfahrungen im Gotthard-Straßentunnel [39], bei dessen Erstellung das System der Ausbruchsklassen mit der Gebirgsklassifizierung von BIENIAWSKI verglichen wurde, leitet SCHNEIDER ab, dass das RMR-System im Bereich schlechter Standfestigkeiten nicht gut mit der Wirklichkeit übereinstimmt. Er führt das vor allem auf die zu stark gewichtete Wasserführung zurück und schlägt vor, diese mit der Gesteinsfestigkeit zu kombinieren. Diese Kombination wird nach seinem Dafürhalten notwendig, da die Standfestigkeit bei wenig festem Fels bereits stark herabsetzt wird, allerdings bei sehr gutem Fels ein starker Wasserzufluss zwar störend auf die Vortriebsarbeiten, aber in sehr beschränktem Maße auf die Standfestigkeit auswirkt.

Rock Mass Quality Q

JOHN [20] merkt an, dass bei dem System der Rock Mass Quality Q weder die Gesteinsfestigkeiten noch die Verformungseigenschaften des Gesteins berücksichtigt werden. Des Weiteren finden die Tatsachen, dass von der Gesteinsfestigkeit der Einfluss der Trennflächen abhängt und die Veränderlichkeit unter Wasserzutritt - wie z.B. Quellen oder Aufweichen - keinen Eingang in die Wertung.

RMR-System und Rock Mass Quality Q

BIENIAWSKI [7] gibt als Zusammenhang zwischen dem RMR-System und der Rock Mass Quality Q folgende Gleichung an:

$$RMR = 9 * \ln(Q) + 44$$

Gleichung 28: Korrelation zwischen RMR-System und Rock Mass Quality Q nach [7]

Zwischenwertung der Verfahren

In Tabelle 15 sind die Vor- und Nachteile der einzelnen Verfahren zusammengestellt.

Vor- und Nachteile der Verfahren

Tabelle 15: Vor- und Nachteile der Verfahren

Verfahren	Vorteile	Nachteile
RQD-Index	<ul style="list-style-type: none"> • einfach zu berechnen • sehr breite Datenbasis (da Parameter für andere Verfahren) 	<ul style="list-style-type: none"> • schlechte Differenzierung geomechanisch sehr unterschiedlicher Gegebenheiten • keine Beachtung von Kernverlusten, vernachlässigen von Bruchstücken ≤ 10 cm Länge • skalierbar • schlechte Differenzierung von besseren Gebirgsverhältnissen
Klüftigkeitsziffer k	<ul style="list-style-type: none"> • einfach zu berechnen 	<ul style="list-style-type: none"> • einheitenbehaftet • keine Beachtung von Kernverlusten und zerdrückten Bereichen
mittlere Kernlänge \bar{k}	<ul style="list-style-type: none"> • einfache Berechnungsvorschrift 	<ul style="list-style-type: none"> • einheitenbehaftet • keine Beachtung von Kernverlusten und zerdrückten Bereichen • bedingt skalierbar
RMR-System	<ul style="list-style-type: none"> • Einbeziehung von geomechanischen Gebirgseigenschaften • Beachtung des Wasserzutritts 	<ul style="list-style-type: none"> • komplexe Berechnungsvorschrift (Tabellen) • keine Beachtung von Kernverlusten • stark subjektiv
Gebirgsfaktor C	<ul style="list-style-type: none"> • keine 	<ul style="list-style-type: none"> • keine Beachtung von Kernverlusten, zerdrückten Bereichen und Kernbruchstücken kleiner dem Kerndurchmesser • skalierbar
L_m -Wert	<ul style="list-style-type: none"> • Beachtung der Erhaltungsklassen C, D und E 	<ul style="list-style-type: none"> • einheitenbehaftet
Rock Mass Quality Q	<ul style="list-style-type: none"> • indirekte Beachtung der Erhaltungsklassen C und D • Einbeziehung von Gebirgseigenschaften • Beachtung des Wasserzutritts 	<ul style="list-style-type: none"> • komplexe Berechnungsvorschrift (Tabellen) • keine Beachtung von Kernverlusten • skalierbar
Zerrüttungsgrad Z_G	<ul style="list-style-type: none"> • Beachtung der Erhaltungsklassen C, D und E 	<ul style="list-style-type: none"> • geomechanisch ähnliche Bereiche haben stark unterschiedliche Z_G-Werte • skalierbar
SMR-System	<ul style="list-style-type: none"> • Spezialverfahren für die Klassifizierung von Böschungen und Hängen 	<ul style="list-style-type: none"> • komplexe Berechnungsvorschrift (durch RMR) • keine Beachtung von Kernverlusten

2.3.2 Darstellung der Datenbasis für den Vergleich der Auswerteverfahren

Als Datenbasis für den Vergleich der Auswerteverfahren wurden 301,65 m Bohrkern verwendet. Die Daten wurden aus vorliegenden Feldbüchern von verschiedenen Objekten entnommen und in eine auf dem beiliegenden Datenträger enthaltene Datenbank eingegeben (Lage auf dem Datenträger: „\Datenbasis\Datenbasis.mdb“). In die Datenbank wurden Kerne von Projekten in folgenden Objekten mit einbezogen (Tabelle 16).

Tabelle 16: Orte der in die Datenbasis aufgenommenen Bohrkern

Objekte	Anzahl der Bohrungen	aufgenommener Bohrkern	anstehendes Gestein
Gottleuba	6	37,95 m	Gneis
Lauenstein	6	173,40 m	Gneis
Johanngeorgenstadt	4	46,70 m	Schiefer
Muldenberg	2	43,60 m	Schiefer
Summe	18	301,65 m	

Neben reinen Geometriewerten wie der Kernmarschlänge und den Längen der einzelnen Bruchstücke wurden für die Auswerteverfahren Rock Mass Quality Q, RMR System und Gebirgsfaktor C weitere Parameter angenommen, die unter Berücksichtigung der Beschreibungen in den Feldbüchern der Projekte gewählt wurden. Aus der Tabelle 17 kann entnommen werden, welche Parameter für welche Verfahren benötigt werden.

Zusätzlich wurden in die Tabelle 17 noch die benötigten Parameter des SMR-Systems mit aufgenommen. Dieses wurde allerdings nicht mit ausgewertet und mit den anderen Verfahren verglichen, da es im Gegensatz zu den anderen Kernauswerteverfahren primär für die Beschreibung und Klassifizierung von Böschungen konzipiert wurde.

Für das Johanngeorgenstädter Objekt wurden in die oben genannte Datenbank die fotografischen Aufnahmen der Bohrkern mit eingefügt. Da für die restlichen Projekte keine Aufnahmen verfügbar waren, konnten diese nicht in die Datenbank aufgenommen werden.

Tabelle 17: Für die Berechnung der Verfahren benötigte Parameter

Parameter	Verfahren								
	RQD-Index	Klüftigkeitsziffer k	mittlere Kernlänge \bar{k}	RMR-System	Gebirgsfaktor C	L_m -Wert	Rock Mass Quality Q	Zerrüttungsgrad Z_G	SMR-System
S	●	●	○	⊙	●	●	⊙	●	⊙
x_n	●	○	●	⊙	○	●	⊙	○	⊙
$R_q(Q_u)$				●					⊙
R_a				●					⊙
R_b				●					⊙
R_c				●					⊙
R_d				●					⊙
p					●				
H					●				
J_n							●		
J_r							●		
J_a							●		
J_w							●		
SRF							●		
$F_1(\alpha)$									●
$F_2(\beta)$									●
F_3									●
F_4									●

Die Erklärung der einzelnen Parameter findet bei der Definition der Verfahren in Abschnitt 2.2 statt.

Symbolerklärung:

- Parameter erforderlich
- ⊙ Parameter erforderlich, durch eingehendes (Unter-)Verfahren
- Parameter bedingt erforderlich (Parameter besitzt für das Verfahren redundante Informationen)

2.3.3 Gegenüberstellung der berechneten Werte

Direkt gegenübergestellt wurden zum einen der RQD-Index, die Klüftigkeitsziffer k , die mittlere Kernlänge \bar{k} , der Gebirgsfaktor C , der L_m -Wert und der Zerrüttungsgrad Z_G und zum anderen das RMR-System mit der Rock Mass Quality Q . Die Trennung zwischen den beiden zuletzt genannten und den restlichen Verfahren wurde getroffen, da diese weitaus mehr und detailliertere Parameter fordern und somit an sie auch höhere Anforderungen gestellt werden können. Damit ergeben sich 15 + 1 mögliche Kombinationen der verschiedenen Verfahren (da der Tausch der Verfahren je Diagramm zwischen Abszisse

und Ordinate lediglich eine Spiegelung an der Funktion $y(x) = x$ darstellt, wurden die „doppelten“ Kombinationen nicht gezählt).

Anlage XI: „Vergleich der Auswerteverfahren (Diagramme)“ enthält die Diagramme mit den eingetragenen Korrelationsfunktionen. Zum Teil wiederholen sich Verfahrenskombinationen - dann allerdings mit verschiedenen Korrelationsfunktionen, da einige Kombinationen besser mit nichtlinearen Funktionen annähern lassen. In der am Ende der Anlage angefügten Tabelle wurden jedoch aufgrund der besseren Vergleichbarkeit untereinander nur die aus linearen Korrelationen stammenden Bestimmtheitsmaße aufgetragen.

Allgemein lässt sich die Aussage treffen, dass die Verfahren innerhalb der Objekte besser korrelieren, als bei der Betrachtung der Gesamtheit aller Daten. Dies erscheint auch plausibel, da innerhalb der Projekte jeweils nur ein recht geringes Spektrum an unterschiedlichen Gesteinen angetroffen wurde und damit die davon abhängigen Eigenschaften (wie z.B. Festigkeit) oder die Klüftung nur gering streuen. Dieser Streubereich vergrößert sich entscheidend bei der zusätzlichen Betrachtung weiterer Objekte.

Am schlechtesten korreliert von den „Low Level“-Verfahren (RQD-Index, L_m -Wert, Klüftigkeitsziffer k , mittlere Kernlänge \bar{k} , Gebirgsfaktor C , und Zerrüttungsgrad Z_G) die Klüftigkeitsziffer k mit den restlichen Systemen. Das äußert sich in dem extrem niedrigen mittleren Bestimmtheitsmaß (R^2) von 0,029 (je nach Objekt schwankt das Bestimmtheitsmaß zwischen 0,114 und 0,000). Des Weiteren zeigt der Zerrüttungsgrad Z_G ein sehr niedriges mittleres Bestimmtheitsmaß von 0,236. Aus diesen Werten folgt, dass von einer Bestimmung der Werte für die Klüftigkeitsziffer k und der Zerrüttungsgrad Z_G aus den restlichen hier diskutierten Verfahren abgeraten werden muss.

Mit Ausnahme der Verfahrenskombination RQD-Index und mittlere Kernlänge \bar{k} (für welche das Bestimmtheitsmaß 0,584 beträgt) korrelieren die restlichen Verfahren gut bis sehr gut miteinander. Die nichtlinearen (logarithmischen) Korrelationen des L_m -Wertes mit den Verfahren RQD-Index, Gebirgsfaktor C und der mittleren Kernlänge \bar{k} mit einem mittleren Bestimmtheitsmaß von 0,843 führen das Feld an. Daraus folgt zum einen, dass eine Abschätzung eines Verfahrens aus den vier anderen Verfahren möglich ist und zum anderen, dass bei einem gegebenen Gebirge die Grundaussagen der vier Verfahren gut übereinstimmen. Die Unterschiede zwischen diesen Verfahren zeigen sich vor allem in der Behandlung von Grenzbereichen und Sonderfällen.

Das RMR-System und die Rock Mass Quality Q wurden in der Literatur schon mehrfach verglichen und gegenübergestellt. Stellvertretend seien an dieser Stelle nur KAISER

“Evaluation of Rock Classification at B.C. Rail Tumbler Ridge Tunnels” [21] und BIENIAWSKI [7] genannt. Die in diesen Literaturstellen angegebene logarithmische Korrelation konnte mit den hier vorliegenden Daten mit leicht abweichenden Korrelationskonstanten mit einem Bestimmtheitsmaß von über 0,8 nachvollzogen werden. Damit ergibt sich folgende Gleichung:

$$RMR = 6,2 * \ln(Q) + 48,3$$

Gleichung 29: Korrelation zwischen RMR-System und Rock Mass Quality Q

3 Versuch einer Gebirgsklassifikation nach dem L_m -Wert anhand von ausgewählten Anwendungsbeispielen

3.1 Gegenüberstellung des L_m -Wertes mit anderen Klassifikationsverfahren

Um die Eigenschaften des L_m -Wertes zu prüfen, die Vergleichbarkeit der Endergebnisse und die Unterschiede mit andern Auswerteverfahren zu verbessern, wurde der L_m -Wert verschiedenen Klassifikationsverfahren gegenübergestellt und versucht aus diesen Gegenüberstellungen Erkenntnisse abzuleiten.

Verglichen wurde der L_m -Wert mit dem RQD-Index und dem Zerrüttungsgrad Z_G . Die Wahl fiel auf diese Verfahren, da ein Vergleich nur mit Verfahren geschehen sollte, welche die gleichen Eingangsdaten nutzen. Um die Wechselbeziehungen zwischen den Verfahren besser untersuchen zu können, wurden wiederum die in Abschnitt 2.3 verwendeten 4000 Bruchstückkombinationen für einen minimalen Bruchabstand von 1,0 cm und eine Berechnungslänge von 1,0 m genutzt. Die Ergebnisse der Untersuchungen lassen sich jedoch in ihren Aussagen auch auf beliebige Kernmarsch- bzw. Berechnungslängen erweitern.

Um eine weitere Verbesserung der Analysemöglichkeiten zu erhalten, wird ein Grenzfall der von MEIER [29] erstellten Gleichung 11 für die Untersuchungen verwendet. Der Grenzfall zeichnet sich dabei dadurch aus, dass die Länge der Klassen k (in der Regel 0,05 m) auf den fiktiven Wert „Null“ gesetzt werden. Die resultierende, abgeleitete Form kann nachfolgender Gleichung entnommen werden:

$$L_m = \frac{\sum x_n^2}{S} \quad [\text{m}] \quad \text{Gleichung 30: } L_m\text{-Wert mit fiktiver Klassengröße Null}$$

mit:

x_n Längen der gemessenen Kernstücke in [m]

S Länge der Berechnungseinheit in [m]

$$S = \sum x_n$$

Eine Gegenüberstellung der „originalen“ und der „abgeleiteten“ Gleichung kann in Abbildung 11 eingesehen werden. Für diesen Graph wurde angenommen, dass je nur ein Kernbruchstück der Erhaltungsklasse „A“ variabler Länge in die Berechnung einfließt.

Durch diese Veränderung ergeben sich folgende Vorteile für die Untersuchung:

- Die Gleichung wird kompakter, was die analytische Handhabung verbessert und das Ableiten von Gesetzmäßigkeiten erleichtert.
- Die Unstetigkeitsstellen (Sprünge im Graphverlauf) werden entfernt, was sowohl wiederum die analytischen als auch die grafischen Untersuchungen erleichtert.

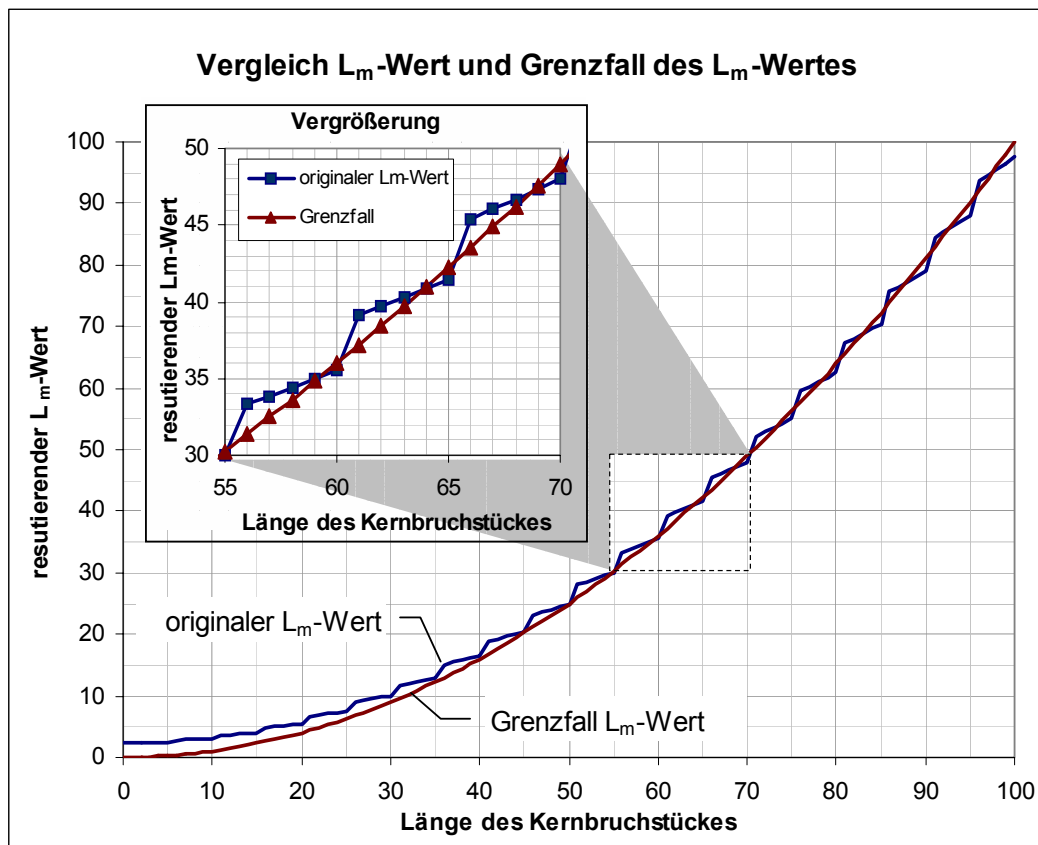


Abbildung 11: Vergleich zwischen L_m -Wert und Grenzfall des L_m -Wertes

3.1.1 Vergleich mit dem RQD-Index

Für die Beziehung zwischen dem RQD-Index und dem L_m -Wert kann keine eindeutige Regressionsfunktion angegeben werden. Es können nur Bereiche ausgegrenzt werden, in denen ein L_m -Wert abhängig von einem RQD-Index bzw. ein RQD-Index abhängig von einem L_m -Wert liegen muss.

In der Abbildung 12 wurden für die Datenbasis die L_m -Werte und die RQD-Indizes berechnet und gegeneinander aufgetragen. Aus diesem Diagramm wird deutlich, dass keine zweidimensionale Regressionsfunktion angegeben werden kann. Auffällig ist allerdings, dass alle Werte im oberen bzw. linken Bereich des Diagramms liegen. Zudem häufen sich die Werte an ausgezeichneten Pfaden.

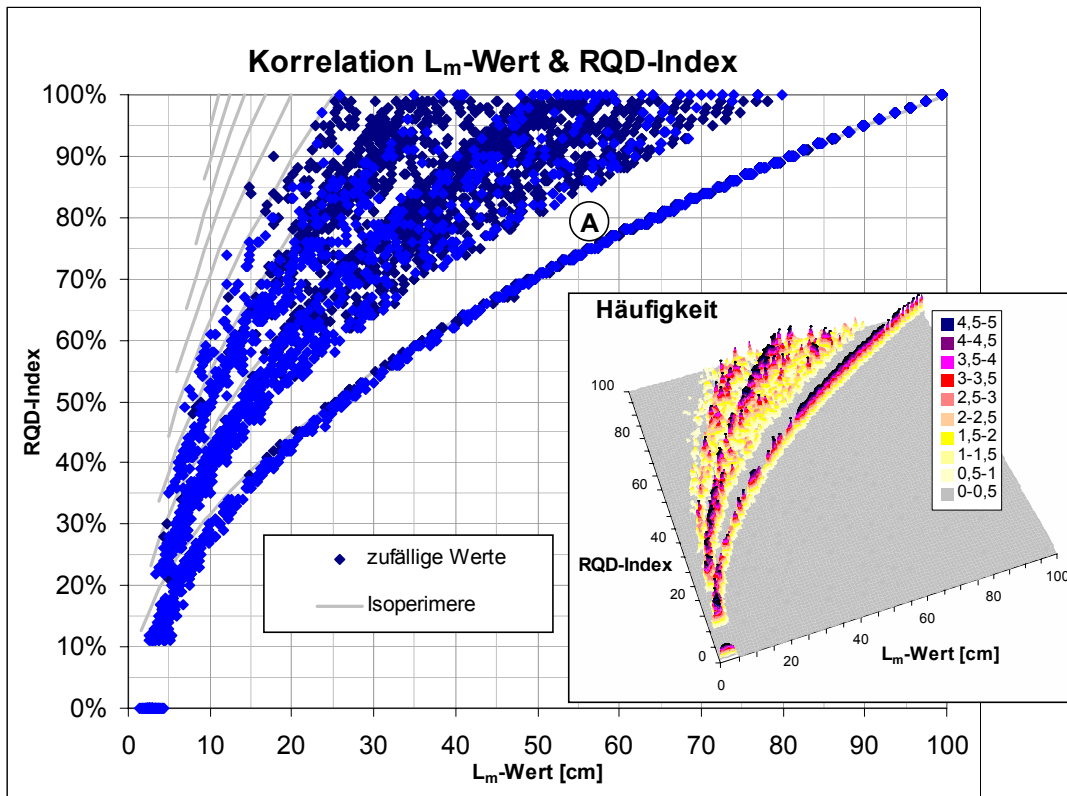


Abbildung 12: Korrelation L_m-Wert und RQD-Index

Anfänglich empirisch und später analytisch konnte gezeigt werden, dass diese ausgezeichneten Pfade Linien gleicher Kernbruchstückzahlen mit gleicher Länge darstellen (im folgenden als „Isoperimere“ bezeichnet). Dabei befindet sich die Linie für ein Kernbruchstück am weitesten rechts im Diagramm. Je höher die Anzahl der Kernbruchstücke wird, desto steiler wird der Anstieg und je weiter links sind die Linien im Diagramm zu finden. Gleichung 31 gibt die Formel für die Berechnung dieser Pfade wieder:

$$RQD_{Iso} = \sqrt{S \cdot n \cdot L_m} \quad [\%] \quad \text{Gleichung 31: Pfade gleicher Kernbruchstückzahlen mit gleicher Länge}$$

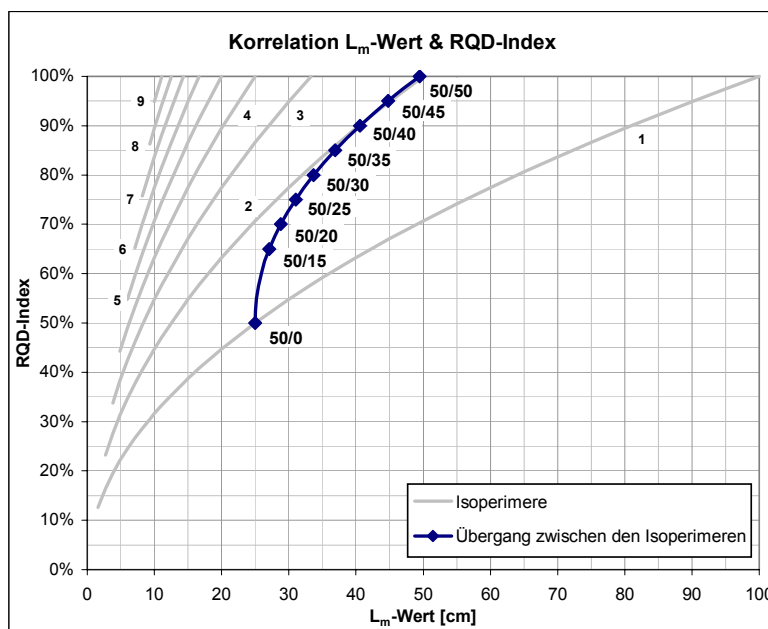
mit:

- RQD_{Iso} RQD-Indizes der jeweiligen n-ten Isoperimere
- S Länge der Berechnungseinheit in [m]
- n Anzahl der Kernbruchstücke mit gleicher Länge
- L_m L_m-Wert, für den der RQD-Index der Isoperimere errechnet wird

Die Streuung der Werte um diese Pfade ist auf zwei Gründe zurückzuführen. Der erste Grund ist die Unstetigkeit des L_m-Wertes, fußend auf der Quantelung der einzelnen

Klassen (Faktoren k_i in Gleichung 11). Durch diese Quantelung streuen die Funktionswerte des L_m -Wertes um die analytische durch Gleichung 31 angegebene Lösung. Als zweiter Grund für die Streuung ist aufzuführen, dass es in der Realität eher unwahrscheinlich ist, dass alle Kernbruchstücke eines Kernmarsches die gleiche Länge aufweisen. Allerdings liegt der L_m -Wert eines Kernmarsches mit gemischt langen Kernbruchstücken mit einem oder mehreren etwa gleich langen und einem oder mehreren kleineren Stücken relativ nahe bei der nächst höheren Isoperimere eines Kernmarsches, der die kleineren Kernbruchstücke nicht enthält.

Diese Eigenschaft zeigt auch Abbildung 13. In das Diagramm wurden die L_m -Werte und RQD-Indizes für einen Kernmarsch (Länge 1,0 m) mit je zwei Kernbruchstücken eingetragen. Die Länge des ersten Kernbruchstückes wurde konstant auf 50 cm gehalten, und die des zweiten Fragments stetig von 0 cm auf 50 cm erhöht. Dabei zeigt sich, dass der Übergang zwischen den beiden Isoperimeren (im Beispiel: von $n = 1$ nach $n = 2$) nicht linear verläuft, sondern sich unterlinear der nächst höheren Isoperimere (im Beispiel: $n = 2$) annähert. Diese Gegebenheit bestätigt auch die in Abbildung 12 aufgetretene Erscheinung, dass sich die Wertepaare nahe den als „ausgezeichnete Pfade“ bezeichneten Isoperimeren befinden.



Bruchstücke	L_m -Wert [cm]	RQD-Index [%]
50/0	25,0	50
50/5	25,2	55
50/10	25,9	60
50/15	27,1	65
50/20	28,8	70
50/25	31,0	75
50/30	33,7	80
50/35	36,9	85
50/40	40,6	90
50/45	44,8	95
50/50	49,5	100

Abbildung 13: Übergang zwischen zwei Isoperimeren

Eine weitere Auffälligkeit ist das Fehlen von Wertepaaren in dem in Abbildung 12 mit \textcircled{A} gekennzeichneten Bereich. Aufgrund der Definition des RQD-Indexes, welcher alle Kernbruchstücke kleiner gleich 10 cm Länge nicht in die Berechnung einfließen lässt, werden die Wertepaare aus diesem Bereich auf der Isoperimere $n = 1$ abgebildet. Der

gleiche Effekt tritt auch über allen andern Isoperimeren ($n > 1$) auf, wird aber durch die Überlagerung von Wertepaaren jeweils höherer Isoperimeren gestört und tritt nicht so deutlich zu tage.

Auf den gleichen Grund ist auch das Abbilden der Wertepaare zwischen $RQD > 0\%$ und $RQD \leq 10\%$ auf $RQD = 0\%$ zurückzuführen. Bei größeren Kernmarschlängen würde diese Schwelle zwar gesenkt werden (z.B. für eine Kernmarschlänge von 2 m auf 5%) aber nicht wegfallen und zudem würden potentiell wichtige Anomalien im Kernmaterial durch Mittelwertbildung verwischt werden.

Basierend auf den oben gezeigten Gegebenheiten kann abgeleitet werden, dass für einen 1 m langen Kernmarsch die Wertepaare in folgendem vereinfachten Bereich liegen:

Abhängige Größe: L_m -Wert:

$$L_m \leq \begin{cases} 10 & \text{für } RQD = 0 \\ 10 + 0,9 * \left(\frac{RQD}{100} \right)^2 & \text{für } RQD > 10 \end{cases} \quad \text{Gleichung 32: Obergrenze des } L_m\text{-Wertes in Abhängigkeit vom RQD-Index}$$

$$L_m \geq \frac{RQD}{10} \quad \text{Gleichung 33: Untergrenze des } L_m\text{-Wertes in Abhängigkeit vom RQD-Index}$$

Abhängige Größe: RQD-Index

$$RQD \leq \begin{cases} L_m * 10 & \text{für } L_m \leq 10 \\ 100 & \text{für } L_m > 10 \end{cases} \quad \text{Gleichung 34: Obergrenze des RQD-Indexes in Abhängigkeit vom } L_m\text{-Wert}$$

$$RQD \geq \begin{cases} 0 & \text{für } L_m \leq 10 \\ \sqrt{S * L_m} - 5 & \text{für } L_m > 10 \end{cases} \quad \text{Gleichung 35: Untergrenze des RQD-Indexes in Abhängigkeit vom } L_m\text{-Wert}$$

Diese Bereichsangabe stellt nur eine Grobeingrenzung der möglichen Wertepaare dar, da durch die Klassenbildung im L_m -Wert Unstetigkeiten auch in die Bereichsgrenzen eingebracht werden und weiterhin die in Gleichung 33 angegebene Gerade nur eine Annäherung an die durch die einzelnen Isoperimeren vorgegebene Grenze darstellt.

3.1.2 Vergleich mit dem Zerrüttungsgrad Z_G

Ähnlich wie für die Beziehung zwischen L_m -Wert und RQD-Index kann auch für die Beziehung des L_m -Wertes mit dem Zerrüttungsgrad Z_G keine eindeutige Regressionsfunktion angegeben werden. In der Abbildung 14 wurden die Verfahren gegeneinander aufgetragen. Auffällig ist, dass für bestimmte Parameter der Verfahren eine lineare, direkte Abhängigkeit feststellbar ist: für diese Eingangswerte entsprechen sich der L_m -Wert und der Zerrüttungsgrad Z_G (vgl. Gleichung 37). Die Gesamtheit aller Funktionswerte liegt auf oder unter dieser Geraden.

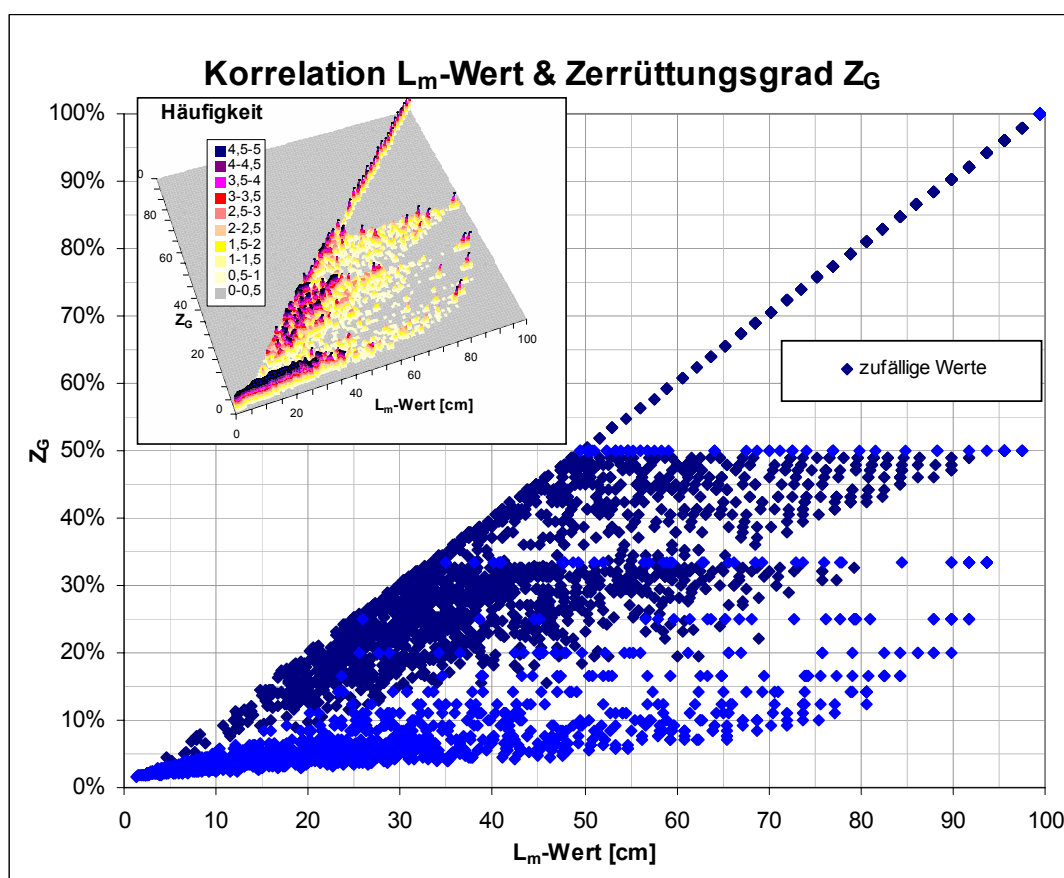


Abbildung 14: Korrelation L_m -Wert und Zerrüttungsgrad Z_G

Empirisch konnte gezeigt werden, dass diese ausgezeichnete Gerade von den Verfahren für Bruchstücke gleicher Längen (alle Bruchstücke haben einheitliche Längen, Kernverlust möglich) eingehalten wird.

Die untere Grenze der möglichen Wertepaare kann über die empirisch bestimmte Gleichung 38 eingegrenzt werden, welche wiederum eine grobe Näherung der Untergrenze der möglichen Wertepaare darstellt.

Auffällig sind in Abbildung 14 die starken Sprünge und Häufungen des Zerrüttungsgrades Z_G . Diese sind auf die bereits in Abschnitt 2.3.1 dargestellte Besonderheit des Verfahrens zurückzuführen, aufgrund welcher bei geringen Änderungen der Eingangsparameter der Funktionswert um bis zu 50% fluktuiert.

Abhängige Größe: L_m -Wert:

$$L_m \geq Z_G$$

Gleichung 36: Untergrenze des L_m -Wertes in Abhängigkeit vom Zerrüttungsgrad Z_G

Abhängige Größe: Zerrüttungsgrad Z_G

$$Z_G \leq \frac{L_m}{100cm} * 100\%$$

Gleichung 37: Obergrenze des Zerrüttungsgrades Z_G in Abhängigkeit vom L_m -Wert

$$Z_G \geq \frac{L_m^\%}{10} + \frac{(L_m^\%)^{15}}{2}$$

Gleichung 38: Untergrenze des Zerrüttungsgrades Z_G in Abhängigkeit vom L_m -Wert

$$\text{mit: } L_m^\% = \frac{L_m}{100cm} * 100\%$$

Zusammenfassend lässt sich sagen, dass der Zerrüttungsgrad Z_G den von ihm abgedeckten Wertebereich sehr ungleichmäßig abdeckt und so verhältnismäßig viele L_m -Werte einem Z_G -Wert zugeordnet werden - der L_m -Wert differenziert die Eingangsparameter deutlicher.

3.2 Vergleich der Klassen

Das Ziel eines Klassifikationssystems, das Gebirge in Klassen aufzuteilen, erreichen die verschiedenen numerischen Verfahren zwar immer mit dem gleichen Basisalgorithmus (Zusammenstellung der Parameter, Einsetzen und Berechnung, Auswahl der Klasse aus einer vorgegebenen Tabelle), aber in den einzelnen Schritten unterscheiden diese sich zum Teil erheblich. Die Anzahl der Klassen schwankt so z.B. zwischen den einzelnen Verfahren recht stark (z.B. 4 Klassen bei der Klüftigkeitsziffer k und 9 Klassen bei der Rock Mass Quality Q).

Auf der Basis der in Abschnitt 2.3.2 vorgestellten Daten wurden der L_m -Wert mit dem RQD-Index, dem Gebirgsfaktor C und dem Zerrüttungsgrad Z_G gegenübergestellt. Ein

Vergleich mit der mittleren Kernlänge \bar{k} war nicht möglich, da vom Autor HANSÁGI [16] keine Einteilung in Klassen angegeben wurde.⁹

Eine absolute und eindeutige Abbildung der Klassen der verschiedenen Verfahren kann es nach den Ergebnissen aus Abschnitt 3.1 nicht geben, da keine eindeutige Korrelation nachgewiesen werden konnte. Hinzu kommen die bereits angesprochene unterschiedliche Anzahl der Klassen der einzelnen Systeme, die in ihrer Gesamtheit jedoch den gleichen Wertebereich umfassen. In Abbildung 15 wurden die Abbildung der Klassen der Verfahren RQD-Index in Gebirgsfaktor C auf die Klassen des L_m -Wertes aufgetragen. Die zugehörigen Diagramme können in Anlage XII: „Vergleich der Klassen der Auswerteverfahren mit den Klassen des L_m -Wertes“ eingesehen werden.

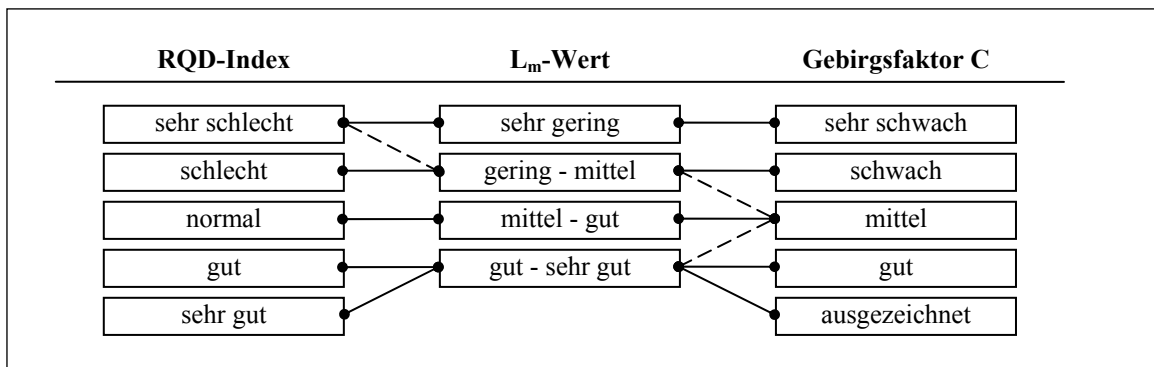


Abbildung 15: Abbildung der Klassen der Verfahren RQD-Index und Gebirgsfaktor C auf die Klassen des L_m -Wertes

Verallgemeinert lässt sich sagen, dass die Klassen des L_m -Wertes recht gut mit denen des RQD-Index und des Gebirgsfaktors C übereinstimmen, jedoch im Bereich von hochwertigen Gebirgsverhältnissen nicht weiter differenziert werden. Aus diesem Grund wird vorgeschlagen, den Klassen des L_m -Wertes eine weitere Klasse mit einer Klassengrenze bei $L_m = 30$ cm hinzuzufügen, so dass die Ergebnisse des L_m -Wertes wie in der Tabelle 18 eingeteilt werden.

⁹ I. HANSÁGI vergleicht die für einen Bereich ermittelte mittlere Kernlänge mit der eines trennflächenfreien Idealgebirges [16].

Tabelle 18: Modifizierte Einteilung in Klassen - L_m -Wert

L_m -Wert	Gebirgseigenschaften	
< 5,0	sehr gering	mürbe, teils zersetzt, sehr stark verwittert und aufgelockert, sehr kleinklüftig
5,1 - 10,0	gering - mittel	stark geklüftet und aufgelockert, teils mürbe, angewittert, lagig
10,1 - 15,0	mittel - gut	teils kompakt, frisch, auf Klüften wenig angewittert, gering entfestigt
15,1 - 30,0	gut - sehr gut	kompakt, fest bis sehr fest, frisch
30,1 <	ausgezeichnet	extrem günstige Verhältnisse, nahezu idealer Fels

Für die Verfahrenskombination L_m -Wert und Zerrüttungsgrad Z_G konnte, wie schon aufgrund des schlechten Korrelationsverhaltens zu vermuten war, keine zufrieden stellende Abbildungsvorschrift für die Klassen der beiden Systeme gefunden werden. Ein nach dem L_m -Wert als „mittel bis gut“ klassifiziertes Gebirge könnte nach dem Zerrüttungsgrad Z_G stark, mäßig oder auch schwach zerrüttet sein.

3.3 Substitution des RQD-Indexes mit dem L_m -Wert

Der RQD-Index ist zwar leicht zu berechnen, besitzt aber die bereits in der Tabelle 15 aufgezählten Nachteile. Dennoch wird er bei der Berechnung verschiedener anderer Klassifikationsverfahren als Parameter herangezogen. Der L_m -Wert differenziert das Gebirge, wie bereits in den Abschnitten 2.3 und 3.1.1 gezeigt, besser als der RQD-Index. Die Diskussion einer Substitution des RQD-Indexes mit dem L_m -Wert ist also naheliegend. Eine eventuelle Substitution kann nicht pauschal für alle „Unterverfahren“ geführt werden, sondern muss für jedes Verfahren neu vorgenommen und auf Sinn und Güte geprüft werden, was jedoch im Rahmen dieser Arbeit nicht möglich ist. Stattdessen soll am Beispiel der Rock Mass Quality Q ein Vorschlag für eine Substitution aufgezeigt werden.

Ziel ist es, folgende Einschränkungen zu beseitigen:

- Beseitigen der Einschränkung auf Kernbruchstücke über 10 cm Länge
- Beachtung bzw. Differenzierung von mylonisierten Bereichen und Kernverlusten
- Bessere Differenzierung von guten Gebirgen
- Entfernung der durch den RQD-Index eingetragenen Skalierbarkeit

In die Rock Mass Quality Q geht der RQD-Index als einer von sechs Parametern ein. Ersetzt man den RQD-Index durch den in Gleichung 39 angegebenen Term, erhält man die in Gleichung 40 angegebene Berechnungsvorschrift.

$$RQD \Rightarrow \frac{(L_m)^{0,8}}{14}$$

Gleichung 39: Term für die Substitution des RQD-Index mit dem L_m -Wert

$$Q^{L_m} = \frac{(L_m)^{0,8}}{J_n} * \frac{J_r}{J_a} * \frac{J_w}{SRF}$$

Gleichung 40: Rock Mass Quality Q mit substituiertem L_m -Wert

Bei der Untersuchung der Gleichung 40 konnte ein Bestimmtheitsmaß (R^2) von 0,806 zwischen der originalen Gleichung für Q und der substituierten Form nachgewiesen werden. Es kann also von einer Korrelation zwischen der ursprünglichen und der abgewandelten Form ausgegangen werden.

Es sei an dieser Stelle darauf hingewiesen, dass diese Gleichung als Vorschlag zu behandeln ist und auch deren Ergebnisse als solche zu kennzeichnen sind. Es sollte aus diesem Grund für die abgewandelte Berechnungsvorschrift für die Rock Mass Quality Q auch das Formelzeichen Q^{L_m} verwendet werden.

Vorerst ungeklärt bleiben unter anderem folgende Punkte:

1. In welchem Maße Verschiebungen im Funktionswert und in der Zuordnung zu den Klassen auftreten.
2. Theoretisch können bei Kernmarschlängen über 1 m auch L_m -Werte über 100 cm auftreten. Ungeklärt ist, wie sich solche Werte im resultierenden Q^{L_m} -Wert äußern.
3. Der Faktor J_n der Gleichung 40 besitzt redundante (bereits im L_m -Wert enthaltene) Informationen. Wird er durch die Konstante 1 ersetzt, sinkt das Bestimmtheitsmaß allerdings um ca. 10% ab.

4 EDV-gerechte Aufarbeitung der Auswerteverfahren und deren Darstellung

Um die Berechnungen der einzelnen Parameter zu vereinfachen und grafisch übersichtlich darzustellen, wurde das Projekt „Gebirgsklassifikation“ geschrieben¹⁰. Es setzt sich zum einen aus der Anwendung „Gebirgsklassifikation“ und zum anderen aus der ActiveX-Komponente „RockClassX“ zusammen.

Bei der Entwicklung der in diesem Kapitel beschriebenen und auf dem beigelegten Datenträger enthaltenen Softwarekomponenten wurden diese getestet und mehrfach durchgesehen. Dennoch können Fehlfunktionen nicht ausgeschlossen werden und eine Garantie oder Gewähr auf eine fehlerfreie Funktion kann nicht gegeben werden.

4.1 Systemvoraussetzungen

Für eine Nutzung wird ein Microsoft Windows 95 bzw. ein Windows NT 4.0 (Service Pack 3 empfohlen) oder höher auf einem Rechner mit mindestens 25 MB freiem Festplattenspeicher¹¹ und mindestens 24 MB Hauptspeicher (RAM) benötigt. Für die Nutzung der hier vorgestellten ActiveX-Komponente (vgl. Abschnitt 4.3: Beschreibung des ActiveX-Projektes „RockClassX“, Seite 66) sollte eine - vorzugsweise mit Microsoft Visual Basic für Applikationen (VBA) - programmierbare Anwendungssoftware eingerichtet sein. Optimal ist ein Windows 2000 mit installiertem Office 2000 Professional und 128 MB RAM oder höher.

4.2 Beschreibung des Programms „Gebirgsklassifikation“

Das Programm „Gebirgsklassifikation“ dient der fensterbasierenden und mausgesteuerten Eingabe, Anzeige und Management von Bohrungsdaten und berechnet die Werte der einzelnen eingebundenen¹² Gebirgsklassifikationsverfahren. Es wurde für den zügigen Einsatz konzipiert und ist auch für weniger versierte Nutzer bedienbar.

¹⁰ Für die Entwicklung wurde ein Microsoft Visual Basic 6.0 SP3 verwendet.

¹¹ Die Angabe des minimalen Festplattenspeichers bezieht sich auf den für das Programm benötigten Speicher. Die abzuspeichernden Daten erfordern zusätzlich Festplattenkapazität.

¹² Über die in Abschnitt 4.3.4 („Beschreibung der Schnittstelle“, Seite 70) beschriebene Schnittstelle können weitere Klassifikationsverfahren eingebunden werden.

4.2.1 Installation des Programms

Zu der Installation des Programms „Gebirgsklassifikation“ ist die Datei „Setup.exe“ im Verzeichnis „\EDV-gerechte Umsetzung\Install\Gebirgsklassifikation“ zu starten. Unter einem WindowsNT, Windows2000 oder höher sind ausreichende Rechte¹³ zur Installation erforderlich. Nach dem Start der Installationsroutine wird nochmals um eine Bestätigung der Installation gebeten und es muss ein Installationsordner angegeben werden. Nach erfolgtem Installationsvorgang wird eine Meldung ausgegeben, die den Erfolg oder Misserfolg dokumentiert. Es wird im Zuge der Installation ein Eintrag im Startmenü vorgenommen, mit deren Hilfe sich das Programm starten lässt.

Sollten ein Windows95 oder Windows98 zum Einsatz kommen, ist darauf zu achten, dass auf dem betreffenden Rechner die Komponente „DCOM98“ fehlerfrei installiert wurde. Diese Komponente ermöglicht den Zugriff auf Datenbanken und ist für die Ausführung der Anwendung „Gebirgsklassifikation“ dringend erforderlich. Die Installationsroutine für die Komponente befindet sich zum einen auf dem beiliegenden Datenträger und kann zum anderen bei Microsoft (www.microsoft.com) heruntergeladen werden.

4.2.2 Bedienoberfläche

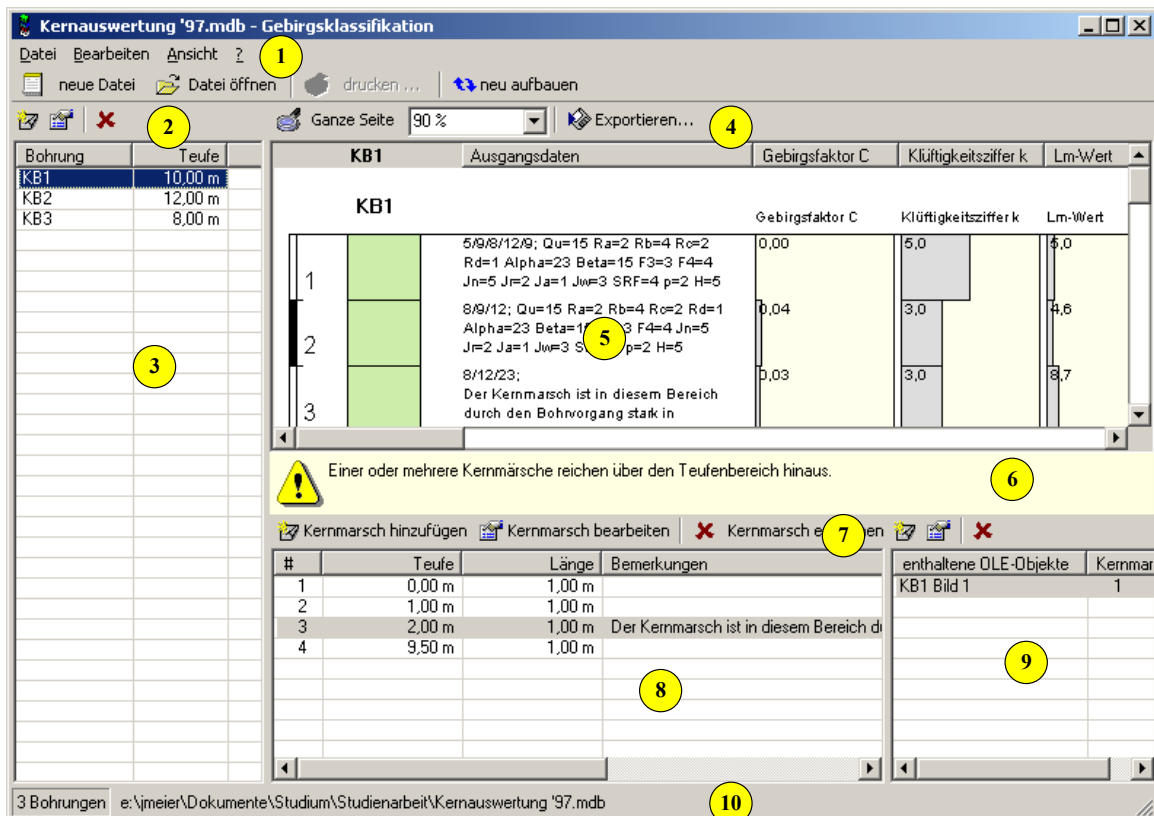
Die Anwendung besitzt - ähnlich dem Editor - die Möglichkeit, nur eine Datei auf einmal zu bearbeiten („Single Document Interface“). Es können allerdings beliebig viele Instanzen gleichzeitig gestartet und so verschiedene Dateien simultan bearbeitet werden.

Die Bedienoberfläche des Programms (vgl. Abbildung 16, die im Folgenden angegebenen „Punkte“ beziehen sich auf diese Abbildung) besteht aus der Menüzeile mit Hauptsymbolleiste (Punkt 1), dem Arbeitsbereich (Punkt 2 bis 8) und der Statusleiste (Punkt 9). Der Arbeitsbereich lässt sich seinerseits wiederum in zwei Unterbereiche aufgliedern: links im Anwendungsfenster befindet sich die Liste der Bohrungen und rechts der Detailbereich für die aktuell ausgewählte Bohrung.

Sobald eine gültige Datei geöffnet wird, füllt sich automatisch die Liste der Bohrungen (Punkt 3) mit allen in der Datei befindlichen Daten und der erste Eintrag wird ausgewählt. Durch die Auswahl einer Bohrung werden im Detailbereich die Kernmärsche der aktiven Bohrung in die Liste der Kernmärsche (Punkt 8) eingetragen und in dem Darstellungsbereich (Punkt 5) grafisch aufgetragen.

¹³ Bei der Installation wird die Komponente am System angemeldet und falls nötig Systemteile aktualisiert oder hinzugefügt. Für diesen Vorgang werden in der Regel Administratorrechte benötigt.

Mit Hilfe der über den genannten Elementen angeordneten Symbolleisten können Operationen mit den Elementen vorgenommen werden (z.B. Einfügen von Bohrungen, Ein- und Auszoomen). Eine Besonderheit stellt der mit Punkt 6 gekennzeichnete Bereich für Fehlermeldungen und Hinweise dar. Er wird angezeigt, wenn bei der Eingabe der Kernmärsche Fehler - wie z.B. überlappende Kernmärsche - gemacht wurden und blendet sich automatisch nach Beseitigung der oder des Fehlers wieder aus.



Nummer	Beschreibung
1	Menüzeile und Hauptsymbolleiste
2	Symbolleiste für das Management der Bohrungen
3	Liste aller verfügbaren Bohrungen
4	Symbolleiste für die aktuelle Bohrung
5	Darstellungsbereich der aktuellen Bohrung

Nummer	Beschreibung
6	Bereich für Fehlermeldungen und Hinweise
7	Symbolleiste der Kernmärsche
8	Liste der Kernmärsche der aktuellen Bohrung
9	Liste der eingefügten Objekte der Bohrung
10	Statusleiste

Abbildung 16: Oberfläche des Programmes

4.2.3 Erstellen, Öffnen und Speichern von Dokumenten

Das Erstellen einer neuen Datei erfolgt nicht automatisch bei dem Start des Programmes. Für die Erstellung kann zum einen das erste Symbol der Hauptsymbolleiste („neue Datei“) oder der Befehl „neu...“ im Menü „Datei“ genutzt werden. Nach dem Auslösen des Befehls wird ein Fenster ähnlich Abbildung 17 geöffnet, in dem der Dateiname und Pfad der zu erstellenden Datei (genauer einer neuen Datenbank) eingegeben werden können. Dies ist nötig, da alle Änderungen sofort auf die Datei geschrieben werden. Aufgrund dieses für Datenbanken typischen Verhaltens ist es nicht nötig, die Daten nach einer

Bearbeitung explizit zu speichern. **Änderungen an den Daten werden sofort in die Datei übernommen.**

Über den Befehl „Datei öffnen“ der Hauptsymbolleiste oder dem Befehl „öffnen...“ im Menü „Datei“ kann eine bestehende Datenbank geladen werden (vgl. Abbildung 17). Es ist dabei auch möglich, Dateien im lokalen Netzwerk beziehungsweise Intranet anzugeben und gegebenenfalls auch gleichzeitig mit anderen Nutzern zu bearbeiten. Die neun zuletzt bearbeiteten Dateien werden in das Menü „Datei“ aufgenommen und können so beim nächsten Start schneller wieder geöffnet werden.

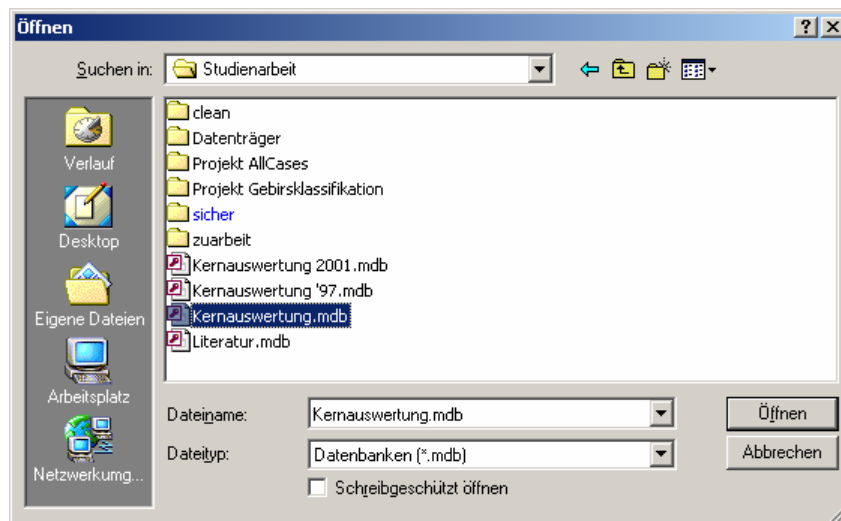


Abbildung 17: Dialog zum Öffnen von Dateien

4.2.4 Erstellen, Bearbeiten und Löschen von Bohrungen

Bevor die eigentlichen bei der Kernaufnahme gewonnenen Daten eingegeben werden können, muss erst die betreffende Bohrung in der Datenbank erstellt werden. Dies geschieht entweder über den Befehl „Bohrung einfügen“ im Menü „Bearbeiten“ oder mit der ersten Schaltfläche in der Symbolleiste für die Bohrungen (vgl. Abbildung 16, Punkt 2). Es wird ein Fenster für die Eingabe der Bohrungsdaten geöffnet, in welche die bekannten Daten eingegeben werden können (vgl. Abbildung 18). Sollten Daten - wie z.B. die exakten Koordinaten der Bohrung - nicht bekannt sein, so können die betreffenden Felder (mit Ausnahme des Namens der Bohrung) leer gelassen werden. Dieses Fenster kann auch später durch Betätigung des Menüeintrags „Eigenschaften Bohrung...“ (im Menü „Bearbeiten“) oder der zweiten Schaltfläche der Bohrungssymbolleiste angezeigt und die Einträge bearbeitet werden (Tastenkürzel: F8).

Abbildung 18: Fenster zum Eingeben der Bohrungsdaten

Über die dritte Schaltfläche der Bohrungssymbolleiste oder dem Befehl „Bohrung löschen...“ im Menü „Bearbeiten“ kann die jeweils markierte Bohrung nach einer Sicherheitsabfrage wieder gelöscht werden. Für diesen Löschvorgang müssen erst alle Kernmärsche aus der Bohrung entfernt worden sein. Falls die Liste der Bohrungen den Tastaturfokus besitzt, kann alternativ auch die Taste „Entf“ für das Löschen genutzt werden.

4.2.5 Eingeben und Bearbeiten der Bohrkerndaten

Falls eine neue Bohrung erstellt und eine Teufe größer Null angegeben wurde wird automatisch ein Assistent für die automatisierte Erstellung von Kernmärschen angezeigt (vgl. Abbildung 19). Mit Hilfe dieses Assistenten können gleichverteilt (Schritt 1) über die Bohrung Kernmärsche mit einer frei definierbaren Vorbelegung (Schritt 2) erzeugt werden. Nach Bestätigung mit der Schaltfläche „Ok“ werden die Kernmärsche hinzugefügt. Sollen keine Kernmärsche zu der Bohrung hinzugefügt werden, kann das Fenster mit dem Befehl „Abbruch“ geschlossen werden.

Abbildung 19: Assistent für das Hinzufügen von Kernmärschen

Manuell können entweder mit dem Menübefehl „Kernmarsch einfügen“ im Menü „Bearbeiten“ oder mit der ersten Schaltfläche der Kernmarschsymbolleiste (vgl. Abbildung 16, Punkt 7) Kernmärsche zu der aktuellen Bohrung hinzugefügt werden. Nach dem manuellen Hinzufügen wird das Fenster „Kernmarsch bearbeiten“ geöffnet, in dem die Daten des Kernmarsches (z.B. Teufe, Länge, Bruchstücke usw.) eingegeben werden können (vgl. Abbildung 20). Vergleichbar mit dem Bereich für Fehlermeldungen und Hinweise (vgl. Abbildung 16, Punkt 6) werden auch in diesem Fenster Hinweise bei möglicherweise fehlerhaften Daten angezeigt.

Parameter	Wert
Qu	12
Ra	5
Rb	20
Rc	-2
Rd	7
Alpha	10
Beta	8
F3	-25

Abbildung 20: Fenster für die Eingabe der Kernmarschdaten

Dieses Fenster kann über den Befehl „Eigenschaften Kernmarsch...“ im Menü „Bearbeiten“ oder mit der zweiten Schaltfläche der Kernmarschsymbolleiste angezeigt werden. Falls die Liste der Kernmärsche (vgl. Abbildung 16, Punkt 8) den Tastaturfokus besitzt, kann dieses Fenster auch mit der Eingabetaste erreicht werden.

Analog zu dem Löschen der Bohrungen könne auch die Kernmärsche mit der „Entf“-Taste gelöscht werden (falls die Liste der Kernmärsche den Tastaturfokus besitzt). Bedeutungs- gleich sind auch der Menübefehl „Kernmarsch löschen...“ und die dritte Schaltfläche der Kernmarschsymbolleiste. Das Löschen der Kernmärsche erfolgt erst nach der Bestätigung der Sicherheitsabfrage.

4.2.6 Ausgabe der Ergebnisse

Die primäre grafische Ausgabe erfolgt in dem Darstellungsbereich der aktuellen Bohrung (vgl. Abbildung 16, Punkt 5). Über die oberhalb des Darstellungsbereiches angeordnete Symbolleiste kann die Vergrößerungsstufe der Darstellung gewählt bzw. die Darstellung auf Bildschirmgröße gebracht werden.

Ganz links außen befindet sich eine Kote für die schnelle Orientierung in den Teufen und rechts davon die säulenartige Darstellung der Bohrung mit den Rohdaten. Rötlich gekennzeichnete Bereiche in der Bohrsäule weisen darauf hin, dass für diese Abschnitte keine Kernmärsche vorliegen oder eingegeben wurden. Der restliche Bereich ist der Darstellung der Kernauswerteverfahren in einem Diagramm vorbehalten.

Die für die Auswertung herangezogenen Kernauswerteverfahren können im Menü „Bearbeiten“, Unterpunkt „Verweise: Kernauswerteverfahren“ bearbeitet werden (vgl. Abbildung 21). Standardgemäß sind alle in der ActiveX-Komponente „RockClassX“ enthaltenen Verfahren vertreten - es können allerdings Verfahren ausgeschlossen werden (Verweis löschen) und/oder neue Verfahren hinzugefügt werden. Bei dem Hinzufügen neuer Verfahren müssen diese die Schnittstelle „clsClassification“ implementieren und auf dem Rechner (Angabe des Servers nicht nötig) oder einem Server im Internet bzw. Intranet (Angabe des Servers nötig) installiert sein. Der zur Nutzung nötige Verweistext sollte von dem Hersteller der jeweiligen Komponente geliefert werden; die Verweistexte der Kernauswerteverfahren der „RockClassX“ können über den Befehl „zurücksetzen“ wiederhergestellt werden.

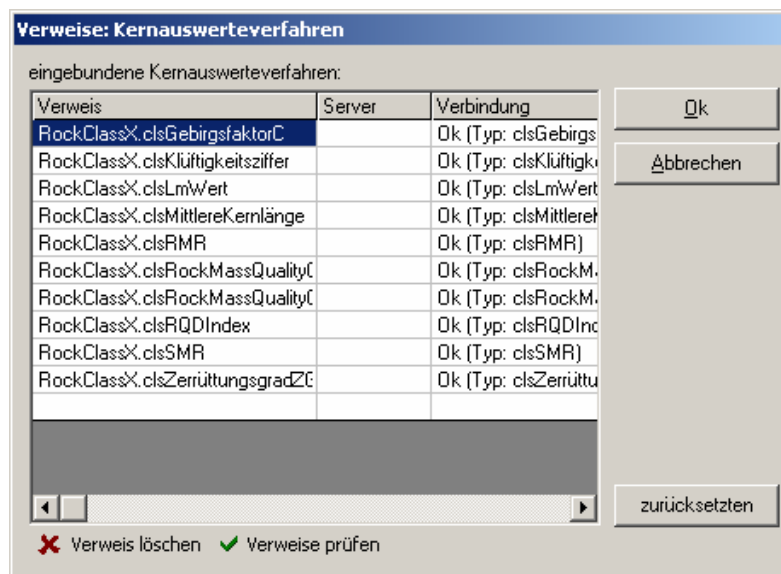


Abbildung 21: Verweise auf die Kernauswerteverfahren

Um die eingegebenen Daten weiteren, im Programm nicht vorgesehenen oder noch nicht implementierten, Analysen zugänglich zu machen, können die Daten der jeweils aktuellen Bohrung exportiert werden. Dies ist zum einen als Zeichnung (Windows Enhanced Metafile: Dateierdung WMF, Vektorgrafik) und zum anderen als Tabelle (Export nach MS Excel) möglich.

Die Exportfunktionen können im Menü „Datei“, Untermenü „Exportieren“ erreicht werden. Der Export der aktuellen Ansicht (WMF-Datei) benötigt keine Programme von Drittanbietern. Die erstellten Dateien können dann in beliebigen, das unterstützenden Programmen eingefügt werden (z.B.: Word, AutoCAD, Excel usw.). Für den Export der Daten als Tabelle nach MS Excel wird ein fehlerfrei installiertes MS Excel der Version Excel97 oder höher benötigt.

4.2.7 Beschreibung der zugrunde liegenden Datenbank

Als Dateiformat für das Programm „Gebirgsklassifikation“ wurde eine Access-Datenbank gewählt. Damit können die eingegebenen Daten in andere Programme eingebunden werden¹⁴ oder in MS Access ausgewertet, gedruckt bzw. weiterbearbeitet werden. Auch eine Weitergabe der Rohdaten an Dritte ist möglich, ohne dass diese das Programm benötigen oder installieren müssen.

Jede erstellte Datei (Datenbank) kann Daten über eine große Anzahl¹⁵ von Bohrungen aufnehmen. Jeder dieser Bohrungen wird automatisch eine Identifikationsnummer¹⁶ zugewiesen, die diese **weltweit** eindeutig identifiziert. Untergeordnet unter jede Bohrung werden die einzelnen Kernmärsche mit ihren Daten abgelegt. Neben den Bohrungen

¹⁴ Die Daten der Datenbank können in andere Programme eingebunden werden, falls diese das Verknüpfen oder Einfügen von Daten aus einer MS Access-Datenbank unterstützen. Informationen zu den Fähigkeiten der Drittsoftware und der Einbindeprozedur sollten in dem zugehörigen Handbuch oder der Hilfe-datei enthalten sein.

¹⁵ Die Maximalgröße einer Access-Datenbank beträgt 2 Gigabyte (das entspricht 2^{31} Zeichen). Die wahre Anzahl der Einträge lässt sich nicht vorherbestimmen, da sie von den unterschiedlichen Größen der Einzeleinträge abhängt. Da die Datenbank aber Verknüpfungen zu Tabellen aus anderen Dateien enthalten kann, ist die Gesamtgröße lediglich durch die verfügbare Speicherkapazität beschränkt. (Quelle: MS Access 2000 - Hilfe)

¹⁶ Diese „Replikations-ID“ (auch „Globally unique identifier“, „GUID“) ist ein 16 Byte großes Feld, welches einen Eintrag oder eine Gruppe von Einträgen weltweit einmalig definiert.
Beispiel: „{D7F26DBC-3E42-4FD8-A326-319601B3DD55}“

können in die Datenbank die zu den Kernmärschen gehörenden Abbildungen oder beliebige Objekte¹⁷ eingefügt werden (Abbildung 22).

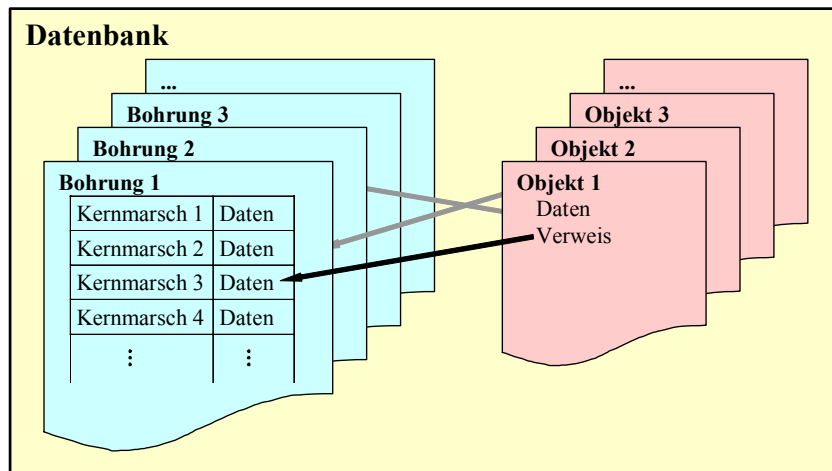


Abbildung 22: Struktur der Datenbank

Die Datenbank enthält 5 Tabellen in denen die vorgenannten Informationen abgelegt werden (vgl. Tabelle 19, unten). Die Anlage „Anlage V: Definitionen der Tabellen der zugrunde liegenden Datenbank“ zeigt die exakte Definition der in der Datenbank verwendeten Tabellen. Die Tabelle „Bohrungen“ enthält als Einstiegspunkt eine Liste aller in der Datenbank verankerten Bohrungen mit zusätzlichen Informationen wie z.B. Durchmesser, Teufe und Koordinaten der Bohrung. Mit der Identifikationsnummer im „BohrungID“-Feld können alle zu einer Bohrung gehörenden Kernmärsche aus der Tabelle „Kernmärsche“ extrahiert werden. Gleiches gilt für das „KernmarschID“-Feld in der „Objekte“-Tabelle. Die beiden restlichen Tabellen „Bohrverfahren“ und „Bohrwerkzeug“ sind Hilfstabellen und werden von der Tabelle „Bohrungen“ als Hilfe zum schnelleren Nachschlagen und Eingeben der gleichnamigen Felder genutzt.

Tabelle 19 : Tabellen der Datenbank

Tabelle	Beschreibung
Bohrungen	Tabelle aller Bohrungen mit zusätzlichen Informationen
Kernmärsche	Tabelle aller Kernmärsche mit deren Daten und Bezug auf je eine Bohrung
Objekte	enthält die Abbildungen bzw. OLE-Daten zu den einzelnen Kernmärschen
Bohrverfahren	Liste mit Bohrverfahren zur schnellen Auswahl (ausgeblendet)
Bohrwerkzeuge	Liste mit Bohrwerkzeugen zur schnellen Auswahl (ausgeblendet)

¹⁷ Ein Objekt ist eine Einheit von Daten, die durch eine Gruppe von Attributen beschrieben wird. Im Sinne der Datenverarbeitung ist ein „Ball“ ein Objekt mit Attributen wie z.B. „Farbe“, „Temperatur“, „Gewicht“. Im vorliegenden Fall können Objekte beliebige Daten wie z.B. Bilder, Videos, Texte, Tabellen oder Audiodaten enthalten.

Den Tabellen übergeordnet sind die Abfragen. Sie enthalten keine eigenen Daten, sondern greifen auf die oben beschriebenen Tabellen zu und bereiten diese auf. Sie werden in der Regel für Auswahllisten (bessere Bedienbarkeit der Datenbank und der mit der Datenbank arbeitenden Programme) und Berichte genutzt. In Abbildung 23 sind die Tabellen, Abfragen und Beziehungen als Graph zusammengefasst dargestellt.

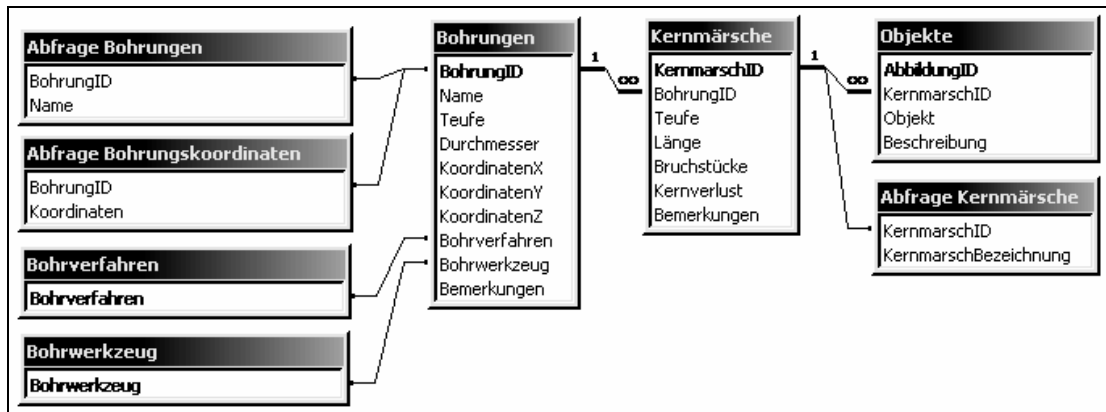


Abbildung 23: Beziehungen der Datenbank

4.3 Beschreibung des ActiveX-Projektes „RockClassX“

4.3.1 Erläuterung des Konzeptes

Um die geschriebene Software möglichst anwenderfreundlich zu gestalten, die Fehlersuche und eine mögliche spätere Erweiterung zu erleichtern, wurde ein mehrschichtiges Programmiermodell gewählt (Abbildung 24).

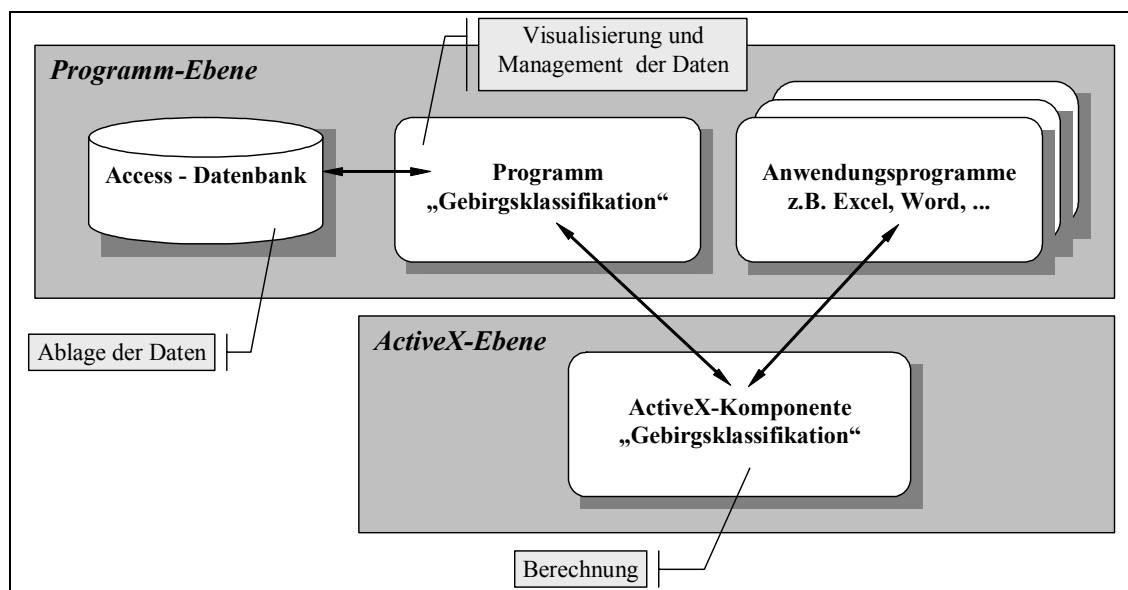


Abbildung 24: Mehrschichtiges Programmiermodell

Das Projekt wurde in die ActiveX-Ebene¹⁸ und die Programm-Ebene aufgespalten. Die ActiveX-Komponente besitzt keine Fenster, in die Daten eingegeben werden können, sondern kann nur von anderer Software über klar definierte Kommunikationsschnittstellen erreicht werden (vgl. Abschnitt 4.3.4: „Beschreibung der Schnittstelle“, Seite 70). Für sich allein genommen ist die ActiveX-Komponente damit nutzlos. Ihre Aufgabe ist es, von anderen Programmen gelieferte Daten zu verarbeiten und diese Ergebnisse zurückzugeben. Das Programm „Gebirgsklassifikation“ nutzt die Komponente standardgemäß und für den Nutzer unsichtbar für alle Berechnungen der Klassifikationsverfahren. Anwendungen von Drittherstellern können die Komponente ebenfalls nutzen, falls sie programmierbar sind (vgl. Abschnitt 4.1: Systemvoraussetzungen, Seite 57). Am Beispiel von MS Excel wird in Abschnitt 4.3.3 die Prozedur für das Einbinden in eine Anwendungssoftware gezeigt.

4.3.2 Installation der Komponente

Bei der Installation des Programms „Gebirgsklassifikation“ wird automatisch auch die ActiveX-Komponente „RockClassX“ installiert und braucht nicht gesondert nochmals installiert werden.

Zur gesonderten Installation der ActiveX-Komponente ist die Datei „Setup.exe“ im Verzeichnis „\EDV-gerechte Umsetzung\Install\RockClassX“ zu starten. Unter den Betriebssystemen WindowsNT, Windows2000 oder höher sind ausreichende Rechte¹⁹ zur Installation erforderlich. Nach dem Start der Installationsroutine muss nach einer Bestätigung der Installation ein Installationsordner angegeben werden. Nach erfolgreichem Installationsvorgang wird eine Meldung ausgegeben, die den Erfolg oder Misserfolg dokumentiert. Im Falle eines Erfolges steht die Komponente nun für jeden Nutzer des Rechners zur Verfügung.

4.3.3 Einbinden der ActiveX-DLL in Anwendungsprogramme am Beispiel von MS Excel

In der auf dem beigelegten Datenträger enthaltenen Microsoft Excel 2000 - Datei „Gebirgsklassifikation.xls“ (vgl. Abbildung 25, Seite 68) wurde die ActiveX-Komponente

¹⁸ ActiveX ist der Oberbegriff für eine Reihe von Technologien, mit denen Softwarekomponenten in einer vernetzten Umgebung ungeachtet der Sprache, in der die Komponenten erstellt wurden, interagieren können.

¹⁹ Bei der Installation wird die Komponente am System angemeldet und falls nötig Systemteile aktualisiert oder hinzugefügt. Für diesen Vorgang werden in der Regel Administratorrechte benötigt.

„RockClassX“ zu Demonstrationszwecken eingebunden. Im Folgenden wird beschrieben, welche Maßnahmen für eine Nutzung notwendig sind. In der Datei „Gebirgsklassifikation.xls“ sind diese Punkte bereits abgearbeitet und können an ihr nachvollzogen werden.

Demonstration der Schnittstelle RockClassX												
Integration der ActiveX-Komponente in Microsoft Excel												
Bohrung	Kernmarsch			Bruchstücke	RQD-Index				L _m -Wert			
	von [cm]	bis [cm]	Länge [cm]		Wert	Klasse			Wert [cm]	Klasse		
KB1	Bohrung bei den Koordinaten 1232x3512											
	0	100	100	5/3/4/9/8/11/1/1/1/1/1	11,00	1	sehr gering (very poor)			4,45	1	sehr gering
	100	200	100	5/8/12/6/7/21	33,00	2	gering (poor)			8,95	2	gering - mittel
	200	300	100	25/16/23/21	85,00	4	gut (good)			18,70	4	gut - sehr gut
	300	400	100	15/9/23/27	65,00	3	mittel (fair)			15,80	4	gut - sehr gut
	400	500	100	35/30/33	98,00	5	sehr gut (excellent)			30,40	4	gut - sehr gut
	500	600	100	77/23	100,00	5	sehr gut (excellent)			64,85	4	gut - sehr gut
	600	700	100	100	100,00	5	sehr gut (excellent)			97,50	4	gut - sehr gut
	700	800	100	15/9/23/51	89,00	4	gut (good)			34,55	4	gut - sehr gut
	800	900	100	15/23/30	68,00	3	mittel (fair)			16,10	4	gut - sehr gut
	900	1.000	100	15/23/6	38,00	2	gering (poor)			8,90	2	gering - mittel
	Gesamtiefe 1.000			Mittelwert:	68,70	3	mittel (fair)			30,02	4	gut - sehr gut
				Minimalwert:	11,00	1	sehr gering (very poor)			4,45	1	sehr gering
				Maximalwert:	100,00	5	sehr gut (excellent)			97,50	4	gut - sehr gut

Abbildung 25: Excel-Tabelle „Gebirgsklassifikation.xls“

Zur Nutzung der Funktionen der Komponente ist es nötig, ein kurzes Skript in der MS Excel eigenen Programmiersprache (Visual Basic für Applikationen, VBA) zu schreiben. Das Skript dient als „Mittler“ zwischen der ActiveX-Komponente und Excel, da ein direkter Aufruf aus der Excel-Tabelle nicht möglich ist.

Um nicht bei jeder neuen MS Excel-Tabelle das Skript neu schreiben zu müssen und um eventuelle Fehler bei der Eingabe zu vermeiden, wurde ein Excel - Add-In geschrieben und auf dem Datenträger beigelegt (Datei „RockClassX.xla“). Dieses Add-In enthält einen vollständigen Satz von Mittlerfunktionen, um die Kernauswerteverfahren der Komponente RockClassX nutzen zu können. In Tabelle 20 wurden alle Funktionen aufgelistet, die zur Verfügung gestellt werden. Der Aufruf der Funktionen erfolgt nach einbinden des Add-In's (Menü „Extras“ im Untermenü „Add-Ins...“, nähere Informationen siehe Hilfe von Excel), wie in dem in Abbildung 26 gezeigten Beispiel.

	A	B	C	D	E	F
1			Berechnung des L_m-Wertes:			
2			Kernmarschlänge:	100 cm		
3			Bruchstücke:	18A/12B/25A/8A		
4			L _m Wert:	11,8		
5						
6						

Formel:
= LmWert(D2; D3)

Abbildung 26: Aufruf der Funktionen des Add-In's RockClassX

Tabelle 20: Funktionen des MS Excel - Add-In's RockClassX

I		II	
Funktion Parameter	Beschreibung	Funktion Parameter	Beschreibung
GetCaseCount BruchAlle As Long	Berechnet die Anzahl der möglichen Bruchstückkombinationen für einen 1 m langen Kemmarsch, wenn ein Bruch in diskreten Abständen auftreten kann	MittlereKernlänge Kemmarsch As Double, Pieces As String	Berechnet die mittlere Kernlänge
GebirgsfaktorC Kemmarsch As Double, Pieces As String, SampleCount As Long, SampleHeight As Double	Berechnet den Gebirgsfaktor C	MittlereKernlängeClassDesignation Value As Double	Gibt die Beschreibung für eine berechnete mittlere Kernlänge zurück
GebirgsfaktorCC Value As Double	Gibt die Beschreibung für einen berechneten Gebirgsfaktor C zurück	MittlereKernlängeClassIndex Value As Double	Gibt die Index der Klasse für eine berechnete mittlere Kernlänge zurück
GebirgsfaktorCCSampleCountFromPieces Pieces As String, SampleHeight As Double	Gibt die Index der Klasse für einen berechneten Gebirgsfaktor C zurück	RMR Kemmarsch As Double, Pieces As String, Qu As Double, Ra As Double, Rb As Double, Rc As Double, Rd As Double	Berechnet den RMR
Klüftigkeitsziffer Kemmarsch As Double, Pieces As String	Berechnet die Anzahl der max. möglichen Probenkörper für gegebene Bruchstücke	RMRClassDesignation Value As Double	Gibt die Beschreibung für einen berechneten RMR zurück
KlüftigkeitszifferClassDesignation Value As Double	Berechnet die Klüftigkeitsziffer k	RMRClassIndex Value As Double	Gibt die Index der Klasse für einen berechneten RMR zurück
KlüftigkeitszifferClassIndex Value As Double	Gibt die Beschreibung für eine berechnete Klüftigkeitsziffer zurück	RockMassQualityQ Kemmarsch As Double, Pieces As String, Jn As Double, Jr As Double, Ja As Double, Jw As Double, SRF As Double	Berechnet die Rock Mass Quality Q
LmWert Kemmarsch As Double, Pieces As String, Optional Klassengröße As Double = 5	Gibt die Beschreibung für einen berechneten L _m Wert zurück	RockMassQualityQC Value As Double	Gibt die Beschreibung für eine berechnete Rock Mass Quality Q zurück
LmWertClassDesignation Value As Double	Gibt die Beschreibung für einen berechneten L _m Wert zurück	RockMassQualityQCClassIndex Value As Double	Gibt die Index der Klasse für eine berechnete Rock Mass Quality Q zurück
LmWertClassIndex Value As Double	Gibt die Index der Klasse für einen berechneten L _m Wert zurück	RQDIndex Kemmarsch As Double, Pieces As String	Berechnet den RQD-Index
		RQDIndexClassDesignation Value As Double	Gibt die Beschreibung für einen berechneten RQD-Index zurück
		RQDIndexClassIndex Value As Double	Gibt die Index der Klasse für einen berechneten RQD-Index zurück

Tabelle 20: Funktionen des MS Excel - Add-In's RockClassX (Fortsetzung)

III	
Funktion Parameter	Beschreibung
RockMassQualityQTBM Kernmarsch As Double, Pieces As String, Jn As Double, Jr As Double, Ja As Double, Jw As Double, SRF As Double, sigma As Double, f As Double, cli As Double, q As Double, s As Double	Berechnet die Rock Mass Quality Q_{TBM}
RockMassQualityQTBMClassDesignation Value As Double	Gibt die Beschreibung für eine berechnete Rock Mass Quality Q_{TBM} zurück
RockMassQualityQTBMClassIndex Value As Double	Gibt die Index der Klasse für eine berechnete Rock Mass Quality Q_{TBM} zurück
SMR Kernmarsch As Double, Pieces As String, Qu As Double, Ra As Double, Rb As Double, Rc As Double, Rd As Double, F1 As Double, F2 As Double, F3 As Double, F4 As Double	Berechnet den SMR
SMRClassDesignation Value As Double	Gibt die Beschreibung für einen berechneten SMR zurück
SMRClassIndex Value As Double	Gibt die Index der Klasse für einen berechneten SMR zurück
SMRCalculateF1 AlphaRAD As Double	Berechnet den Parameter F1 über den Winkel Alpha (in RAD)
SMRCalculateF2 BetaRAD As Double	Berechnet den Parameter F2 über den Winkel Beta (in RAD)
ZG Kernmarsch As Double, Pieces As String	Berechnet den Zerrüttungsgrad Z_G
ZGClassDesignation Value As Double	Gibt die Beschreibung für einen berechneten Zerrüttungsgrad Z_G zurück
ZGClassIndex Value As Double	Gibt die Index der Klasse für einen berechneten Zerrüttungsgrad Z_G zurück

Sollte der Einsatz des Add-In's nicht möglich sein, nicht gewünscht werden oder weitere (nicht in der hier vorgestellten Version der Komponente RockClassX enthaltene) Verfahren eingebunden werden, wird in Anlage VII: „Manuelles Einbinden der Komponente RockClassX“ die Vorgehensweise beschrieben.

Hinweis: Unter Umständen kann es dazu kommen, dass trotz des korrekten Einbindens des Add-In's die Funktionen nicht erkannt werden. In diesem Fall muss in dem Visual Basic-Editor noch zusätzlich der Verweis auf das beschriebene Add-In gesetzt werden. (vgl. Anlage VII: „Manuelles Einbinden der Komponente RockClassX“)

4.3.4 Beschreibung der Schnittstelle clsClassification

Innerhalb der ActiveX-Komponente „RockClassX“ verwenden einheitlich alle Klassifikationssysteme die in der Klasse „clsClassification“ definierte Schnittstelle. Damit wird zum einen die Implementierung und Verwendung der einzelnen Klassifikationssysteme vereinfacht und vereinheitlicht, zum anderen können so relativ schnell neue Klassifikationssysteme erstellt und eingebunden werden (diese müssen nur die hier beschriebene Schnittstelle bedienen).

Der Begriff „Klasse“ ist an dieser Stelle nicht mit dem der Klasse in einem der Klassifikationssysteme zu verwechseln. In der objektorientierten Programmierung ist eine „Klasse“ eine Schablone oder Vorlage, die die Merkmale eines Objekts definiert und Aussehen und Verhalten des auf ihr basierenden Objekts beschreibt. Ähnlich wie in PLATONS Ideenlehre stellt eine Klasse die „ideale Vorstellung“ eines Gegenstandes dar (z.B.: Segelschiff). Aufgrund dieser Vorstellung können nun beliebig viele „reale Gegenstände“ erstellt werden, die dieser Vorstellung entsprechen (z.B. Schiffe von CHRISTOPH KOLUMBUS „Santa María“, „Niña“, „Pinta“ usw.).

Um die Schnittstelle zu nutzen, muss in der Entwicklungsumgebung eine neue Klasse erstellt und diese mit der Schnittstelle verbunden werden. Im Visual Basic wird diese Verbindung mit folgender Quellcodezeile hergestellt (eingetragen im Deklarationsbereich der neuen Klasse):

```
Implements clsClassification
```

Abbildung 27: Implementierung der „clsClassification“

Die neue Klasse besitzt nun die gleichen Prozeduren und Eigenschaften wie die Ursprungsklasse, welche nun noch gemäß den Anforderungen der einzubindenden Klassifikation zu besetzen sind. In Tabelle 21 sind alle von der Klasse „clsClassification“ zur Verfügung gestellten Prozeduren und Eigenschaften aufgelistet.

Tabelle 21: Prozeduren der Klasse „clsClassification“

Prozedurtyp	Name Parameter	Rückgabewert	Beschreibung
Function	Calculate Parameter As Scripting.Dictionary	Double	Berechnet den eigentlichen Zahlenwert der Klassifikation
Function	ClassDesignation ByVal Value As Double	String	Benennung der Klasse des angegebenen Wertes
Function	ClassDesignationByIndex ByVal Index As Long	String	Index der Klasse des angegebenen Wertes
Function	ClassIndexByValue ByVal Value As Double	Long	Index der Klasse, in der der Wert Value liegt
Property Get	ClassCount	Long	Anzahl der Klassen der Klassifikation
Property Get	ClassificationName	String	Bezeichnung der Klassifikation
Property Get	ClassificationAutorYear	String	Autor und Jahr der Entwicklung
Property Get	ValueRangeMin	Double	Untergrenze des Wertebereiches
Property Get	ValueRangeMax Berechnungslänge As Double	Double	Obergrenze des Wertebereiches
Property Get	Unit	String	Einheit der zurückgegebenen Werte
Property Get	FormatString	String	Zahlenformat der Klassifikation

5 Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein Überblick über die wichtigsten quantitativen Kernauswerteverfahren gegeben, der Versuch einer Gebirgsklassifikation nach dem L_m -Wert unternommen, der Grundstein für eine offene Datenverwaltung, -verarbeitung und -ablage von Kernaufnahmen gelegt und verschiedene Auswerteverfahren EDV-gerecht aufgearbeitet.

Bei der Analyse verschiedener, bereits in der Ingenieurgeologie etablierter Auswerteverfahren, wurden ausgehend von der allgemeinen Kernaufnahme mit zusätzlicher Apposition von Erhaltungsklassen, der RQD-Index, die Klüftigkeitsziffer k , die mittlere Kernlänge \bar{k} , das RMR-System, der Gebirgsfaktor C , der L_m -Wert, die Rock Mass Quality Q , der Zerrüttungsgrad Z_G und das SMR-System mit allen benötigten Parametern vorgestellt und anhand von theoretischen Betrachtungen und Fallbeispielen verglichen. Durch diese Gegenüberstellungen konnte nachgewiesen werden, dass die Klüftigkeitsziffer k und der Zerrüttungsgrad Z_G die ungünstigsten Korrelationsverhalten gegenüber den restlichen Verfahren besitzen und somit bei einer Nutzung zur Vorsicht geraten werden muss. Für den L_m -Wert hingegen konnte eine besonders gute Korrelation nachgewiesen werden. Er konnte als besonders „sensibel“ eingestuft werden. Zum L_m -Wert wurden Ergänzungen vorgenommen und die Substitution des RQD-Indexes durch den L_m -Wert diskutiert. Ungeklärt bleiben der Ersatz des RQD-Indexes bei anderen Kernauswerteverfahren.

Im Rahmen der EDV-gerechten Aufarbeitung der Auswerteverfahren wurde ein offener Datenbankstandard für die Speicherung von Aufnahmedaten vorgeschlagen und ein darauf aufbauendes erweiterbares Programmsystem entwickelt. Außerdem wurde die Möglichkeit einer Nutzung der Auswerteverfahren in beliebigen Anwendungen von Drittherstellern geschaffen.

Die Entwicklung der Kernauswerteverfahren hat seit der Mitte des 20. Jahrhunderts ausgehend von den beschreibenden, qualitativen Verfahren große Fortschritte erzielt. Ein Ende der weiter Entwicklung auf diesem Gebiet ist aus heutiger Sicht noch nicht abzusehen, was auch auf die heute noch vorhandenen Schwächen und Einschränkungen der einzelnen Verfahren zurückzuführen ist.

In der Praxis sollte nicht nur ein Verfahren pro Projekt genutzt werden. Es sollten mindestens zwei Verfahren zum Einsatz kommen und deren Ergebnisse kritisch geprüft und verglichen werden. Dadurch können mögliche Fehlerquellen sowohl in den Eingangsdaten als auch in den Ergebnissen eingegrenzt und vermindert werden.

6 Verwendete Variablen

Tabelle 22: Verwendete Variablen

Variable	Einheit	Beschreibung
c	[kN/m ²], [kPa]	Kohäsion
d, m	[m]	Abstand, Längenmaß
k	[m]	Klassenlänge
L	[m]	Länge der Kernstrecke parallel zur Bohrlochlängsachse
n, i	-	Anzahl oder Ordnungsnummer
q_u	[kN/m ²]	einaxial ermittelte Gesteinsdruckfestigkeit
D	[m], [cm], [mm]	Durchmesser des Bohrkernes
S	[m]	Länge der Berechnungseinheit
x	[m], [cm]	Kernbruchstücke
u	[kN/mm ²]	Wasserdruck
α, β	[°]	Winkel
ϕ	[°], [g]	Reibungswinkel
σ	[kN/m ²]	Spannung
σ_1, σ_3	[kN/m ²]	größte, kleinste Hauptnormalspannung
σ_t	[kN/m ²]	Zugfestigkeit
σ_c	[kN/m ²]	Druckfestigkeit ohne Seitendruck

7 Literaturverzeichnis

- [1] BARTON, N.: Recent experiences with the Q-system of tunnel support design - Proceedings of the Symposium on Exploration for Rock Engineering, Johannesburg, November 1976
- [2] BARTON, N.; GRIMSTAD, E.: The Q-System following Twenty Years of Application in NMT Support Selection - Felsbau 12 (1994) Nr. 6, Verlag Glückauf GmbH
- [3] BARTON, N.; JHUN: TBM Performance estimation in rock using Q_{TBM} - SR005.pdf
- [4] BARTON, N.; LIEN, R.; LUNDE, J.: Engineering Classification of Rock Masses for the Design of Tunnel Support - Rock Mechanics 6, Springer-Verlag, 1974
- [5] BIENIAWSKI, Z. T.: Engineering Classification Of Jointed Rock Masses - The Civil Engineer in South Africa, December 1973
- [6] BIENIAWSKI, Z. T.: Engineering Rock Mass Classifications - John Wiley & Sons, Inc. 1989
- [7] BIENIAWSKI, Z. T.: Rock mass classification in rock engineering - Proceedings of the Symposium on Exploration for Rock Engineering, Johannesburg, November 1976
- [8] CUMMINGS, R.A., KENDORSKI, F.S. and BIENIAWSKI, Z.T.: Caving rock mass classification and support estimation - U.S. Bureau of Mines Contract Report #J0100103. Chicago: Engineers International Inc., 1982
- [9] DEERE, D. U.: Technical Description of Rock Cores for Engineering Purposes - Rock Mech. Eng. Geol. 1, 1963, S. 16-22.
- [10] DEERE, D. U.; DEERE, D. W.: The rock Quality Designation (RQD) Index in Practice - Proc. Symp. Rock Classification for Engineering Purposes, Cincinnati (ASTM Special Technical Publication 984, Philadelphia), 1988
- [11] DEPARTMENT OF MINERALS AND ENERGY, WESTERN AUSTRALIA: Technical Update No.1: The "Q-System" geotechnical design method was updated in 1994 - Mining Operation Division Mineral House, Perth
- [12] DIN 4022, Teil 1 September 1987, Teil 2 März 1981, Teil 3 Mai 1982
- [13] DITTRICH, E.: Die Ermittlung ingenieurgeologisch bedeutsamer Gefügeparameter an Hand von Bohrkernen - Z. geol. Wiss. Berlin 1978 Band 6; S. 329-340
- [14] EICHLER, K.: Fels- und Tunnelbau - expert-Verlag 2000
- [15] HANSÁGI, I.: A Method of Determining the Degree of Fissuration of Rock - Int. J. Rock Mech. Min. Sci. & Geomech. Abstr. Vol. 11, Pergamon Press 1974

- [16] HANSÁGI, I.: Bergmännisches Verfahren der Gebirgsfestigkeitsbestimmung und der Gebirgsklassifizierung - Proc. 1. Congr. Int. Soc. Rock Mech. 1, Lissabon 1967
- [17] HOEK, E.: Practical Rock Engineering - University of Toronto, 1998,
<http://bh.knu.ac.kr/~gyokim/lectnote.htm>
- [18] HUDSON, J. A.: Comprehensive rock engineering: principles, practice and projects (Volume 3: Rock testing and site characterization) - Pergamon Press Ltd. 1993
- [19] HUDSON, J. A.: Comprehensive rock engineering: principles, practice and projects (Volume 5, Surface and underground case histories) - Pergamon Press Ltd. 1993
- [20] JOHN, M.: Zielsetzung der Gebirgsklassifizierung - Felsbau 12 (1994) Nr. 6, Verlag Glückauf GmbH
- [21] KAISER, P. K.; MACKAY, C.; GALE, A. D.: Evaluation of Rock Classification at B. C. Rail Tumbler Ridge Tunnels - Rock Mechanics and Rock Engineering, Springer-Verlag, 1986
- [22] LAUBSCHER, D. H.: "Design Aspects and Effectiveness of Support Systems in Different Mining Conditions." Trans. Inst. Min. Metall. 93, S. A70-A81, 1984
- [23] LAUBSCHER, D. H.: "Geomechanics Classification of Jointed Rock Masses - Mining Applications." Trans. Instn. Min. Metall. 86, S. A-1-A-7, 1977
- [24] LAUBSCHER, D. H.; TAYLOR, H. W.: The importance of geomechanics classification of jointed rock masses in mining operations - Proceedings of the Symposium on Exploration for Rock Engineering, Johannesburg, November 1976
- [25] LAUBSCHER, D. M. and PAGE, C. H.: The design of rock support in high stress or weak rock environments. Proc. 92nd Can. Inst. Min. Metall. AGM, Ottawa, Paper # 91, 1990
- [26] LAUFFER, H.: Gebirgsklassifizierung für den Stollenbau - Geologie und Bauwesen 24 (1958), S 46-51
- [27] MAIDL: Handbuch des Tunnel- und Stollenbaus - Verlag Glückauf GmbH Essen 1995, Band II, 2. Auflage
- [28] MEIER, G.: Ein repräsentatives Verfahren zur ingenieurgeologischen Bohrkernausswertung im Fels in Altbergbaugebieten - Berichte 12. Nat. Tagung f. Ing.-Geol. Halle 1999
- [29] MEIER, G.: Grundsätze von Bergsicherungsarbeiten im Gangbau - WTI Jahrgang 32/1991 Reihe A Heft 1
- [30] MEIER, G.: Ingenieurgeologische Geländeuntersuchungen über repräsentative Ermittlungs- und Darstellungsmethoden strukturgeologischer Parameter zur

- Bewertung der hydraulischen Gebirgsverhältnisse und der Injizierbarkeit von klüftigem Gebirge - BA Freiberg Diss 1974
- [31] MÖBUS, G.: Methodik der tektonischen Bearbeitung des Kernmaterials von Bohrungen, Freiburger Forschungshefte, C 372 Leipzig 1982
- [32] MÖBUS, G.: Zur Bearbeitung der Klüftigkeit des Kernmaterials von Bohrungen - Zeitschrift für angewandte Geologie, Bd. 20 (1974) Heft 8
- [33] MOLEK, H.: Aussagemöglichkeiten strukturgeologischer Bohrkernuntersuchungsmethoden in Subrosionsgebieten - Zeitschrift für angewandte Geologie, Bd. 29 (1983) Heft 10
- [34] MOLEK, H.: Ingenieurgeologische Untersuchungen in Subrosionsgebieten unter besonderer Berücksichtigung der Ermittlung des Zerrüttungsgrades im postsalinaren Deckgebirge - BA Freiberg Diss. 1984
- [35] OLIVER, H. J.: Determination of RQD from petroscope observations - Proceedings of the Symposium on Exploration for Rock Engineering, Johannesburg, Nov. 1976
- [36] PRINZ, H.: Abriss der Ingenieurgeologie, 2. Auflage
- [37] ROMANA, M.: New adjustment ratings for application of Bieniawski classification to slopes. Proc. Int. Symp. on the Role of Rock Mech. S. 49-53, Zacatecas, 1985
- [38] ROMANA, M.: Practice of SMR classification for slope appraisal - Proceedings 5th International Symposium on Landsides, Lausanne, Balkema, Rotterdam (1988)
- [39] SCHNEIDER, T. R.: Vergleich der Ausbruchsklassen im Sicherheitsstollen und Straßentunnel, Los Nord Gotthard-Straßentunnel - Schweizer Ingenieur und Architekt 98 (1980)
- [40] TERZAGHI, K.: Rock defects and loads on tunnel supports - Rock Tunneling with Steel Supports, eds PROCTOR, R. V. and WHITW, T., Commercial Shearing and Stamping Co, Youngstown (1946), S 15-99

8 Anlagenverzeichnis

- Anlage I: Eingangswerte Rock Mass Quality Q
- Anlage II: Rock Mass Rating System (nach BIENIAWSKI [6])
- Anlage III: Nomogramm der Beziehung zwischen RQD-Index und Gebirgsfaktor C
- Anlage IV: Datenblatt Programm und ActiveX-DLL
- Anlage V: Definitionen der Tabellen der zugrunde liegenden Datenbank
- Anlage VI: Quellcode des Programms und der ActiveX-Komponente
- Anlage VII: Manuelles Einbinden der Komponente RockClassX
- Anlage VIII: Code für Generierung aller Bruchstückkombinationen eines Kernmarsches
- Anlage IX: Anzahl möglicher Kernbruchstückkombinationen
- Anlage X: Korrelation L_m -Wert, RQD-Index und Zerrüttungsgrad Z_G
- Anlage XI: Vergleich der Auswerteverfahren (Diagramme)
- Anlage XII: Vergleich der Klassen der Auswerteverfahren mit den Klassen des L_m -Wertes
- Anlage XIII: Bibliographie

9 Abbildungs-, Tabellen und Gleichungsverzeichnis

Abbildungsverzeichnis

Abbildung 1: Einsatzgebiete von Kernauswerteverfahren	7
Abbildung 2: Klassifikation der Auswerteverfahren nach qualitativen und quantitativen Aspekten	8
Abbildung 3: Klassifikation nach technologie- und gebirgsbezogenen Aspekten	9
Abbildung 4: Beispielaufnahme eines Bohrkernes	14
Abbildung 5: Korrektur des Schnitteffektes	18
Abbildung 6: Gebirgsklassen nach BIENIAWSKI im LAUFFER-Diagramm [6]	21
Abbildung 7: Schema für die Bestimmung des MRMR [6]	22
Abbildung 8: Korrelation RQD-Index und Zerrüttungsgrad Z_G	38
Abbildung 9: Korrelation RQD-Index und Gebirgsfaktor C nach HANSÁGI [34]	39
Abbildung 10: Nomogramm der Beziehung zwischen RQD-Index und Gebirgsfaktor C ..	40
Abbildung 11: Vergleich zwischen L_m -Wert und Grenzfall des L_m -Wertes	48
Abbildung 12: Korrelation L_m -Wert und RQD-Index	49
Abbildung 13: Übergang zwischen zwei Isoperimeren	50
Abbildung 14: Korrelation L_m -Wert und Zerrüttungsgrad Z_G	52
Abbildung 15: Abbildung der Klassen der Verfahren RQD-Index und Gebirgsfaktor C auf die Klassen des L_m -Wertes	54
Abbildung 16: Oberfläche des Programmes	59
Abbildung 17: Dialog zum Öffnen von Dateien	60
Abbildung 18: Fenster zum Eingeben der Bohrungsdaten	61
Abbildung 19: Assistent für das Hinzufügen von Kernmärschen	61
Abbildung 20: Fenster für die Eingabe der Kernmarschdaten	62
Abbildung 21: Verweise auf die Kernauswerteverfahren	63
Abbildung 22: Struktur der Datenbank	65
Abbildung 23: Beziehungen der Datenbank	66
Abbildung 24: Mehrschichtiges Programmiermodell	66
Abbildung 25: Excel-Tabelle „Gebirgsklassifikation.xls“	68
Abbildung 26: Aufruf der Funktionen des Add-In's RockClassX	68
Abbildung 27: Implementierung der „clsClassification“	71

Tabellenverzeichnis

Tabelle 1: Zusammenfassung der Gebirgsklassifikationsverfahren	10
Tabelle 2: erweiterte Tabelle nach DIN 4022 Teil 2 [12]	14
Tabelle 3: Einteilung in Klassen - RQD-Index	15
Tabelle 4: Einteilung des RQD-Indexes in Unterklassen	16
Tabelle 5: Einteilung in Klassen - Klüftigkeitsziffer k	17
Tabelle 6: Einteilung in Klassen - RMR-System	20
Tabelle 7: Einteilung in Klassen - Gebirgsfaktor C	23
Tabelle 8: Einteilung in Klassen - L_m -Wert	25
Tabelle 9: Einteilung in Klassen - Rock Mass Quality Q	27
Tabelle 10: Änderungen am System der Rock Mass Quality Q	29
Tabelle 11: Einteilung in Klassen - Zerrüttungsgrad Z_G	31
Tabelle 12: Einteilung in Klassen - SMR-System	32
Tabelle 13: Vergleich der Auswerteverfahren	34
Tabelle 14: Skalierbarkeit der Verfahren	35
Tabelle 15: Vor- und Nachteile der Verfahren	42
Tabelle 16: Orte der in die Datenbasis aufgenommenen Bohrkern	43
Tabelle 17: Für die Berechnung der Verfahren benötigte Parameter	44
Tabelle 18: Modifizierte Einteilung in Klassen - L_m -Wert	55
Tabelle 19 : Tabellen der Datenbank	65
Tabelle 20: Funktionen des MS Excel - Add-In's RockClassX	69
Tabelle 21: Prozeduren der Klasse „clsClassification“	71
Tabelle 22: Verwendete Variablen	73

Gleichungsverzeichnis

Gleichung 1: RQD-Index	15
Gleichung 2: Zusammenhang zwischen dem RQD-Index und J_v	16
Gleichung 3: Klüftigkeitsziffer k	17
Gleichung 4: Korrektur des Schnitteffektes	18
Gleichung 5: mittlere Kernlänge \bar{k}	18
Gleichung 6: RMR-System	19
Gleichung 7: Gebirgsfaktor C	23
Gleichung 8: Gebirgsfaktor C ; Probenfaktor	23
Gleichung 9: Gebirgsfaktor C ; Klüftigkeitsfaktor	23

Gleichung 10: Gebirgsfaktor C (zusammengefügte Gleichung).....	23
Gleichung 11: L_m -Wert.....	24
Gleichung 12: L_m -Wert; größte Klasse.....	25
Gleichung 13: Rock Mass Quality Q.....	26
Gleichung 14: Annäherungsgleichung für die Scherspannungen zwischen Klüftkörpern..	26
Gleichung 15: Ausbaudruck im Firstbereich.....	28
Gleichung 16: Rock Mass Quality Q_{TBM}	29
Gleichung 17: Zerrüttungsgrad Z_G	30
Gleichung 18: Abminderungsfaktor.....	30
Gleichung 19: SMR-System.....	31
Gleichung 20: Vereinfachte Formel für den Zerrüttungsgrad Z_G	37
Gleichung 21: verallgemeinerte Formel für Abminderungsfaktor f	37
Gleichung 22: erste Annahme für Gleichung 21.....	37
Gleichung 23: zweite Annahme für Gleichung 21.....	37
Gleichung 24: Vereinfachte Formel für Z_G in Abhängigkeit vom RQD-Index.....	37
Gleichung 25: Annahme für Gleichung 24.....	37
Gleichung 26: nach dem RQD-Index umgestellte Gleichung 24.....	38
Gleichung 27: nach dem RQD-Index umgestellte Gleichung 24.....	39
Gleichung 28: Korrelation zwischen RMR-System und Rock Mass Quality Q nach [7]..	41
Gleichung 29: Korrelation zwischen RMR-System und Rock Mass Quality Q.....	46
Gleichung 30: L_m -Wert mit fiktiver Klassengröße Null.....	47
Gleichung 31: Pfade gleicher Kernbruchstückzahlen mit gleicher Länge.....	49
Gleichung 32: Obergrenze des L_m -Wertes in Abhängigkeit vom RQD-Index.....	51
Gleichung 33: Untergrenze des L_m -Wertes in Abhängigkeit vom RQD-Index.....	51
Gleichung 34: Obergrenze des RQD-Indexes in Abhängigkeit vom L_m -Wert.....	51
Gleichung 35: Untergrenze des RQD-Indexes in Abhängigkeit vom L_m -Wert.....	51
Gleichung 36: Untergrenze des L_m -Wertes in Abhängigkeit vom Zerrüttungsgrad Z_G	53
Gleichung 37: Obergrenze des Zerrüttungsgrades Z_G in Abhängigkeit vom L_m -Wert.....	53
Gleichung 38: Untergrenze des Zerrüttungsgrades Z_G in Abhängigkeit vom L_m -Wert.....	53
Gleichung 39: Term für die Substitution des RQD-Index mit dem L_m -Wert.....	56
Gleichung 40: Rock Mass Quality Q mit substituiertem L_m -Wert.....	56

10 Stichwortverzeichnis

- B -

Barton..... 8, 25, 27, 28, 29
 Bieniawski..... 19, 20, 21, 46

- C -

CSIR Geomech. System..... *siehe* RMR-System
 Cummings..... 22

- D -

Datenbank..... 59, 60, 64-66
 Deere..... 15, 19, 20, 26, 27
 DS 853..... 8

- E -

Erhaltungsgrad..... 13, 14, 24, 33, 34

- G -

Gebirgsfaktor C .. 10, **22-24**, 34, 35, 39-40, 42, 45,
 54
 Gefügefaktor..... *siehe* Gebirgsfaktor C
 GSI..... 21

- H -

Hansági..... 18, 19, 23, 24

- I -

Isoperimere..... 49, 50

- J -

Jhun Itoh..... 29

- K -

Karst..... 30
 Kendorski..... 22
 Kerndurchmesser..... 12, 16, 23, 24, 30, 34
 Kernrohrtyp..... 12
 Klüftigkeitsziffer k..... 10, **17-18**, 34, 35, 42, 45

- L -

Laubscher..... 22
 Lauffer..... 6, 20
 Lien..... 25
 L_m-Wert..... 10, **24-25**, 34, 35, 38-39, 40, 42, 45,
 47-56
 Lunde..... 25

- M -

MBR-System..... **22**
 Meier..... 24, 47
 Mittlere Kernlänge ... 10, **18-19**, 34, 35, 40, 42, 45
 Möbus..... 17, 18
 Molek..... 30
 MRMR-System..... **21-22**

- N -

NGI-Verfahren..... *siehe* Rock Mass Quality Q

- O -

ÖNORM B2203..... 8

- P -

Page..... 22

- R -

RMR-System 10, **19-21**, 31, 34, 35, 40, 41, 42, 45
 Rock Mass Quality Q.... 10, **25-29**, 34, 35, 41, 42,
 45, 55-56
 Rock Mass Quality Q_{TBM}..... 11, **29-30**, 34, 35
 Rock Mass Ratio..... *siehe* RMR-System
 Rock Quality Designation..... *siehe* RQD-Index
 Romana..... 31
 RQD-Index..... 10, **15-16**, 19, 20, 26, 27, 34, 35,
 36-40, 42, 45, 48-51, 54, 55-56

- S -

SIA 198..... 8
 SMR-System..... 11, **31-32**, 34, 35, 42
 Stini..... 17

- T -

Taylor..... 22
 Terzaghi..... 6
 Trennflächenziffer..... *siehe* Klüftigkeitsziffer k

- Z -

Zerrüttungsgrad Z_G... 11, **30-31**, 34, 35, 36-39, 42,
 45, 52-53, 55

Anlagen

Anlage I: Eingangswerte Rock Mass Quality Q

J_n	Anzahl der Kluftscharen	
	Klüftigkeit	
	A	massiv, keine oder wenige Klüfte
	B (C)	eine Kluftchar
	D (E)	zwei Kluftscharen
	F (G)	drei Kluftscharen
	H	\geq vier Kluftscharen, ungerichtete Klüfte, stark zerklüftet
	J	zerdrücktes Gebirge
Klammerwerte bei zusätzlichem Auftreten von ungerichteten Einzelklüften.		
für Tunnelverschneidungen: $3.0 * J_n$		
für Portalbereiche: $2.0 * J_n$		
J_r	Rauhigkeit für die schwächste Spaltebene	
	Rauhigkeit	
	Kontakt der Kluftflächen und	
	Kontakt der Kluftflächen vor 10 cm Scherverschiebung	
	A	nicht durchstreichende Kluft
	B	rau oder unregelmäßig, wellig
	C	glatt, wellig
	D	beharnischt, wellig
	E	rau oder unregelmäßig, eben
	F	glatt, eben
	G	beharnischt, eben
kein Kontakt der Kluftflächen bei der Scherverschiebung		
H, J	kein Wand-Wand-Kontakt, Letten-, Ton- oder Sandfüllung	
bei Durchschnitsabstand des anwendbaren Kluftsystems größer als 3 m: $J_r + 1$		
für ebene beharnischte Klüfte mit günstig orientierter Strömung: $J_r = 0,5$		
J_w	Kluftwasserführung	
	Kluftwasserführung	J_w
		Wasserdruck u [N/mm ²]
	A	Trockener Vortrieb oder örtlich geringer Zufluss (> 5 l/min)
	B	Mittlerer Zufluss oder Druck, gelegentlich Ausspülung der Kluftfüllungen
	C	starker Zufluss oder hoher Druck in festem Gestein mit ungefüllten Klüften
	D	starker Zufluss oder hoher Druck, bedeutende Ausspülungen der Kluftfüllungen
E	Übermäßig hoher Zufluss oder Wasserdruck beim Sprengen, mit der Zeit abnehmend	
F	Übermäßig hoher Zufluss oder Wasserdruck, gleichbleibend ohne bemerkenswerte Abnahme	
Anmerkung: Die Faktoren C bis F sind grob geschätzt. Zunahme von J_w bei Einbau von Drainage. Spezielle Probleme durch Eisbildung sind nicht berücksichtigt.		

J_a	Umwandlungsgrad (Charakter der Klüfte oder Füllung längs der schwächsten Spaltebene)		
	Umwandlungsgrad	ϕ_r	
	Kontakt der Klufflächen		
	A fest verheilt, hart, nicht entfestigt, undurchlässige Füllung (z.B. Quarz, Epidote)	0,75	
	B nicht umgewandelte, verfärbte Klufflächen	1,0	25° - 35°
	C geringfügig verwitterte Klufflächen, nicht entfestigte Kluffbeläge, sandige Partikel, verwittertes Gestein	2,0	25° - 30°
	D siltige oder sandige, tonige Kluffbeläge, geringe Tonfraktion	3,0	20° - 25°
	E erweichbar oder mit geringer Reibung, Kluffbeläge (z.B.: Kaolit, Chlorite, Talk, Gips, Graphit), geringe Mengen von quellfähigem Ton (nicht durchstreichende Kluffbeläge, Dicke: kleiner als 1 bis 2 mm)	4,0	8° - 16°
	Kontakt der Klufflächen vor 10 cm Scherverschiebung		
	F sandige Partikel, tonfreies verwittertes Gestein	4,0	25° - 30°
	G stark überverdichtet, verhärtet, nicht erweichbare Tonmineralfüllungen (durchgehend, Dicke: < 5 mm)	6,0	16° - 24°
	H mittel oder gering verdichtet, erweichbar, tonige Mineralfüllung (durchgehend, Dicke: < 5 mm)	8,0	12° - 16°
	J quellfähige tonige Füllungen z.B.: Montmorillonit (durchgehend, Dicke: < 5 mm), (die Werte von J_a hängen vom Anteil quellfähiger Partikel ab und sind anfällig auf Wasserzutritt)	8,0 - 12,0	6° - 12°
	kein Kontakt der Klufflächen bei der Scherverschiebung		
	K Zonen oder Bänder von verwittertem	6,0	6° - 24°
	L oder zerbrochenem Gestein und	8,0	
	M Ton (siehe G, H, J für die Beschreibung des Tones)	8,0 - 12,0	
	N Zonen oder Bänder von siltigem oder sandigem Ton, geringe Tonfraktion (nicht erweichbar)	5,0	
	O Dicke, durchgehende Bänder und	10,0	6° - 24°
	P Zonen von Ton (siehe G, H, J, für die Beschreibung des Tones)	13,0 oder 13,0 - 20,0	

Anmerkung: Die Werte von ϕ_r sind ausgelegt für einen annähernden Führungswert in Bezug auf die mineralogischen Eigenschaften der Materialien (falls vorhanden)

Spannungsniveau		Spannungszustand		SRF
Schwächezonen kreuzen Vortrieb, welche verursachen können, dass das Gebirge beim Tunnelvortrieb aufgelockert wird.				
A	Vielfaches Vorkommen von schwachen Zonen, die Ton oder chemisch verwittertes Gestein enthalten. Sehr loses umgebendes Gestein. (jede Tiefe des Vortriebes)			10,0
B	Einzelne Schwächezonen, die Ton oder chemisch verwittertes Gestein enthalten. (Tiefe des Vortriebes ≤ 50 m)			5,0
C	Einzelne Schwächezonen, die Ton oder chemisch verwittertes Gestein enthalten. (Tiefe des Vortriebes > 50 m)			2,5
D	Mehrfache Scherzonen im festen Gestein (tonfrei), loses umgebendes Gestein (jede Tiefe des Vortriebes)			7,5
E	Einzelne Scherzonen im festen Gestein (frei von Ton), (Tiefe des Vortriebes ≤ 50 m)			5,0
F	Einzelne Scherzonen im festen Gestein (frei von Ton), (Tiefe des Vortriebes > 50 m)			2,5
G	Aufgelockerte offene Klüfte, stark zerklüftet (blockig gewürfelt, jede Tiefe des Vortriebes)			5,0
Festes Gestein, Probleme aus Gebirgsspannungen		σ_2/σ_1	σ_3/σ_1	
H	geringe Spannung, nahe der Oberfläche	>200	>13	2,5
J	mittlere Spannung	200 - 10	13 - 0,66	1,0
K	hohe Spannungen, geschlossene Strukturen (gewöhnlich günstig für Stabilität, kann ungünstig sein für Standsicherheit der Ulme)	10 - 5	0,66 - 0,33	0,5 - 2,0
L	nicht explosive Gebirgsschlaggefahr (massives Gestein)	5 - 2,5	0,33 - 0,16	5 - 10
M	explosive Gebirgsschlaggefahr (massives Gestein)	$<2,5$	$<1,6$	10 - 20
Druckhaftes Gebirge, plastisches Fließen von festem Gestein unter dem Einfluss hoher Gebirgsdrücke				
N	geringfügig druckhaftes Gebirge			5 - 10
O	sehr druckhaftes Gebirge			10 - 20
Quellendes Gestein, chemische Quellung in direkter Abhängigkeit vom Vorhandensein von Wasser				
P	schwach quellendes Gebirge			5 - 10
R	stark quellendes Gebirge			10 - 15

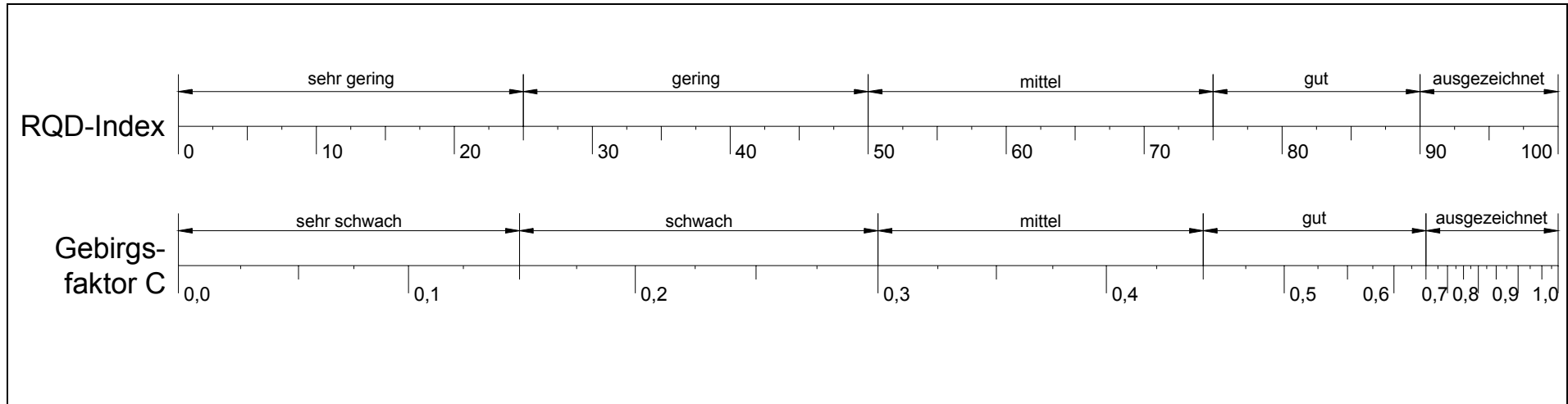
Anlage II: Rock Mass Rating System (nach BIENIAWSKI [6])

A. KLASIFIKATIONSPARAMETER UND DEREN WERTUNG									
Parameter			Wertebereich						
1	Festigkeit des intakten Gesteins	Punktbezogener Festigkeitsindex	> 10 MPa	4-10 MPa	2-4 MPa	1-2 MPa	< 1 MPa siehe Uniaxiale Druckfestigkeit		
		Einaxiale Druckfestigkeit	> 250 MPa	100-250 MPa	50-100 MPa	25-50 MPa	5-25 MPa	1-5 MPa	< 1 MPa
	Wertung	15	12	7	4	2	1	0	
2	RQD-Index		100% - 90%	90% - 75%	75% - 50%	50% - 25%	< 25%		
	Wertung		20	17	13	8	3		
3	Kluftabstand		> 2 m	0,6 - 2,0 m	200 - 600 mm	60 - 200 mm	< 60 mm		
	Wertung		20	15	10	8	5		
4	Kluftbeschaffenheit (See E)		sehr raue, unverwitterte Oberfläche, nicht kontinuierlich, keine Durchtrennung	schwach raue Oberfläche, Durchtrennung < 1 mm leicht angewittert	schwach raue Oberfläche, Durchtrennung < 1 mm, stark verwittert	harnischartige Oberfl. oder Letten < 5 mm Dicke oder Öffnungsweite 1-5 mm durchgehend	weiche Letten > 5 mm Dicke oder Öffnungsweite > 5 mm durchgehend		
	Wertung		30	25	20	10	0		
5	Zutritt von Grundwasser	Zufluss pro 10 m Tunnellänge (l/min)	(kein)	< 10	10 - 25	25 - 125	> 125		
		Kluftwasserdruck σ_1	0	< 0,1	0,1 - 0,2	0,2 - 0,5	> 0,5		
	allgem. Bedingungen		vollständig trocken	feucht	nass	tropfend	fließend		
	Wertung		15	10	7	4	0		
B. EINSCHÄTZUNG DER RAUMSTELLUNG DER KLÜFTE (VGL. F)									
Orientierung von Streichen und Fallen			sehr günstig	günstig	gut	ungünstig	sehr ungünstig		
Wertungen	Tunnel & Bergwerke		0	-2	-5	-10	-12		
	Gründungen		0	-2	-7	-15	-25		
	Hänge & Böschungen		0	-5	-25	-50			
C. GESTEINSKLASSEN, BESTIMMT ANHAND DER GESAMTWERTUNG									
Wertung			100 ← 81	80 ← 61	60 ← 41	40 ← 21	< 21		
Klasse			I	II	III	IV	V		
Beschreibung			sehr guter Fels	guter Fels	genügender Fels	schlechter Fels	sehr schlechter Fels		
D. BEDEUTUNG DER GESTEINSKLASSEN									
Klasse des Gesteins			I	II	III	IV	V		
durchschnittliche Standzeit			20 Jahre für 15 m Spannweite	1 Jahr für 10 m Spannweite	1 Woche für 5 m Spannweite	10 Stunden für 2,5 m Spannweite	30 Minuten für 1 m Spannweite		
Kohäsion c (kPa)			> 400	300 - 400	200 - 300	100 - 200	< 100		
Reibungswinkel (Grad)			> 45°	35° - 45°	25° - 35°	15° - 25°	< 15°		
E. RICHTLINIEN DER KLASIFIKATION VON KLUFTEBEDINGUNGEN*									
Kluftlänge (Ausbeißen)			< 1 m	1 - 3 m	3 - 10 m	10 - 20 m	> 20 m		
Wertung			6	4	2	1	0		
Durchtrennung (Öffnung)			(keine)	< 0,1 mm	0,1 - 1,0 mm	1 - 5 mm	> 5 mm		
Wertung			6	5	4	1	0		
Rauhigkeit			sehr rau	rau	schwach rau	glatt	harnischartig		
Wertung			6	5	3	1	0		
Füllung (Letten)			keine	hart < 5 mm	hart > 5 mm	weich < 5 mm	weich > 5 mm		
Wertung			6	4	3	3	0		
Verwitterungsgrad			unverwittert	angewittert	mäßig verwittert	stark verwittert	zersetzt		
Wertung			6	5	3	1	0		
F. EFFEKT DES STREICHENS UND FALLENS IN BEZUG AUF DAS TUNNELBAUWERK**									
Streichen senkrecht zu Tunnelachse				Streichen parallel zu Tunnelachse					
Vortrieb mit Fallen - 45 - 90° sehr ungünstig			Vortrieb mit Fallen - 20 - 45° günstig		Fallen 45 - 90° sehr ungünstig		Fallen 20 - 45° genügend		
Vortrieb gegen Fallen - 45 - 90° genügend			Vortrieb gegen Fallen - 20 - 45° ungünstig		Fallen 0-20° - unabhängig von Streichen genügend				

* Einige dieser Bedingungen hängen voneinander ab. Zum Beispiel bei Anwesenheit von Füllungen wird die Rauigkeit der Oberfläche überprägt. In diesen Fällen wird empfohlen A.4 zu nutzen.

** modifiziert nach WICKHAM et al (1972).

Anlage III: Nomogramm der Beziehung zwischen RQD-Index und Gebirgsfaktor C



Anlage IV: Datenblatt Programm und ActiveX-DLL**Ausführbares Programm: Gebirgsklassifikation**

Eigenschaft	Wert
Programmname	Gebirgsklassifikation
Version	1.0.3
benötigter Festplattenplatz	mind. 25 MB + DCOM98 (falls benötigt)
Mindestgröße RAM	24 MB
Name der ausführbaren Datei	Gebirgsklassifikation.exe
Position auf dem Datenträger	\\EDV-gerechte Umsetzung\Install\Gebirgsklassifikation

ActiveX-Komponente: RockClassX

Eigenschaft	Wert
Name d. ActiveX-Komponente	RockClassX
Version	1.0.16
Projektname	RockClassX
Projektbeschreibung	Gebirgsklassifikation ActiveX-DLL (Studienarbeit Jörg Meier, 2001)
Projekttyp	ActiveX-DLL
Dateiname der ActiveX-DLL	RockClassification.dll
Versionskompatibilität	Projekt-Kompatibilität
Position auf dem Datenträger	\\EDV-gerechte Umsetzung\Install\RockClassX

Anlage V: Definitionen der Tabellen der zugrunde liegenden Datenbank

Tabelle: Bohrungen

Feldname	Datentyp	Beschreibung
BohrungID	AutoWert (GUID)	Replikations-ID (GUID)
Name	Text (50 Zeichen)	Eindeutige Bezeichnung der Bohrung
Teufe	Zahl (Double)	Tiefe der Bohrung in Metern
Durchmesser	Zahl (Double)	Durchmesser in Millimeter
KoordinatenX	Zahl (Double)	Rechtswert des Bohrloches
KoordinatenY	Zahl (Double)	Hochwert des Bohrloches
KoordinatenZ	Zahl (Double)	Höhe des Bohrloches
Bohrverfahren	Text (255 Zeichen)	Name des verwendeten Bohrverfahrens
Bohrwerkzeug	Text (255 Zeichen)	Name des verwendeten Bohrwerkzeugs
Bemerkungen	Memo	Bemerkungen

Tabelle: Kernmärsche

Feldname	Datentyp	Beschreibung
KernmarschID	AutoWert (GUID)	Replikations-ID (GUID)
BohrungID	Zahl (GUID)	Verweis auf die übergeordnete Bohrung
Teufe	Zahl (Double)	Teufe unter Bohrlochoberkante in Metern
Länge	Zahl (Double)	Länge des Kernmarsches in Metern
Bruchstücke	Memo	Bruchstücke des Bohrkernes in Zentimetern; getrennt durch Slash "/" oder Semikolon ";"; Angabe der Erhaltungsklassen möglich
Bemerkungen	Memo	Bemerkungen

Tabelle: Objekte

Feldname	Datentyp	Beschreibung
AbbildungID	AutoWert (GUID)	Replikations-ID (GUID)
KernmarschID	Zahl (GUID)	Verweis auf den übergeordneten Kernmarsch
Objekt	OLE-Objekt	Eingebundenes OLE-Objekt (z.B. Bild)
Beschreibung	Memo	Bemerkungen und Erklärungen

Tabelle: Bohrverfahren

Feldname	Datentyp	Beschreibung
Bohrverfahren	Text (255 Zeichen)	Name des Bohrverfahrens

Tabelle: Bohrwerkzeug

Feldname	Datentyp	Beschreibung
Bohrwerkzeug	Text (255 Zeichen)	Name des Bohrwerkzeuges

Anlage VI: Quellcode des Programms und der ActiveX-Komponente

Im Folgenden wurde der Quellcode des Programms „Gebirgsklassifikation“ und der ActiveX-Komponente „RockClassX“ abgedruckt. Der vollständige Quellcode befindet sich zusätzlich zu der kompilierten Version des Quellcodes nochmals auf dem beigelegten Datenträger im Verzeichnis „\EDV-gerechte Umsetzung\Quelltext“.

Da im Rahmen dieser Arbeit eine vollständige Beschreibung der Funktionen und Vorgänge innerhalb der programmierten Projekte nicht möglich ist und um eine Einarbeitung in den Quelltext zu erleichtern, wurden in den Quelltext Kommentare eingefügt. Diese Kommentare werden durch ein Hochkomma eingeleitet (gelten dann für den Rest der Zeile) und werden vom Compiler ignoriert. Sie wurden an Stellen eingefügt, wo es der Autor für sinnvoll hielt. Aufgrund der hohen Zahl von Quelltextzeilen kann nicht jeder Befehlsblock oder jede Funktion von einem Kommentar erklärt werden. Wurden gleiche Funktionstypen mehrfach verwendet, so wurden diese in der Regel nur bei ihrer ersten Definition kommentiert. (Beispiel: Die Funktionen der Klasse „clsClassification“, welche in den Klassen der einzelnen Verfahren immer wieder implementiert werden, enthalten nur in der Klasse „clsClassification.cls“ Kommentare.)

Um die Lesbarkeit des Quelltextes zu verbessern wurden am Anfang jeder Zeile Zeilennummern eingefügt. Diese beziehen sich auf den Quelltext, nicht auf die hier abgedruckte Zeilenzahl (Sie gehören nicht zum eigentlichen Quelltext!). Sollte die Quelltextzeile länger sein, als die hier zur Verfügung stehende Seitenbreite, so wird der Quelltext umgebrochen und keine neue Zeilennummer angezeigt.

Quellcode: Programm Gebirgsklassifikation

Datei: Gebirgsklassifikation.vbg (VB Group Project)

01	VBGROUP 5.0
02	Project=RockClassX\RockClassX.vbp
03	StartupProject=Gebirgsklassifikation\Gebirgsklassifikation.vbp
04	Project=..\..\..\..\Programmierung\Tests\Sheet\Sheet.vbp
05	

Datei: Gebirgsklassifikation.vbp (VB Project)

01	Type=Exe
02	Reference=*\G{00020430-0000-0000-C000-00000000046}#2.0#0#C:\WINNT\System32\stdole2.tlb#OLE Automation
03	Reference=*\G{00025E01-0000-0000-C000-00000000046}#4.0#0#C:\Programme\Gemeinsame Dateien\Microsoft Shared\DAO\DAO350.DLL#Microsoft DAO 3.51 Object Library
04	Reference=*\G{00000200-0000-0010-8000-00AA006D2EA4}#2.0#0#C:\Programme\Gemeinsame Dateien\System\ADO\msado20.tlb#Microsoft ActiveX Data Objects 2.0 Library
05	Reference=*\G{56BF9020-7A2F-11D0-9482-00A0C91110ED}#1.0#0#C:\WINNT\System32\MSBIND.DLL#Microsoft Data Binding Collection
06	Reference=*\G{420B2830-E718-11CF-893D-00A0C9054228}#1.0#0#C:\WINNT\System32\scrrun.dll#Microsoft Scripting Runtime
07	Reference=*\G{1A65CB0B-5352-48BE-8A60-A9F3223E8004}#b.0#0#..\RockClassX\RockClassX.dll#Gebirgsklassifikation ActiveX-DLL (Studienarbeit Jörg Meier, 2001)
08	Reference=*\G{13F84498-7563-4D9D-9DD6-7F1F8BF4E57F}#3.0#0#..\..\..\..\Programmierung\Tests\Sheet\Sheet.dll#
09	Object={831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0; MSCOMCTL.OCX
10	Object={F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0; COMDLG32.OCX
11	Object={67397AA1-7FB1-11D0-B148-00A0C922E820}#6.0#0; MSADODC.OCX
12	Object={5E9E78A0-531B-11CF-91F6-C2863C385E30}#1.0#0; MSFLXGRD.OCX
13	Object={86CF1D34-0C5F-11D2-A9FC-0000F8754DA1}#2.0#0; MSCOMCT2.OCX

14	Object={BDC217C8-ED16-11CD-956C-0000C04E4C0A}#1.1#0; TABCTL32.OCX
15	Object={6FBA474E-43AC-11CE-9A0E-00AA0062BB4C}#1.0#0; SYSINFO.OCX
16	Module=basMain; Gebirgsklassifikation.bas
17	Form=frmIntro.frm
18	Form=frmMain.frm
19	Form=frmInfo.frm
20	UserControl=ctlParameter.ctl
21	Form=frmEditKernmarsch.frm
22	Class=clsEditGrid; ..\..\..\..\Programmierung\CLS\EditGrid\clsEditGrid.cls
23	Module=APIStandard; ..\..\..\..\Programmierung\BAS\APIStandard.bas
24	Module=WindowAndControlFunctions; ..\..\..\..\Programmierung\BAS\WindowAndControlFunctions.bas
25	Module=Standard; ..\..\..\..\Programmierung\BAS\Standard.bas
26	Module=FileCommands; ..\..\..\..\Programmierung\BAS\FileCommands.bas
27	UserControl=ctlBohrung.ctl
28	Form=frmEditBohrung.frm
29	Form=frmVerweise.frm
30	Class=clsClassCol; clsClassificationCollections.cls
31	Form=frmAssistentKernmärsche.frm
32	Form=frmOLEObjekt.frm
33	IconForm="frmMain"
34	Startup="Sub Main"
35	HelpFile=""
36	Title="Gebirgsklassifikation"
37	ExeName32="Gebirgsklassifikation.exe"
38	Command32=""
39	Name="Gebirgsklassifikation"
40	HelpContextID="0"
41	CompatibleMode="0"
42	MajorVer=1
43	MinorVer=0
44	RevisionVer=3
45	AutoIncrementVer=0
46	ServerSupportFiles=0
47	VersionCompanyName=""
48	VersionLegalCopyright="2001, Jörg Meier"
49	VersionProductName="Gebirgsklassifikation"
50	CompilationType=0
51	OptimizationType=0
52	FavorPentiumPro(tm)=0
53	CodeViewDebugInfo=0
54	NoAliasing=0
55	BoundsCheck=0
56	OverflowCheck=0

```

57 FlPointCheck=0
58 FDIVCheck=0
59 UnroundedFP=0
60 StartMode=0
61 Unattended=0
62 Retained=0
63 ThreadPerObject=0
64 MaxNumberOfThreads=1
65 DebugStartupOption=0
66
67 [MS Transaction Server]
68 AutoRefresh=1
69

```

Datei: Gebirgsklassifikation.bas (VB Module)

```

01 Attribute VB Name = "basMain"
02 Option Explicit
03 Public ClassificationCollection As clsClassCol
04
05 Public Sub Main()
06     frmIntro.Show
07
08     Set ClassificationCollection = New clsClassCol
09     With ClassificationCollection
10         .Load App.ProductName, "Classifications"
11         If .Count = 0 Then
12             .Reset
13         End If
14     End With
15
16     Dim T As Double
17     T = Timer
18     Do: DoEvents: Loop Until Abs(Timer - T) >= 0.5
19
20     frmMain.Show
21     Unload frmIntro
22 End Sub
23
24 Public Sub PutTextBox(Field As DAO.Field, TextBox As Object)
25     'füllt eine TextBox mit dem Inhalt einen Datenbankfeldes
26     With Field
27         If IsNull(.Value) Then
28             TextBox = vbNullString
29         Else
30             TextBox = .Value
31         End If

```

```

32     End With
33 End Sub
34 Public Sub GetTextBox(Field As DAO.Field, TextBox As Object)
35     'füllt eines Datenbankfeldes mit dem Inhalt einer TextBox
36     On Error Resume Next
37     Err.Clear
38
39     If TextBox = vbNullString And Not Field.Required Then
40         Field.Value = Null
41     Else
42         Field.Value = TextBox
43     End If
44
45     If Err <> 0 Then
46         MsgBox "Bei der Übertragung der Daten in die Datenbank ist
ein Fehler im Feld "" & Field.Name & "" (Wert: "" &
TextBox.Text & """) aufgetreten: " & Err.Description & "
(Fehlernummer: " & Err.Number & "). Das Feld wird nicht
aktualisiert", vbExclamation, "Datenbankanbindung"
47     End If
48 End Sub
49

```

Datei: clsClassificationCollections.cls (VB Class Module)

```

01 VERSION 1.0 CLASS
02 BEGIN
03     MultiUse = -1 'True
04     Persistable = 0 'NotPersistable
05     DataBindingBehavior = 0 'vbNone
06     DataSourceBehavior = 0 'vbNone
07     MTSTransactionMode = 0 'NotAnMTSObject
08 END
09 Attribute VB Name = "clsClassCol"
10 Attribute VB GlobalNameSpace = False
11 Attribute VB Creatable = True
12 Attribute VB_PredeclaredId = False
13 Attribute VB_Exposed = False
14 Option Explicit
15
16 Public Event Change()
17 Public ChancelChange As Boolean
18 Private prvChancelChange As Boolean
19
20 Private Type ClsType
21     Class As String
22     Server As String

```

```
23 End Type
24 Private Classes() As ClsType
25
26 Private Const keyClassCount As String = "ClassificationCount"
27 Private Const keyClassClass As String = "ClassificationClass"
28 Private Const keyClassServer As String = "ClassificationServer"
29
30 Public Property Get Count() As Long
31     Count = UBound(Classes)
32 End Property
33
34 Public Property Get Classification(Item As Long) As
RockClassX.clsClassification
35     With Classes(Item)
36         If .Server = vbNullString Then
37             Set Classification = CreateObject(.Class)
38         Else
39             Set Classification = CreateObject(.Class, .Server)
40         End If
41     End With
42 End Property
43
44 Public Sub Add(Class As String, Optional Server As String =
vbNullString)
45     If Not ((Class = vbNullString And Server = vbNullString) Or
(Class = vbNullString)) Then
46         If Count = 0 Then
47             ReDim Classes(1 To 1)
48         Else
49             ReDim Preserve Classes(1 To Count + 1)
50         End If
51         With Classes(Count)
52             .Class = Class
53             .Server = Server
54         End With
55     End If
56
57     If Not (ChancelChange Or prvChancelChange) Then RaiseEvent
Change
58 End Sub
59
60 Public Property Get ClassificationClass(Item As Long) As String
61     ClassificationClass = Classes(Item).Class
62 End Property
63 Public Property Get ClassificationServer(Item As Long) As String
64     ClassificationServer = Classes(Item).Server
65 End Property
66
```

```
67 Private Sub Class_Initialize()
68     Clear
69 End Sub
70
71 Public Sub Reset()
72     prvChancelChange = True
73     Clear
74
75     Dim RCXClassifications As Collection, C As clsClassification
76     Set RCXClassifications = RockClassX.ClassificationList
77     For Each C In RCXClassifications
78         Add "RockClassX." & TypeName(C)
79     Next
80
81     prvChancelChange = False
82     If Not (ChancelChange Or prvChancelChange) Then RaiseEvent
Change
83 End Sub
84
85 Public Sub Clear()
86     ReDim Classes(0 To 0)
87     If Not (ChancelChange Or prvChancelChange) Then RaiseEvent
Change
88 End Sub
89
90 Public Sub Load(AppName As String, Section As String)
91     Dim N As Long, C As Long
92     Dim Class As String, Server As String
93
94     prvChancelChange = True
95     Clear
96
97     C = Val(GetSetting(AppName, Section, keyClassCount, 0))
98     For N = 1 To C
99         Class = GetSetting(AppName, Section, keyClassClass & N,
vbNullString)
100        Server = GetSetting(AppName, Section, keyClassServer & N,
vbNullString)
101        Add Class, Server
102    Next
103    prvChancelChange = False
104    If Not (ChancelChange Or prvChancelChange) Then RaiseEvent
Change
105 End Sub
106 Public Sub Save(AppName As String, Section As String)
107     Dim N As Long
108
109     SaveSetting AppName, Section, keyClassCount, Count
```

```

110     For N = 1 To Count
111         With Classes(N)
112             SaveSetting AppName, Section, keyClassClass & N, .Class
113             SaveSetting AppName, Section, keyClassServer & N, .Server
114         End With
115     Next
116 End Sub
117
118 Public Sub RaiseChange()
119     RaiseEvent Change
120 End Sub
121
122 Public Function ClassificationClassByClassname(Classname As
String) As RockClassX.clsClassification
123     Dim N As Long
124     For N = 1 To Count
125         With Classes(N)
126             If .Class = Classname Then
127                 Set ClassificationClassByClassname = Classification(N)
128             End If
129         End With
130     Next
131 End Function
132

```

Datei: ctlBohrung.ctl (VB User Control)

```

01 VERSION 5.00
02 Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0";
"MSCOMCTL.OCX"
03 Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0";
"COMDLG32.OCX"
04 Object = "{6FBA474E-43AC-11CE-9A0E-00AA0062BB4C}#1.0#0";
"SYSINFO.OCX"
05 Begin VB.UserControl ctlBohrung
06     ClientHeight = 6705
07     ClientLeft = 0
08     ClientTop = 0
09     ClientWidth = 9180
10     ScaleHeight = 6705
11     ScaleWidth = 9180
12     Begin SysInfoLib.SysInfo SysInfo1
13         Left = 6960
14         Top = 720
15         _ExtentX = 1005
16         ExtentY = 1005
17         _Version = 393216

```

```

18     End
19     Begin MSComDlg.CommonDialog CommonDialog1
20         Left = 8040
21         Top = 960
22         ExtentX = 847
23         _ExtentY = 847
24         _Version = 393216
25     End
26     Begin VB.ComboBox Combol
27         Height = 315
28         Left = 4320
29         TabIndex = 15
30         Text = "Combol"
31         ToolTipText = "Zoom"
32         Top = 30
33         Width = 1455
34     End
35     Begin VB.PictureBox NfoPanel
36         BackColor = &H80000018&
37         BorderStyle = 0 'Kein
38         Height = 615
39         Left = 1800
40         ScaleHeight = 615
41         ScaleWidth = 6735
42         TabIndex = 13
43         Top = 1680
44         Width = 6735
45     Begin VB.Image NfoPanelImage
46         Height = 480
47         Left = 75
48         Picture = "ctlBohrung.ctx":0000
49         Top = 75
50         Width = 480
51     End
52     Begin VB.Label NfoPanelLabel
53         BackStyle = 0 'Transparent
54         Caption = "Label2"
55         ForeColor = &H80000017&
56         Height = 615
57         Left = 600
58         TabIndex = 14
59         Top = 75
60         Width = 5295
61     End
62     End
63     Begin MSComctlLib.ImageList ImageList2
64         Left = 8040
65         Top = 120

```

```

66     _ExtentX      = 1005
67     _ExtentY      = 1005
68     BackColor     = -2147483643
69     ImageWidth    = 16
70     ImageHeight   = 16
71     MaskColor     = 12632256
72     _Version      = 393216
73     BeginProperty Images {2C247F25-8591-11D1-B16A-
00C0F0283628}
74         NumListImages = 2
75         BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-
00C0F0283628}
76             Picture      = "ctlBohrung.ctx":0442
77             Key          = "export"
78         EndProperty
79         BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-
00C0F0283628}
80             Picture      = "ctlBohrung.ctx":059C
81             Key          = "ZoomAlles"
82         EndProperty
83     EndProperty
84 End
85 Begin MSComctlLib.Toolbar Toolbar2
86     Align          = 1 'Oben ausrichten
87     Height         = 330
88     Left           = 0
89     TabIndex       = 12
90     Top            = 0
91     Width          = 9180
92     _ExtentX      = 16193
93     _ExtentY      = 582
94     ButtonWidth   = 1138
95     ButtonHeight  = 582
96     AllowCustomize = 0 'False
97     Wrappable     = 0 'False
98     Style          = 1
99     TextAlignment = 1
100    Version        = 393216
101 End
102 Begin VB.PictureBox Picture6
103     BorderStyle    = 0 'Kein
104     Height         = 2775
105     Left           = 840
106     ScaleHeight   = 2775
107     ScaleWidth    = 7695
108     TabIndex       = 9
109     TabStop        = 0 'False
110     Top            = 2400

```

```

111     Width          = 7695
112     Begin MSComctlLib.ImageList ImageList3
113         Left       = 6120
114         Top        = 240
115         ExtentX    = 1005
116         _ExtentY  = 1005
117         BackColor  = -2147483643
118         ImageWidth = 16
119         ImageHeight = 16
120         MaskColor  = 12632256
121         Version    = 393216
122     BeginProperty Images {2C247F25-8591-11D1-B16A-
00C0F0283628}
123         NumListImages = 3
124         BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-
00C0F0283628}
125             Picture      = "ctlBohrung.ctx":0E76
126             Key          = "new"
127         EndProperty
128         BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-
00C0F0283628}
129             Picture      = "ctlBohrung.ctx":12C8
130             Key          = "edt"
131         EndProperty
132         BeginProperty ListImage3 {2C247F27-8591-11D1-B16A-
00C0F0283628}
133             Picture      = "ctlBohrung.ctx":1422
134             Key          = "del"
135         EndProperty
136     EndProperty
137 End
138 Begin MSComctlLib.ListView ListView3
139     Height         = 2415
140     Left           = 4920
141     TabIndex       = 16
142     Top            = 360
143     Width          = 2775
144     ExtentX        = 4895
145     ExtentY        = 4260
146     View           = 3
147     LabelEdit      = 1
148     LabelWrap      = -1 'True
149     HideSelection  = 0 'False
150     FullRowSelect  = -1 'True
151     GridLines      = -1 'True
152     Version        = 393217
153     ForeColor      = -2147483640
154     BackColor      = -2147483643

```

```

155     BorderStyle      = 1
156     Appearance      = 1
157     NumItems        = 2
158     BeginProperty ColumnHeader(1) {BDD1F052-858B-11D1-B16A-
00C0F0283628}
159         Text          = "enthaltene OLE-Objekte"
160         Object.Width   = 3528
161     EndProperty
162     BeginProperty ColumnHeader(2) {BDD1F052-858B-11D1-B16A-
00C0F0283628}
163         SubItemIndex  = 1
164         Text          = "Kernmarsch"
165         Object.Width   = 1764
166     EndProperty
167 End
168 Begin MSComctlLib.ImageList ImageList1
169     Left              = 3600
170     Top               = 120
171     _ExtentX         = 1005
172     _ExtentY         = 1005
173     BackColor        = -2147483643
174     ImageWidth       = 16
175     ImageHeight      = 16
176     MaskColor        = 12632256
177     _Version         = 393216
178     BeginProperty Images {2C247F25-8591-11D1-B16A-
00C0F0283628}
179         NumListImages = 3
180     BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-
00C0F0283628}
181         Picture       = "ctlBohrung.ctx":1874
182         Key           = "new"
183     EndProperty
184     BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-
00C0F0283628}
185         Picture       = "ctlBohrung.ctx":1CC6
186         Key           = "del"
187     EndProperty
188     BeginProperty ListImage3 {2C247F27-8591-11D1-B16A-
00C0F0283628}
189         Picture       = "ctlBohrung.ctx":2118
190         Key           = "edt"
191     EndProperty
192 EndProperty
193 End
194 Begin MSComctlLib.Toolbar Toolbar1
195     Height            = 330
196     Left              = 0

```

```

197     TabIndex         = 11
198     Top              = 0
199     Width            = 4815
200     ExtentX          = 8493
201     ExtentY          = 582
202     ButtonWidth      = 1138
203     ButtonHeight     = 582
204     AllowCustomize   = 0 'False
205     Wrappable        = 0 'False
206     Style            = 1
207     TextAlignment    = 1
208     Version          = 393216
209 End
210 Begin MSComctlLib.ListView ListView2
211     Height           = 2415
212     Left             = 0
213     TabIndex        = 10
214     Top             = 360
215     Width           = 4815
216     _ExtentX       = 8493
217     _ExtentY       = 4260
218     SortKey         = 1
219     View            = 3
220     LabelEdit       = 1
221     Sorted          = -1 'True
222     LabelWrap       = -1 'True
223     HideSelection   = 0 'False
224     FullRowSelect   = -1 'True
225     GridLines       = -1 'True
226     _Version        = 393217
227     ForeColor       = -2147483640
228     BackColor       = -2147483643
229     BorderStyle     = 1
230     Appearance      = 1
231     NumItems        = 4
232     BeginProperty ColumnHeader(1) {BDD1F052-858B-11D1-B16A-
00C0F0283628}
233         Text          = "#"
234         Object.Width   = 882
235     EndProperty
236     BeginProperty ColumnHeader(2) {BDD1F052-858B-11D1-B16A-
00C0F0283628}
237         Alignment     = 1
238         SubItemIndex  = 1
239         Text           = "Teufe"
240         Object.Width   = 2540
241     EndProperty

```

```
242      BeginProperty ColumnHeader(3) {BDD1F052-858B-11D1-B16A-
00C0F0283628}
243          Alignment      = 1
244          SubItemIndex    = 2
245          Text             = "Länge"
246          Object.Width    = 2540
247      EndProperty
248      BeginProperty ColumnHeader(4) {BDD1F052-858B-11D1-B16A-
00C0F0283628}
249          SubItemIndex    = 3
250          Text             = "Bemerkungen"
251          Object.Width    = 14111
252      EndProperty
253  End
254  Begin MSComctlLib.Toolbar Toolbar3
255      Height      = 330
256      Left        = 5040
257      TabIndex    = 17
258      Top         = 0
259      Width       = 2175
260      ExtentX     = 3836
261      ExtentY     = 582
262      ButtonWidth = 609
263      ButtonHeight = 582
264      AllowCustomize = 0 'False
265      Wrappable   = 0 'False
266      Style       = 1
267      Version     = 393216
268  End
269  End
270  Begin VB.Timer Timer1
271      Interval      = 100
272      Left          = 7440
273      Top          = 1200
274  End
275  Begin VB.PictureBox Picture1
276      Height        = 5775
277      Left          = 0
278      ScaleHeight   = 381
279      ScaleMode     = 3 'Pixel
280      ScaleWidth    = 445
281      TabIndex      = 0
282      TabStop       = 0 'False
283      Top           = 720
284      Width         = 6735
285  Begin VB.PictureBox Picture2
286      Appearance    = 0 '2D
287      BackColor     = &H80000005&
```

```
288      BorderStyle  = 0 'Kein
289      ForeColor     = &H80000008&
290      Height        = 5145
291      Left          = 15
292      ScaleHeight   = 343
293      ScaleMode     = 3 'Pixel
294      ScaleWidth    = 415
295      TabIndex      = 3
296      TabStop       = 0 'False
297      Top           = 0
298      Width         = 6225
299  Begin VB.PictureBox Picture5
300      Appearance    = 0 '2D
301      BorderStyle  = 0 'Kein
302      ForeColor     = &H80000008&
303      Height        = 4890
304      Left          = -15
305      ScaleHeight   = 326
306      ScaleMode     = 3 'Pixel
307      ScaleWidth    = 399
308      TabIndex      = 4
309      TabStop       = 0 'False
310      Top           = 0
311      Width         = 5985
312  Begin VB.PictureBox Picture3
313      Appearance    = 0 '2D
314      BackColor     = &H00FFFFFF&
315      BorderStyle  = 0 'Kein
316      ForeColor     = &H80000008&
317      Height        = 4350
318      Left          = 15
319      ScaleHeight   = 290
320      ScaleMode     = 3 'Pixel
321      ScaleWidth    = 399
322      TabIndex      = 5
323      TabStop       = 0 'False
324      Top           = 270
325      Width         = 5985
326  Begin VB.PictureBox Picture4
327      Appearance    = 0 '2D
328      AutoRedraw    = -1 'True
329      BackColor     = &H00FFFFFF&
330      BorderStyle  = 0 'Kein
331      ForeColor     = &H80000008&
332      Height        = 3555
333      Left          = 0
334      ScaleHeight   = 237
335      ScaleMode     = 3 'Pixel
```

```
336         ScaleWidth      = 286
337         TabIndex        = 6
338         TabStop         = 0 'False
339         Top              = 0
340         Width           = 4290
341     End
342 End
343 Begin MSComctlLib.ListView ListView1
344     CausesValidation= 0 'False
345     Height            = 900
346     Left              = 1950
347     TabIndex         = 7
348     TabStop          = 0 'False
349     Top               = 0
350     Width            = 4035
351     ExtentX          = 7117
352     ExtentY          = 1588
353     View              = 3
354     LabelEdit        = 1
355     LabelWrap         = 0 'False
356     HideSelection    = -1 'True
357     AllowReorder     = -1 'True
358     Version           = 393217
359     ForeColor         = -2147483640
360     BackColor         = -2147483643
361     Appearance       = 0
362     NumItems         = 0
363 End
364 Begin VB.Label Label1
365     Alignment         = 2 'Zentriert
366     BackStyle         = 0 'Transparent
367     Caption           = "Bohrprofil"
368     BeginProperty Font
369         Name           = "MS Sans Serif"
370         Size           = 8.25
371         Charset        = 0
372         Weight         = 700
373         Underline      = 0 'False
374         Italic         = 0 'False
375         Strikethrough  = 0 'False
376     EndProperty
377     Height            = 240
378     Left              = 15
379     TabIndex         = 8
380     Top               = 30
381     Width             = 1920
382 End
383 End
```

```
384     End
385     Begin VB.VScrollBar VScroll1
386         Height         = 5220
387         Left           = 6375
388         Max            = 500
389         TabIndex      = 2
390         TabStop       = 0 'False
391         Top            = 15
392         Width         = 285
393     End
394     Begin VB.HScrollBar HScroll1
395         Height         = 300
396         Left           = 60
397         Max            = 500
398         TabIndex      = 1
399         TabStop       = 0 'False
400         Top            = 5310
401         Width         = 6240
402     End
403 End
404 End
405 Attribute VB Name = "ctlBohrung"
406 Attribute VB GlobalNameSpace = False
407 Attribute VB_Creatable = True
408 Attribute VB_PredeclaredId = False
409 Attribute VB_Exposed = False
410 Option Explicit
411
412 Private Declare Function CreateEnhMetaFile Lib "gdi32" Alias
"CreateEnhMetaFileA" (ByVal hdcRef As Long, ByVal lpFileName As
String, lpRect As Rect, ByVal lpDescription As String) As Long
413 Private Declare Function CloseEnhMetaFile Lib "gdi32" (ByVal hdc
As Long) As Long
414 Private Declare Function DeleteEnhMetaFile Lib "gdi32" (ByVal
hemf As Long) As Long
415
416 Private RSB Bohrungen As DAO.Recordset
417 Private RSKernmärsche As DAO.Recordset
418 Private RSObjekte As DAO.Recordset
419 Private BohrungKey As String
420 Private DatabaseFileName As String
421
422 Private Sheet As prjSheet.Sheet, NextSheetHeight As Double
423 Private WithEvents MyClassificationCollection As clsClassCol
424 Attribute MyClassificationCollection.VB VarHelpID = -1
425
426 Private ShowNfoPanel As Boolean
```

```
427
428 Private Type ColumnType
429     pos As Double
430     Width As Double
431     Key As String
432     Classification As RockClassX.clsClassification
433 End Type
434
435 Public Sub Export2WMF()
436     Toolbar2 ButtonClick Toolbar2.Buttons("Export")
437 End Sub
438
439 Private Function FormatDouble(Zahl As Double, FormatString,
440 Stellen As Long) As String
441     Dim s As String
442
443     s = Format(Zahl, FormatString)
444     FormatDouble = String(Stellen - Len(s), " ") & s
445 End Function
446
447 Private Function getParameterDictionary(RS As DAO.Recordset) As
448 Scripting.Dictionary
449     Set getParameterDictionary =
450 RockClassX.String2Dictionary(RS.Fields("Parameter") &
451 vbNullString)
452     getParameterDictionary.Add "pieces", RS.Fields("Bruchstücke")
453     getParameterDictionary.Add "kernmarsch", RS.Fields("Länge") *
454 100
455 End Function
456
457 Private Sub Combo1_Click()
458     DrawSheet
459 End Sub
460
461 Private Sub Combo1_KeyDown(KeyCode As Integer, Shift As Integer)
462     If KeyCode = vbKeyReturn Then
463         Combo1 = Fix(CMe(Combo1)) & " %"
464         Call Combo1_Click
465     End If
466 End Sub
467
468 Private Sub Combo1_LostFocus()
469     Combo1_Click
470 End Sub
471
472 Private Sub ListView2_ColumnClick(ByVal ColumnHeader As
473 MSComctlLib.ColumnHeader)
474     ListViewColumnClick ListView2, ColumnHeader.Index
```

```
469 End Sub
470
471 Private Sub ListView2_DblClick()
472     Toolbar1 ButtonClick Toolbar1.Buttons("edt")
473 End Sub
474
475 Private Sub ListView2_KeyDown(KeyCode As Integer, Shift As
476 Integer)
477     If KeyCode = vbKeyDelete Then
478         Toolbar1 ButtonClick Toolbar1.Buttons("del")
479     End If
480 End Sub
481
482 Private Sub ListView2_KeyPress(KeyAscii As Integer)
483     If KeyAscii = vbKeyReturn Then
484         Call ListView2_DblClick
485     End If
486 End Sub
487
488 Private Sub ListView3_ColumnClick(ByVal ColumnHeader As
489 MSComctlLib.ColumnHeader)
490     ListViewColumnClick ListView3, ColumnHeader.Index
491 End Sub
492
493 Private Sub ListView3_DblClick()
494     Toolbar3 ButtonClick Toolbar3.Buttons("edt")
495 End Sub
496
497 Private Sub ListView3_KeyDown(KeyCode As Integer, Shift As
498 Integer)
499     If KeyCode = vbKeyDelete Then
500         Toolbar3 ButtonClick Toolbar3.Buttons("del")
501     End If
502 End Sub
503
504 Private Sub ListView3_KeyPress(KeyAscii As Integer)
505     If KeyAscii = vbKeyReturn Then
506         Call ListView3_DblClick
507     End If
508 End Sub
509
510 Private Sub MyClassificationCollection_Change()
511     Dim H As ColumnHeader, B As Boolean
512     Dim N As Long, Bl As Boolean
513
514     With ListView1
515         'ersten hinzufügen
516         If Not IsKeyInCollection("Base", .ColumnHeaders) Then
517             .ColumnHeaders.Add , "Base", "Ausgangsdaten", 200
```

```
514 End If
515
516 'Alle entfernen, die nicht mehr vorkommen
517 Do
518     B = True
519     For Each H In .ColumnHeaders
520         If Not H.Key = "Base" Then
521             B1 = True
522             For N = 1 To MyClassificationCollection.Count
523                 If H.Key =
MyClassificationCollection.ClassificationClass(N) Then
524                     B1 = False
525                 End If
526             Next
527             If B1 Then
528                 B = False
529                 .ColumnHeaders.Remove H.Key
530             Exit For
531             End If
532
533         End If
534     Next
535 Loop Until B
536
537 'die, die fehlen hinzufügen
538 For N = 1 To MyClassificationCollection.Count
539     B = Not
IsKeyInCollection(MyClassificationCollection.ClassificationClass
(N), .ColumnHeaders)
540     If B Then
541         .ColumnHeaders.Add ,
MyClassificationCollection.ClassificationClass(N) ,
MyClassificationCollection.Classification(N).ClassificationName,
100
542     End If
543 Next
544 End With
545 DrawSheet
546 End Sub
547
548 Private Sub Picture4 GotFocus()
549     Picture5.BackColor = vbHighlight
550     Labell1.ForeColor = vbHighlightText
551 End Sub
552
553 Private Sub Picture4 KeyDown(KeyCode As Integer, Shift As
Integer)
554     On Error Resume Next
```

```
555
556 Dim SH As Single, SV As Single
557
558 SH = HScroll11.Value
559 SV = VScroll11.Value
560
561 Select Case KeyCode
562     Case vbKeyLeft:     SH = SH - HScroll11.SmallChange
563     Case vbKeyRight:    SH = SH + HScroll11.SmallChange
564     Case vbKeyUp:       SV = SV - VScroll11.SmallChange
565     Case vbKeyDown:     SV = SV + VScroll11.SmallChange
566     Case vbKeyPageUp:   SV = SV - VScroll11.LargeChange
567     Case vbKeyPageDown: SV = SV + VScroll11.LargeChange
568 End Select
569
570 If SH > HScroll11.Max Then SH = HScroll11.Max
571 If SH < HScroll11.Min Then SH = HScroll11.Min
572 If SV > VScroll11.Max Then SV = VScroll11.Max
573 If SV < VScroll11.Min Then SV = VScroll11.Min
574
575 If HScroll11.Value <> SH Then HScroll11.Value = SH
576 If VScroll11.Value <> SV Then VScroll11.Value = SV
577 End Sub
578
579 Private Sub Picture4_LostFocus()
580     Picture5.BackColor = vbButtonFace
581     Labell1.ForeColor = vbButtonText
582 End Sub
583
584 Private Sub Picture6_Resize()
585     On Error Resume Next
586
587     Dim L1 As Single, L2 As Single
588
589     L1 = Toolbar1.Buttons("del").Left +
Toolbar1.Buttons("del").Width
590     L2 = Picture6.ScaleWidth - ListView3.Width
591     If L2 < L1 Then L2 = L1
592
593     Toolbar1.Width = L2
594     ListView2.Move 0, Toolbar1.Height, L2 - 2 *
Screen.TwipsPerPixelX, Picture6.ScaleHeight - Toolbar1.Height
595     ListView3.Move L2, ListView2.Top, ListView3.Width,
ListView2.Height
596     Toolbar3.Move L2, 0, ListView3.Width
597 End Sub
598
```

```

599 Private Sub Toolbar1_ButtonClick(ByVal Button As
MSComctlLib.Button)
600     On Error Resume Next
601     Dim KKey As String, frm As frmEditKernmarsch
602
603     Select Case Button.Key
604     Case "new"
605         With RSKernmärsche
606             .AddNew
607             .Fields("BohrungID") = BohrungKey
608             .Fields("Teufe") = 0
609             .Fields("Länge") = 1
610             KKey = .Fields("KernmarschID")
611             .Update
612             RefillKernmarschList KKey
613         End With
614         Toolbar1_ButtonClick Toolbar1.Buttons("edt")
615
616     Case "del"
617         If Not ListView2.SelectedItem Is Nothing Then
618             KKey = ListView2.SelectedItem.Key
619             With RSKernmärsche
620                 .MoveFirst
621                 Do Until .EOF
622                     If .Fields("KernmarschID") = KKey Then
623                         If MsgBox("Soll der Kernmarsch bei " &
Format(.Fields("Teufe"), "0.00") & " m gelöscht werden?",
vbQuestion Or vbYesNo Or vbDefaultButton2, "Kernmarsch löschen")
= vbYes Then
624                             .Delete
625                             Call Redraw
626                         End If
627                     Exit Do
628                 End If
629                 .MoveNext
630             Loop
631             End With
632         End If
633
634     Case "edt"
635         If Not ListView2.SelectedItem Is Nothing Then
636             KKey = ListView2.SelectedItem.Key
637
638             Set frm = frmEditKernmarsch
639             frm.Hide
640             With RSKernmärsche.Clone
641                 .MoveFirst
642                 Do Until .EOF

```

```

643         If .Fields("KernmarschID") = KKey Then
644
645             .Edit
646             PutTextBox .Fields("Teufe"), frm.Text3
647             PutTextBox .Fields("Länge"), frm.Text4
648             PutTextBox .Fields("Bruchstücke"), frm.Text5
649             PutTextBox .Fields("Parameter"), frm.Text1:
frm.Text1_KeyUp 0, 0
650             PutTextBox .Fields("Bemerkungen"), frm.Text2
651
652             frm.Show 1, Me
653
654             If frmEditKernmarsch.LastExitOK Then
655                 'Werte in die Datenbank übernehmen
656                 GetTextBox .Fields("Teufe"), frm.Text3
657                 GetTextBox .Fields("Länge"), frm.Text4
658                 GetTextBox .Fields("Bruchstücke"), frm.Text5
659                 GetTextBox .Fields("Parameter"), frm.Text1
660                 GetTextBox .Fields("Bemerkungen"), frm.Text2
661                 .Update
662
663                 Call Redraw
664             End If
665
666             Exit Do
667             End If
668             .MoveNext
669         Loop
670         End With
671
672     End If
673 End Select
674 End Sub
675
676 Private Sub Toolbar2_ButtonClick(ByVal Button As
MSComctlLib.Button)
677     Dim p As Double
678
679     Select Case Button.Key
680     Case "ZoomAll"
681         p = Picture2.ScaleHeight / (Teufe + 3) * 2
682         Comb1 = Fix(p) & " %"
683         Comb1_Click
684
685     Case "Export"
686         Dim FName As String, hMF As Long, MFRect As Rect, MFf As
Long
687         Dim hDCDescriptor As hDCDescriptor

```

```
688
689     If Not (RSBohrungen Is Nothing Or RSKernmärsche Is
Nothing) Then
690         On Error Resume Next
691         With CommonDialog1
692             .CancelError = True
693             .Flags = cdloFNOverwritePrompt Or cdloFNLongNames Or
cdloFNNoChangeDir
694             .Filter = "Windows Meta File (*.wmf)|*.wmf"
695             .ShowSave
696
697         If Err = 0 Then
698             FName = .FileName
699
700             With MFRect
701                 .Left = 0
702                 .Top = 0
703                 .Right = Picture4.ScaleWidth * 40
704                 .Bottom = Picture4.ScaleHeight * 50
705             End With
706
707             hMF = CreateEnhMetaFile(0, FName, MFRect,
App.ProductName & vbNullChar & BohrungName & vbNullChar &
vbNullChar)
708
709             hDCDescriptor = DrawSheetGeneratehDCDescriptor
710             hDCDescriptor.hdc = hMF
711             DrawSheetToHDC hDCDescriptor
712
713             MFf = CloseEnhMetaFile(hMF)
714             Call DeleteEnhMetaFile(MFf)
715
716             DrawSheet
717         End If
718     End With
719 End If
720
721 End Select
722 End Sub
723
724 Private Sub Toolbar3.ButtonClick(ByVal Button As
MSComctlLib.Button)
725     On Error Resume Next
726     Dim KKey As String, OKey As String
727
728     Select Case Button.Key
729     Case "edt"
730         If Not ListView3.SelectedItem Is Nothing Then
```

```
731         OKey = ListView3.SelectedItem.Key
732
733         With RSObjekte.Clone
734             .MoveFirst
735             Do Until .EOF
736                 If .Fields("AbbildungID") = OKey Then
737                     .Edit
738
739                     With frmOLEObjekt.Data1
740                         .DatabaseName = DatabaseFileName
741                         .Refresh
742
743                         .Recordset.MoveFirst
744                         Do Until .Recordset.EOF
745                             If .Recordset.Fields("AbbildungID") = OKey
746
747                             Then
748                                 Exit Do
749                                 End If
750                                 .Recordset.MoveNext
751                                 Loop
752                                 .UpdateControls
753                                 End With
754
755                                 PutTextBox .Fields("Name"),
frmOLEObjekt.Text1
756                                 PutTextBox .Fields("Beschreibung"),
frmOLEObjekt.Text2
757                                 frmOLEObjekt.Show 1, Me
758                                 GetTextBox .Fields("Name"),
frmOLEObjekt.Text1
759                                 GetTextBox .Fields("Beschreibung"),
frmOLEObjekt.Text2
760                                 .Update
761                                 Call Redraw
762                                 Exit Do
763                                 End If
764                                 .MoveNext
765                                 Loop
766                                 End With
767                                 End If
768                                 Case "del"
769                                 If Not ListView3.SelectedItem Is Nothing Then
```

```

770         OKey = ListView3.SelectedItem.Key
771         With RSObjekte.Clone
772             .MoveFirst
773             Do Until .EOF
774                 If .Fields("AbbildungID") = OKey Then
775                     If MsgBox("Soll das Objekt "" & .Fields("Name") &
"" gelöcht werden?", vbQuestion Or vbYesNo Or
vbDefaultButton2, "Objekt löschen") = vbYes Then
776                         .Delete
777                         Call Redraw
778                     End If
779                     Exit Do
780                 End If
781                 .MoveNext
782             Loop
783         End With
784     End If

785     Case "new"
786         If Not ListView2.SelectedItem Is Nothing Then
787             KKey = ListView2.SelectedItem.Key
788
789             Dim NewName As String, NewNameOk As Boolean,
NewNameIndex As Long
790             With RSObjekte.Clone
791                 NewNameIndex = 0
792                 Do
793                     NewNameIndex = NewNameIndex + 1
794                     NewName = "Objekt " & NewNameIndex
795                     NewNameOk = True
796                     .MoveFirst
797                     Do Until .EOF
798                         If .Fields("Name") = NewName Then
799                             NewNameOk = False
800                             Exit Do
801                         End If
802                     .MoveNext
803                 Loop
804                 Loop Until NewNameOk
805             End With
806
807             If MsgBox("Soll dem Kernmarsch bei " &
Trim(ListView2.SelectedItem.SubItems(1)) & " m ein neues Objekt
hinzu gefügt werden?", vbYesNo Or vbQuestion Or
vbDefaultButton2, "Objekt hinzufügen") = vbYes Then
809                 With RSObjekte
810                     .AddNew

```

```

811             .Fields("KernmarschID") = KKey
812             .Fields("Name") = NewName
813             OKey = .Fields("AbbildungID")
814             .Update
815             RefillKernmarschList KKey, OKey
816         End With
817         Toolbar3_ButtonClick Toolbar3.Buttons("edt")
818     End If
819 End If

820
821     End Select
822 End Sub

823
824 Private Sub UserControl_Initialize()
825     Set Toolbar1.ImageList = ImageList1
826     With Toolbar1.Buttons
827         .Add(, "new", "Kernmarsch hinzufügen", , "new").ToolTipText
= "neuen Kernmarsch erstellen"
828         .Add(, "edt", "Kernmarsch bearbeiten", , "edt").ToolTipText
= "Eigenschaften des markierten Kernmarsches anzeigen (Enter)"
829         .Add(, , tbrSeparator
830         .Add(, "del", "Kernmarsch entfernen", , "del").ToolTipText =
"markierten Kernmarsch löschen (Entf)"
831     End With

832
833     Set MyClassificationCollection = ClassificationCollection
834
835     Set Toolbar2.ImageList = ImageList2
836     With Toolbar2.Buttons
837         .Add(, "ZoomAll", "Ganze Seite ", ,
"ZoomAlles").ToolTipText = "Zeigt das gesamte Arbeitsblatt an"
838         .Add(, "Zoom", , tbrPlaceholder).Width = Combol.Width
839         .Add(, , tbrSeparator
840         .Add(, "Export", "Exportieren...", , "export").ToolTipText =
"Exportiert die aktuelle Ansicht in eine Datei."
841     End With

842
843     Set Toolbar3.ImageList = ImageList3
844     With Toolbar3.Buttons
845         .Add(, "new", , , "new").ToolTipText = "neues OLE-Objekt
erstellen"
846         .Add(, "edt", , , "edt").ToolTipText = "Markiertes OLE-
Objekt anzeigen (Enter)"
847         .Add(, , , tbrSeparator
848         .Add(, "del", , , "del").ToolTipText = "markiertes OLE-
Objekt löschen (Entf)"
849     End With
850

```

```
851 Combol.AddItem "10 %"
852 Combol.AddItem "20 %"
853 Combol.AddItem "50 %"
854 Combol.AddItem "75 %"
855 Combol.AddItem "100 %"
856 Combol.AddItem "120 %"
857 Combol.AddItem "150 %"
858 Combol.AddItem "175 %"
859 Combol.AddItem "200 %"
860 Combol.AddItem "250 %"
861 Combol.AddItem "500 %"
862 Combol.Text = "100 %"
863
864 Set Sheet = New prjSheet.Sheet
865 With Sheet
866 .AddCircle 50, 50, 30, 0, False, 255
867 End With
868 End Sub
869
870 Private Sub UserControl_Resize()
871 On Error Resume Next
872 Dim T As Double
873
874 T = Toolbar2.Top + Toolbar2.Height + Screen.TwipsPerPixelY * 3
875 Picture6.Move 0, ScaleHeight - Picture6.Height, ScaleWidth
876
877 Dim H As Double
878 NfoPanelLabel.Width = Picture6.Width - NfoPanelLabel.Left -
NfoPanelLabel.Top
879 NfoPanelLabel.AutoSize = True
880 H = NfoPanelLabel.Height + 2 * NfoPanelLabel.Top
881 If H < NfoPanelImage.Height + 2 * NfoPanelImage.Top Then H =
NfoPanelImage.Height + 2 * NfoPanelImage.Top
882
883 NfoPanel.Visible = ShowNfoPanel
884 If Not ShowNfoPanel Then H = 0
885
886 Picture1.Move 0, T, ScaleWidth, ScaleHeight - Picture6.Height
- T - H
887 NfoPanel.Move 0, Picture1.Top + Picture1.Height,
Picture1.Width, H
888
889 With Toolbar2.Buttons("Zoom")
890 Combol.Left = .Left
891 End With
892 End Sub
893
894 Private Sub HScroll1_Change()
```

```
895 On Error Resume Next
896
897 If HScroll1.Enabled Then
898 Picture5.Left = -HScroll1.Value
899 Else
900 Picture5.Left = 0
901 End If
902
903 Dim N As Long, C As ColumnHeader, R As Long, B As Boolean
904 Dim a As Long, W As Single, DoRefill As Boolean
905 With ListView1
906 If .ColumnHeaders.Count > 1 Then
907 R = .ColumnHeaders(2).Left - .ColumnHeaders(1).Width
908 Else
909 R = 0
910 End If
911 N = 0
912 For Each C In .ColumnHeaders
913 N = N + C.Width + R
914 Next
915 W = .Left + N
916 If W <> Picture4.Width Then
917 Picture4.Width = W
918 DoRefill = True
919 End If
920 Picture3.Move Picture3.Left, 17, W, Picture2.ScaleHeight -
Picture3.Top
921 Picture5.Move Picture5.Left, Picture5.Top, Picture3.Width,
Picture2.ScaleHeight
.Width = N
922
923
924 With HScroll1
925 N = Picture4.Width - Picture2.ScaleWidth
926 If N < 0 Then N = 0
927 B = N > 0
928 HScroll1.Enabled = B
929 If B Then
930 If .Max <> N Then
931 .Max = N
932 End If
933
934 a = Picture2.ScaleWidth / 3
935 If .LargeChange <> a Then
936 .LargeChange = a
937 End If
938
939 a = Picture2.ScaleWidth / 10
940 If .SmallChange <> a Then
```

```
941         .SmallChange = a
942     End If
943 End With
944 End With
945 End With
946
947 With VScroll1
948     N = Picture4.Height - Picture3.ScaleHeight + 5
949     If N < 0 Then N = 0
950     B = N > 0
951     .Enabled = B
952     If B Then
953         If .Max <> N Then
954             .Max = N
955         End If
956
957         a = Picture2.ScaleHeight / 3
958         If .LargeChange <> a Then
959             .LargeChange = a
960         End If
961
962         a = Picture2.ScaleHeight / 10
963         If .SmallChange <> a Then
964             .SmallChange = a
965         End If
966     End If
967 End With
968
969 If DoRefill Then
970     RefillSheet
971 End If
972 End Sub
973
974 Private Sub HScroll1 Scroll()
975     Call HScroll1 Change
976 End Sub
977
978 Private Sub Picture1 Resize()
979     On Error Resume Next
980     Dim SW As Long, SH As Long
981
982     SW = SysInfo1.ScrollBarSize / Screen.TwipsPerPixelX
983     SH = SysInfo1.ScrollBarSize / Screen.TwipsPerPixelY
984     With Picture1
985         VScroll1.Move .ScaleWidth - SW, 0, .ScaleHeight - SH
986         HScroll1.Move 0, .ScaleHeight - SH, .ScaleWidth - SW, SH
987         Picture2.Move 0, 0, .ScaleWidth - SW, .ScaleHeight - SH
988     End With
```

```
989     HScroll1_Change
990 End Sub
991
992 Private Sub VScroll1 Change()
993     If VScroll1.Enabled Then
994         Picture4.Top = -VScroll1.Value
995     Else
996         Picture4.Top = 0
997     End If
998     Call HScroll1 Change
999 End Sub
1000
1001 Private Sub VScroll1_Scroll()
1002     Call VScroll1_Change
1003 End Sub
1004
1005 Private Sub Timer1 Timer()
1006     VScroll1 Change
1007 End Sub
1008
1009 Public Sub SetBohrung(BohrungenRS As DAO.Recordset,
1010     KernmärscheRS As DAO.Recordset, ObjekteRS As DAO.Recordset,
1011     strBohrungKey As String, strDatabaseFileName As String)
1012     If Not (RSBohrungen Is BohrungenRS And KernmärscheRS Is
1013         RSKernmärsche And BohrungKey = strBohrungKey) Then
1014         'neue Bohrung
1015
1016         Set RSBohrungen = BohrungenRS
1017         Set RSKernmärsche = KernmärscheRS
1018         Set RSObjekte = ObjekteRS
1019         BohrungKey = strBohrungKey
1020         DatabaseFileName = strDatabaseFileName
1021     End If
1022     Redraw
1023 End Sub
1024
1025 Private Sub Redraw()
1026     On Error Resume Next
1027     Dim OldKMKey As String
1028
1029     OldKMKey = vbNullString
1030     If Not ListView2.SelectedItem Is Nothing Then
1031         OldKMKey = ListView2.SelectedItem.Key
1032     End If
1033
1034     Dim BM As Variant
1035     BM = vbNullString
```

```
1033 RSB Bohrungen.MoveFirst
1034 Do Until RSB Bohrungen.EOF
1035     If RSB Bohrungen.Fields("BohrungID") = BohrungKey Then
1036         BM = RSB Bohrungen.Bookmark
1037         Exit Do
1038     End If
1039     RSB Bohrungen.MoveNext
1040 Loop
1041
1042 If Not IsEmpty(BM) Then
1043     Labell = RSB Bohrungen.Fields("Name").Value
1044     RefillKernmarschList OldKMKey
1045 Else
1046     Labell = "unbekannte Bohrung"
1047 End If
1048
1049 RefillSheet
1050 End Sub
1051
1052 Private Sub RefillKernmarschList(SelectKernmarsch As String,
Optional ByVal SelectObject As String)
1053     On Error Resume Next
1054
1055     If SelectObject = vbNullString Then
1056         SelectObject = ListView3.SelectedItem.Key
1057     End If
1058
1059     ListView2.ListItems.Clear
1060     ListView3.ListItems.Clear
1061     If Not RSKernmarsche Is Nothing Then
1062         RSKernmarsche.MoveFirst
1063         Do Until RSKernmarsche.EOF
1064             If CStr(RSKernmarsche.Fields("BohrungID")) = BohrungKey
Then
1065                 With ListView2.ListItems.Add(,
RSKernmarsche.Fields("KernmarschID"),
FormatDouble(ListView2.ListItems.Count + 1, "0", 5))
1066                     .SubItems(1) =
FormatDouble(RSKernmarsche.Fields("Teufe"), "0.00", 7) & " m"
1067                     .SubItems(2) =
FormatDouble(RSKernmarsche.Fields("Länge"), "0.00", 7) & " m"
1068                     .SubItems(3) = RSKernmarsche.Fields("Bemerkungen")
1069
1070                 If Not RSObjekte Is Nothing Then
1071                     RSObjekte.MoveFirst
1072                     Do Until RSObjekte.EOF
1073                         If CStr(RSObjekte.Fields("KernmarschID")) =
CStr(RSKernmarsche.Fields("KernmarschID")) Then
```

```
1074                 With ListView3.ListItems.Add(,
RSObjekte.Fields("AbbildungID"), RSObjekte.Fields("Name") &
vbNullString)
1075                     .SubItems(1) =
FormatDouble(ListView2.ListItems.Count, "0", 5)
1076                     If .Key = SelectObject Then
1077                         .Selected = True
1078                     End If
1079                 End With
1080             End If
1081             RSObjekte.MoveNext
1082         Loop
1083     End If
1084
1085     If .Key = SelectKernmarsch Then
1086         .Selected = True
1087     End If
1088 End With
1089 End If
1090
1091     RSKernmarsche.MoveNext
1092 Loop
1093 End If
1094 End Sub
1095
1096 Private Sub RefillSheet()
1097     On Error Resume Next
1098     Dim N As Long, TH As Double
1099     Dim pTeufe As Double, p As Double, p0 As Double
1100     Dim RXC As RockClassX.clsClassification
1101     Dim B1 As Double, B2 As Double, Dict As Scripting.Dictionary
1102
1103     Const ColRed As Double = 0.95
1104
1105     Dim NfoOutOfRangeFound As Boolean
1106     Dim NfoPanelString As String
1107     NfoPanelString = vbNullString
1108
1109     If (RSBohrungen Is Nothing Or RSKernmarsche Is Nothing) Then
1110         If RSB Bohrungen Is Nothing Then
1111             NfoPanelString = NfoPanelString & "Tabelle der Bohrung
nicht gesetzt. "
1112         End If
1113         NfoPanelString = NfoPanelString & "Tabelle der Kernmarsche
nicht gefunden. " & vbCrLf
1114     Else
1115         'nach Bohrung suchen
1116         pTeufe = Teufe
```

```

1117     If pTeufe = 0 Then
1118         NfoPanelString = NfoPanelString & "Es wurde keine Teufe
angegeben. " & vbCrLf
1119     End If
1120     If pTeufe < 0 Then
1121         NfoPanelString = NfoPanelString & "Die Teufe der Bohrung
ist negativ. " & vbCrLf
1122     End If
1123
1124     Dim SheetFont As SheetFont, TextFormat As TextFormatType
1125     Dim OX As Double, D1 As Double, D2 As Double, Str As String
1126     Dim ColumnHeader As ColumnHeader, Columns() As ColumnType
1127
1128     If ListView1.ColumnHeaders.Count = 0 Then
1129         Call MyClassificationCollection.Change
1130         If ListView1.ColumnHeaders.Count = 0 Then
1131             Exit Sub
1132         End If
1133     End If
1134
1135     'Spalten suchen
1136     With ListView1
1137         ReDim Columns(1 To .ColumnHeaders.Count)
1138
1139         D1 = .Left - Picture4.Left - 50
1140         If .ColumnHeaders.Count > 1 Then
1141             D2 = .ColumnHeaders(2).Left - .ColumnHeaders(1).Width
1142         Else
1143             NfoPanelString = NfoPanelString & "Spalten nicht
gefunden. "
1144             D2 = 0
1145         End If
1146         For N = 1 To .ColumnHeaders.Count
1147             For Each ColumnHeader In .ColumnHeaders
1148                 If ColumnHeader.Position = N Then
1149                     With Columns(ColumnHeader.Index)
1150                         .pos = D1
1151                         .Width = ColumnHeader.Width
1152                         .Key = ColumnHeader.Key
1153                         Set .Classification =
ClassificationCollection.ClassificationClassByClassName(.Key)
1154                     End With
1155                     D1 = D1 + ColumnHeader.Width + D2
1156                 End If
1157             Next
1158         Next
1159     End With
1160

```

```

1161     'Los gehts
1162     Const BarWidth As Double = 50
1163     NextSheetHeight = Fix(pTeufe) - (pTeufe <> Fix(pTeufe))
1164
1165     With Sheet
1166         .Clear
1167         OX = -40
1168         TextFormat.Align = vbCenter
1169         SheetFont.Bold = True
1170         .AddText OX, -0.6, Columns(1).pos - OX, 20,
RSBohrungen.Fields("Name"), SheetFont, TextFormat
1171
1172         TextFormat.Align = vbLeftJustify
1173         SheetFont.Bold = False
1174
1175         'Kote
1176         .AddLine OX + 0, 0, OX + 10, 0, 0
1177         For N = 1 To NextSheetHeight
1178             .AddLine OX + 0, (N), OX + 10, (N), 0
1179             .AddBoxXY OX + 0, (N - 1), OX + 5, (N), 0, N Mod 2, 0
1180             .AddText OX + 10, (N - 0.5), OX + 10, 1, CStr(N),
SheetFont, TextFormat
1181         Next
1182
1183         'Säule
1184         .AddLine OX, 0, Columns(UBound(Columns)).pos +
Columns(UBound(Columns)).Width, 0, 0
1185         .AddBoxXY 0, 0, BarWidth, pTeufe, 0, False, &HCCDDEE
1186
1187         Dim SFont As SheetFont
1188         SFont.Name = "Arial"
1189         SFont.Size = 10
1190
1191         Dim SFormat As TextFormatType
1192         SFormat.Align = vbLeftJustify
1193         SFormat.Wordbreak = True
1194         SFormat.EndEllipses = True
1195         SFormat.Clip = True
1196
1197         For N = 1 To UBound(Columns)
1198             If Columns(N).Key <> "Base" Then
1199                 .AddBoxXY Columns(N).pos, 0, Columns(N).pos +
Columns(N).Width * ColRed, pTeufe, 0, False, &HEEEFFF
1200                 .AddText Columns(N).pos, -0.4, Columns(N).pos +
Columns(N).Width * ColRed, 1,
Columns(N).Classification.ClassificationName, SFont, SFormat
1201             End If
1202         Next

```

```

1203
1204     RSKernmärsche.MoveFirst
1205     Do Until RSKernmärsche.EOF
1206         If RSKernmärsche.Fields("BohrungID") = BohrungKey Then
1207             'dieser Kernmarsch gehört dazu
1208             D1 = RSKernmärsche.Fields("Teufe")
1209             D2 = RSKernmärsche.Fields("Länge")
1210
1211             .AddBoxXY 0, D1, BarWidth, D1 + D2, 0, False, &HAAEECC
1212
1213             'Bereich prüfen
1214             If Not NfoOutOfRangeFound Then
1215                 If D1 < 0 Or D1 + D2 > pTeufe Then
1216                     NfoPanelString = NfoPanelString & "Einer oder
mehrere Kernmärsche reichen über den Teufenbereich hinaus. "
1217                 End If
1218             End If
1219
1220             Str = RSKernmärsche.Fields("Bruchstücke") &
vbNullString
1221             If Str = vbNullString Then Str = "(keine Bruchstücke)"
1222             Str = Str & "; " & RSKernmärsche.Fields("Parameter") &
vbCrLf & RSKernmärsche.Fields("Bemerkungen")
1223             Str = Replace(Str, vbCrLf & vbCrLf, vbCrLf)
1224             If Len(Str) > 2 Then
1225                 If Right(Str, 2) = vbCrLf Then Str = Left(Str,
Len(Str) - 2)
1226             End If
1227             .AddText Columns(1).pos, D1, Columns(1).Width *
ColRed, D2, Str, SFont, SFormat
1228
1229             Set Dict = getParameterDictionary(RSKernmärsche)
1230             For N = 1 To UBound(Columns)
1231                 If Columns(N).Key <> "Base" Then
1232                     Set RXC = Columns(N).Classification
1233                     p = 0
1234                     p0 = 0
1235                     p0 = RXC.Calculate(Dict)
1236                     B1 = RXC.ValueRangeMin()
1237                     B2 =
RXC.ValueRangeMax(Val(RSKernmärsche.Fields("Länge") &
vbNullString) * 100)
1238                     If B1 <> B2 Then
1239                         p = (p0 - B1) / (B2 - B1)
1240                     Else
1241                         p = p0
1242                     End If
1243                     If p > 1 Then p = 1

```

```

1244             If p < 0 Then p = 0
1245             .AddBoxXY Columns(N).pos, D1, Columns(N).pos +
Columns(N).Width * p * ColRed, D1 + D2, 0, False, &HDDDDDD
1246             .AddText Columns(N).pos, D1, Columns(N).pos +
Columns(N).Width * ColRed, D2, CStr(" " & Format(p0,
RXC.FormatString)), SFont, SFormat
1247             End If
1248             Next
1249             End If
1250
1251             RSKernmärsche.MoveNext
1252             Loop
1253
1254             If ListView2.ListItems.Count = 0 Then
1255                 NfoPanelString = NfoPanelString & "Es konnten für die
Bohrung keine Kernmärsche gefunden werden. Bitte fügen sie
welche hinzu."
1256             End If
1257
1258             End With
1259             End If
1260
1261             ShowNfoPanel = Not (NfoPanelString = vbNullString)
1262             NfoPanelLabel = NfoPanelString
1263
1264             UserControl Resize
1265             DrawSheet
1266         End Sub
1267
1268         Public Property Get Teufe() As Double
1269             With RSBohrungen.Clone
1270                 .MoveFirst
1271                 Do Until .EOF
1272                     If .Fields("BohrungID") = BohrungKey Then
1273                         Teufe = CMe(.Fields("Teufe") & vbNullString)
1274                     Exit Do
1275                 End If
1276                 .MoveNext
1277             Loop
1278             End With
1279         End Property
1280
1281         Public Property Get BohrungName() As String
1282             With RSBohrungen.Clone
1283                 .MoveFirst
1284                 Do Until .EOF
1285                     If .Fields("BohrungID") = BohrungKey Then
1286                         BohrungName = .Fields("Name") & vbNullString

```

```

1287         Exit Do
1288     End If
1289     .MoveNext
1290     Loop
1291 End With
1292 End Property
1293 Private Sub DrawSheet()
1294     On Error Resume Next
1295     DrawSheetToHDC DrawSheetGeneratehDCDescriptor
1296 End Sub
1297 Private Function DrawSheetGeneratehDCDescriptor() As
hDCDescriptor
1298     On Error Resume Next
1299     Dim hDCDescriptor As hDCDescriptor, p As Double
1300
1301     p = CMe(Combo1.Text) / 2
1302     If p < 5 Then p = 5
1303     If p > 250 Then p = 250
1304
1305     With Picture4
1306         .Cls
1307
1308         With hDCDescriptor
1309             .OffsetX = 50
1310             .OffsetY = 1
1311             .ZoomX = 1
1312             .ZoomY = p
1313             .FontZoom = .ZoomX
1314             .OffsetY = 1 * .ZoomY
1315         End With
1316         .Height = NextSheetHeight * hDCDescriptor.ZoomY +
hDCDescriptor.OffsetY + 20
1317         With hDCDescriptor
1318             .hdc = Picture4.hdc
1319         End With
1320
1321         DrawSheetGeneratehDCDescriptor = hDCDescriptor
1322     End With
1323 End Function
1324 Private Sub DrawSheetToHDC(hDCDescriptor As hDCDescriptor)
1325     On Error Resume Next
1326     Sheet.RenderToHDC hDCDescriptor
1327 End Sub
1328
1329 Public Sub KernmarschAdd()
1330     Toolbar1.ButtonClick Toolbar1.Buttons("add")
1331 End Sub
1332

```

```

1333 Public Sub KernmarschEdit()
1334     Toolbar1.ButtonClick Toolbar1.Buttons("edt")
1335 End Sub
1336
1337 Public Sub KernmarschDel()
1338     Toolbar1.ButtonClick Toolbar1.Buttons("del")
1339 End Sub
1340

```

Datei: ctlParameter.ctl (VB User Control)

```

01 VERSION 5.00
02 Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0";
"MSCOMCTL.OCX"
03 Object = "{5E9E78A0-531B-11CF-91F6-C2863C385E30}#1.0#0";
"MSFLXGRD.OCX"
04 Begin VB.UserControl ctlParameter
05     ClientHeight = 4425
06     ClientLeft = 0
07     ClientTop = 0
08     ClientWidth = 4800
09     ScaleHeight = 4425
10     ScaleWidth = 4800
11     Begin VB.PictureBox InfoPic
12         Appearance = 0 '2D
13         BackColor = &H80000018&
14         BorderStyle = 0 'Kein
15         ForeColor = &H80000008&
16         Height = 855
17         Left = 120
18         ScaleHeight = 855
19         ScaleWidth = 4455
20         TabIndex = 4
21         Top = 3240
22         Width = 4455
23     Begin VB.CommandButton InfoPicCmd
24         Appearance = 0 '2D
25         Caption = "bestimmen..."
26         Height = 300
27         Left = 3120
28         TabIndex = 6
29         Top = 480
30         Width = 1215
31     End
32     Begin VB.Label InfoPicLabel
33         AutoSize = -1 'True
34         BackStyle = 0 'Transparent

```

```

35     Caption      = "Labell1"
36     ForeColor    = &H80000017&
37     Height       = 195
38     Left         = 360
39     TabIndex     = 5
40     Top          = 60
41     Width        = 3975
42     WordWrap     = -1 'True
43 End
44 Begin VB.Image InfoPicImage
45     Height       = 240
46     Left         = 60
47     Picture      = "ctlParameter.ctx":0000
48     Top          = 60
49     Width        = 240
50 End
51 End
52 Begin MSComctlLib.ImageList ImageList1
53     Left         = 4200
54     Top          = 0
55     ExtentX      = 1005
56     ExtentY      = 1005
57     BackColor    = -2147483643
58     ImageWidth   = 16
59     ImageHeight  = 16
60     MaskColor    = 12632256
61     Version      = 393216
62     BeginProperty Images {2C247F25-8591-11D1-B16A-
00C0F0283628}
63         NumListImages = 2
64         BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-
00C0F0283628}
65             Picture      = "ctlParameter.ctx":014A
66             Key          = "new"
67         EndProperty
68         BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-
00C0F0283628}
69             Picture      = "ctlParameter.ctx":059C
70             Key          = "del"
71         EndProperty
72     EndProperty
73 End
74 Begin MSComctlLib.Toolbar Toolbar1
75     Align        = 1 'Oben ausrichten
76     Height       = 330
77     Left         = 0
78     TabIndex     = 3
79     Top          = 0

```

```

80     Width        = 4800
81     _ExtentX     = 8467
82     ExtentY      = 582
83     ButtonWidth  = 609
84     ButtonHeight = 582
85     AllowCustomize = 0 'False
86     Wrappable    = 0 'False
87     Style        = 1
88     TextAlignment = 1
89     Version      = 393216
90 End
91 Begin VB.ComboBox Combo1
92     Appearance   = 0 '2D
93     Height       = 315
94     Left         = 240
95     TabIndex     = 2
96     Text         = "Combo1"
97     Top          = 2040
98     Visible      = 0 'False
99     Width        = 1935
100 End
101 Begin VB.TextBox Text1
102     BorderStyle  = 0 'Kein
103     Height       = 195
104     Left         = 2280
105     TabIndex     = 1
106     Text         = "Text1"
107     Top          = 2040
108     Visible      = 0 'False
109     Width        = 1695
110 End
111 Begin MSFlexGridLib.MSFlexGrid MSFlexGrid1
112     Height       = 2415
113     Left         = 120
114     TabIndex     = 0
115     Top          = 720
116     Width        = 4455
117     ExtentX      = 7858
118     ExtentY      = 4260
119     Version      = 393216
120     Rows         = 5
121     FixedCols    = 0
122     ScrollTrack  = -1 'True
123     FocusRect    = 0
124 End
125 End
126 Attribute VB Name = "ctlParameter"
127 Attribute VB_GlobalNameSpace = False

```

```
128 Attribute VB_Creatable = True
129 Attribute VB_PredeclaredId = False
130 Attribute VB_Exposed = False
131 Option Explicit
132
133 Public Event Change()
134 Public ClassificationCollection As clsClassCol
135
136 Private WithEvents EditGrid1 As clsEditGrid
137 Attribute EditGrid1.VB_VarHelpID = -1
138 Private StdParameter As Collection
139 Private RaiseChange As Boolean
140 Private InfoPicClassification As clsClassification
141
142 Private Sub EditGrid1_AfterCellEdit(ByVal Row As Long, ByVal Col
As Long, NewValue As String, Cancel As Boolean)
143     RaiseChange = True
144 End Sub
145
146 Private Sub EditGrid1_CellClear(ByVal Row As Long, ByVal Col As
Long)
147     RaiseEvent Change
148 End Sub
149
150 Private Sub EditGrid1_ProcessReady()
151     RaiseChange = False
152     RaiseEvent Change
153 End Sub
154
155 Private Sub InfoPicCmd_Click()
156     Dim V As Variant
157     If Not InfoPicClassification Is Nothing Then
158         With MSFlexGrid1
159             V =
160 InfoPicClassification.ShowParameterWindow(.TextMatrix(.Row, 0),
.TextMatrix(.Row, 1), Me)
161             .TextMatrix(.Row, 1) = V
162         End With
163         EditGrid1.Refresh
164         RaiseEvent Change
165     End If
166 End Sub
167
168 Private Sub MSFlexGrid1_EnterCell()
169     Dim Parameter As String, N As Long, M As Long
170     Dim PList As Scripting.Dictionary, Description As String
171     With MSFlexGrid1
```

```
172         Parameter = Trim(.TextMatrix(.Row, 0))
173     End With
174
175     Description = vbNullString
176     Set InfoPicClassification = Nothing
177     If Not ClassificationCollection Is Nothing Then
178         With ClassificationCollection
179             For N = 1 To .Count
180                 Set PList = .Classification(N).ParameterList
181                 If PList.Exists(Parameter) Then
182                     Description = PList.Item(Parameter)
183                     Set InfoPicClassification = .Classification(N)
184                     Exit For
185                 End If
186             Next
187         End With
188     End If
189
190     If Parameter = vbNullString Or InfoPicClassification Is
Nothing Then
191         InfoPicLabel = "unbekannter Parameter."
192         InfoPicCmd.Enabled = False
193     Else
194         InfoPicLabel = "gewählter Parameter: " & Parameter
195         If Description <> vbNullString Then
196             InfoPicLabel = InfoPicLabel & vbCrLf & Description
197         End If
198         InfoPicCmd.Enabled =
199 InfoPicClassification.ShowParameterWindow(Parameter, Empty,
Nothing, True)
200     End If
201     Call UserControl.Resize
202 End Sub
203
204 Private Sub MSFlexGrid1_GotFocus()
205     MSFlexGrid1.BackColorSel = vbHighlight
206 End Sub
207 Private Sub MSFlexGrid1_LostFocus()
208     MSFlexGrid1.BackColorSel = vbButtonFace
209 End Sub
210
211 Private Sub MSFlexGrid1_RowColChange()
212     Select Case MSFlexGrid1.Col
213     Case 0:         Set EditGrid1.InputObject = Comb0
214     Case Else:     Set EditGrid1.InputObject = Text1
215     End Select
216 End Sub
```

```

217 Private Sub MSFlexGrid1_SelChange()
218     With MSFlexGrid1
219         .Row = .RowSel
220         .Col = .ColSel
221     End With
222 End Sub
223
224 Private Sub Toolbar1_ButtonClick(ByVal Button As
MSComctlLib.Button)
225     With MSFlexGrid1
226         Select Case Button.Key
227             Case "new"
228                 .Rows = .Rows + 1
229                 .Row = .Rows - 1
230             Case "del"
231                 If MsgBox("Soll der Parameter "" & .TextMatrix(.Row, 0)
& "" (Zeile " & .Row & ") gelöscht werden?", vbQuestion Or
vbYesNo, "Parameter löschen") = vbYes Then
232                     EditGrid1.RowsDel .Row, .Row
233                 End If
234             End Select
235         End With
236     End Sub
237
238 Private Sub UserControl_Initialize()
239     Set StdParameter = New Collection
240
241     Call MSFlexGrid1_LostFocus
242     With Toolbar1
243         Set .ImageList = ImageList1
244         .Buttons.Add(, "new", "Parameter hinzufügen", ,
"new").ToolTipText = "neuen Parameter erstellen"
245         .Buttons.Add(, "del", "Parameter entfernen", ,
"del").ToolTipText = "markierten Parameter löschen"
246     End With
247
248     Set EditGrid1 = New clsEditGrid
249     With EditGrid1
250         Set .MSFlexGrid = MSFlexGrid1
251         Call MSFlexGrid1_RowColChange
252         .GridAutoRowAdd = False
253         .GridIgnoreClipboardKeys = True
254         .GridAutoFocus_ColStart = 1
255     End With
256
257     With MSFlexGrid1
258         .TextMatrix(0, 0) = "Parameter"
259         .TextMatrix(0, 1) = "Wert"

```

```

260
261         .ColAlignment(0) = flexAlignLeftTop
262         .ColAlignment(1) = flexAlignLeftTop
263
264         .Col = 1
265     End With
266 End Sub
267
268 Private Sub UserControl_Resize()
269     On Error Resume Next
270     Dim H As Double
271
272     InfoPicLabel.Width = ScaleWidth - InfoPicLabel.Left -
InfoPicLabel.Top
273     InfoPicLabel.AutoSize = True
274     H = InfoPicLabel.Top * 2.5 + InfoPicCmd.Height * (-
InfoPicCmd.Enabled) + InfoPicLabel.Height
275     If H < 750 Then H = 750
276
277     MSFlexGrid1.Move 0, Toolbar1.Height, ScaleWidth, ScaleHeight -
Toolbar1.Height - H
278
279     InfoPic.Move 0, MSFlexGrid1.Top + MSFlexGrid1.Height,
ScaleWidth, H
280     InfoPicCmd.Move InfoPic.ScaleWidth - InfoPicCmd.Width -
InfoPicLabel.Top, InfoPic.ScaleHeight - InfoPicCmd.Height -
InfoPicLabel.Top
281     InfoPicCmd.Visible = InfoPicCmd.Enabled
282
283     MSFlexGrid1.ColWidth(1) = MSFlexGrid1.Width -
MSFlexGrid1.ColWidth(0) - (GetSystemMetrics(SM_CXFIXEDFRAME) * 2
+ 2 + GetSystemMetrics(SM_CXVSCROLL)) * Screen.TwipsPerPixelX
284 End Sub
285
286 Private Sub UserControl_Terminate()
287     Set EditGrid1 = Nothing
288 End Sub
289
290 Public Property Get Parameter() As Scripting.Dictionary
291     On Error Resume Next
292     Dim N As Long
293     Dim T1 As String, T2 As String
294
295     EditGrid1.EndEdit
296     Set Parameter = New Scripting.Dictionary
297     Parameter.CompareMode = TextCompare
298     With MSFlexGrid1
299         For N = 1 To .Rows - 1

```

```

300     T1 = .TextMatrix(N, 0)
301     T2 = .TextMatrix(N, 1)
302
303     If T1 <> vbNullString Then
304         Err.Clear
305         Parameter.Add T1, T2
306         If Err <> 0 Then
307             MsgBox "Fehler: " & Err.Description & " (Zeile " & N &
308             ")", vbExclamation
309         End If
310     Else
311         MsgBox "Fehlender Parametername." & " (Zeile " & N &
312         ")", vbExclamation
313     End If
314 Next
315 End Property
316 Public Property Set Parameter(Param As Scripting.Dictionary)
317     Dim SP As Variant, N As Long, V As Variant, B As Boolean
318     Dim C As Long
319     EditGrid1.EndEdit
320     With MSFlexGrid1
321         .Rows = 1
322         C = 0
323
324         For Each SP In StdParameter
325             C = C + 1
326             If .Rows - 1 < C Then .Rows = C + 1
327
328             .TextMatrix(C, 0) = SP
329             .TextMatrix(C, 1) = GetItem(Param, CStr(SP), vbNullString,
330             False)
331         Next
332
333         If Not Param Is Nothing Then
334             For Each SP In Param
335                 B = True
336                 For N = 1 To .Rows - 1
337                     If LCase(.TextMatrix(N, 0)) = LCase(SP) Then
338                         B = False: Exit For
339                     End If
340                 Next
341
342                 If B Then
343                     C = C + 1
344                     If .Rows - 1 < C Then .Rows = C + 1
345                     .TextMatrix(C, 0) = SP

```

```

345         .TextMatrix(C, 1) = GetItem(Param, CStr(SP),
346         vbNullString, False)
347     End If
348 Next
349 End With
350
351     EditGrid1.Refresh
352     MSFlexGrid1.EnterCell
353 End Property
354
355 Public Property Set StandardParameter(StdParam As Collection)
356     Set StdParameter = StdParam
357
358     Dim SP As Variant
359     With Combol
360         .Clear
361         For Each SP In StdParameter
362             .AddItem SP
363         Next
364     End With
365 End Property
366
367 Public Function GetItem(Parameter As Scripting.Dictionary, Key
368     As String, StandardValue As Variant, RaiseError As Boolean)
369     'Sucht nach einem Eintrag in dem Dictionary
370     'löst einen Fehler aus, falls nicht gefunden (für
371     RaiseError=True)
372
373     Dim K As Variant, B As Boolean
374
375     GetItem = StandardValue
376     If Not Parameter Is Nothing Then
377         With Parameter
378             B = False
379             For Each K In .Keys
380                 If LCase(Trim(Key)) = LCase(Trim(K)) Then
381                     B = True
382                     GetItem = .Item(K)
383                     Exit For
384                 End If
385             Next
386         End With
387
388         If Not B Then
389             If RaiseError Then
390                 Err.Raise RockClassXErr.RCXErr MissingParameter,
391                 App.ProductName, "Parameter nicht gefunden: "" & Key & """"

```

```

389     End If
390     End If
391     End If
392     End Function
393

```

Datei: frmMain.frm (VB Form)

```

01  VERSION 5.00
02  Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0";
03  "MSCOMCTL.OCX"
04  Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0";
05  "COMDLG32.OCX"
06  Begin VB.Form frmMain
07      Caption           = "frmMain"
08      ClientHeight      = 6600
09      ClientLeft        = 165
10      ClientTop         = 735
11      ClientWidth       = 10830
12      Icon              = "frmMain.frx":0000
13      LinkTopic         = "Form1"
14      ScaleHeight       = 6600
15      ScaleWidth        = 10830
16      StartupPosition   = 3 'Windows-Standard
17      WindowState       = 2 'Maximiert
18      Begin VB.PictureBox Picture1
19          Align          = 1 'Oben ausrichten
20          BackColor     = &H80000010&
21          Height        = 15
22          Left          = 0
23          ScaleHeight   = 15
24          ScaleWidth    = 10830
25          TabIndex      = 9
26          Top           = 330
27          Width         = 10830
28      End
29      Begin MSComctlLib.ImageList ImageList2
30          Left          = 7920
31          Top           = 480
32          ExtentX       = 1005
33          ExtentY       = 1005
34          BackColor     = -2147483643
35          ImageWidth    = 16
36          ImageHeight   = 16
37          MaskColor     = 12632256
38          _Version      = 393216

```

```

37      BeginProperty Images {2C247F25-8591-11D1-B16A-
38          00C0F0283628}
39          NumListImages = 4
40          BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-
41              00C0F0283628}
42              Picture      = "frmMain.frx":030A
43              Key          = "new"
44          EndProperty
45          BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-
46              00C0F0283628}
47              Picture      = "frmMain.frx":0954
48              Key          = "opn"
49          EndProperty
50          BeginProperty ListImage3 {2C247F27-8591-11D1-B16A-
51              00C0F0283628}
52              Picture      = "frmMain.frx":0F9E
53              Key          = "prn"
54          EndProperty
55          BeginProperty ListImage4 {2C247F27-8591-11D1-B16A-
56              00C0F0283628}
57              Picture      = "frmMain.frx":3750
58              Key          = "rfr"
59          EndProperty
60      End
61      Begin MSComctlLib.Toolbar Toolbar2
62          Align          = 1 'Oben ausrichten
63          Height        = 330
64          Left          = 0
65          TabIndex      = 8
66          Top           = 0
67          Width         = 10830
68          ExtentX       = 19103
69          ExtentY       = 582
70          ButtonWidth   = 2487
71          ButtonHeight  = 582
72          AllowCustomize = 0 'False
73          Wrappable     = 0 'False
74          Style         = 1
75          TextAlignment = 1
76          Version       = 393216
77      End
78      Begin VB.PictureBox picBohrung
79          BorderStyle   = 0 'Kein
80          Height        = 5535
81          Left          = 2880
82          ScaleHeight   = 5535
83          ScaleWidth    = 4335

```

```

80     TabIndex      = 5
81     TabStop      = 0 'False
82     Top          = 480
83     Width        = 4335
84     Begin Gebirgsklassifikation.ct1Bohrung ctlBohrung1
85         Height   = 4935
86         Left     = 120
87         TabIndex = 7
88         Top      = 0
89         Width   = 3975
90         ExtentX = 7011
91         ExtentY = 8705
92     End
93     Begin VB.Label picBohrungLabel
94         Alignment = 2 'Zentriert
95         BorderStyle = 1 'Fest Einfach
96         Caption = "keine Bohrung gewählt"
97         ForeColor = &H80000011&
98         Height = 255
99         Left = 0
100        TabIndex = 6
101        Top = 5160
102        Width = 3975
103    End
104    End
105    Begin MSComDlg.CommonDialog CommonDialog1
106        Left = 8760
107        Top = 480
108        ExtentX = 847
109        ExtentY = 847
110        Version = 393216
111    End
112    Begin VB.PictureBox Picture6
113        BorderStyle = 0 'Kein
114        Height = 5700
115        Left = 45
116        ScaleHeight = 380
117        ScaleMode = 3 'Pixel
118        ScaleWidth = 174
119        TabIndex = 1
120        TabStop = 0 'False
121        Top = 360
122        Width = 2610
123    Begin MSComctlLib.Toolbar Toolbar1
124        Height = 330
125        Left = 0
126        TabIndex = 3
127        Top = 0

```

```

128        Width = 2265
129        ExtentX = 3995
130        ExtentY = 582
131        ButtonWidth = 609
132        ButtonHeight = 582
133        AllowCustomize = 0 'False
134        Wrappable = 0 'False
135        Style = 1
136        Version = 393216
137    Begin MSComctlLib.ImageList ImageList1
138        Left = 1680
139        Top = 0
140        ExtentX = 1005
141        ExtentY = 1005
142        BackColor = -2147483643
143        ImageWidth = 16
144        ImageHeight = 16
145        MaskColor = 12632256
146        Version = 393216
147        BeginProperty Images {2C247F25-8591-11D1-B16A-
00C0F0283628}
148            NumListImages = 3
149            BeginProperty ListImage1 {2C247F27-8591-11D1-
B16A-00C0F0283628}
150                Picture = "frmMain.frx":38AA
151                Key = "new"
152            EndProperty
153            BeginProperty ListImage2 {2C247F27-8591-11D1-
B16A-00C0F0283628}
154                Picture = "frmMain.frx":3CFC
155                Key = "del"
156            EndProperty
157            BeginProperty ListImage3 {2C247F27-8591-11D1-
B16A-00C0F0283628}
158                Picture = "frmMain.frx":414E
159                Key = "edt"
160            EndProperty
161        EndProperty
162    End
163    End
164    Begin MSComctlLib.ListView ListView2
165        Height = 4755
166        Left = 0
167        TabIndex = 2
168        Top = 360
169        Width = 2280
170        ExtentX = 4022
171        ExtentY = 8387

```

```

172 View = 3
173 Arrange = 1
174 LabelEdit = 1
175 LabelWrap = -1 'True
176 HideSelection = 0 'False
177 FullRowSelect = -1 'True
178 GridLines = -1 'True
179 _Version = 393217
180 ForeColor = -2147483640
181 BackColor = -2147483643
182 BorderStyle = 1
183 Appearance = 1
184 NumItems = 2
185 BeginProperty ColumnHeader(1) {BDD1F052-858B-11D1-B16A-
00C0F0283628}
186 Text = "Bohrung"
187 Object.Width = 2117
188 EndProperty
189 BeginProperty ColumnHeader(2) {BDD1F052-858B-11D1-B16A-
00C0F0283628}
190 Alignment = 1
191 SubItemIndex = 1
192 Text = "Teufe"
193 Object.Width = 1720
194 EndProperty
195 End
196 Begin VB.Label ListView2Label
197 Alignment = 2 'Zentriert
198 BackColor = &H80000005&
199 BorderStyle = 1 'Fest Einfach
200 Caption = "Tabelle gesperrt"
201 ForeColor = &H80000011&
202 Height = 255
203 Left = 0
204 TabIndex = 4
205 Top = 5160
206 Width = 2175
207 End
208 End
209 Begin MSComctlLib.StatusBar StatusBar1
210 Align = 2 'Unten ausrichten
211 Height = 330
212 Left = 0
213 TabIndex = 0
214 Top = 6270
215 Width = 10830
216 ExtentX = 19103
217 _ExtentY = 582

```

```

218 _Version = 393216
219 BeginProperty Panels {8E3867A5-8586-11D1-B16A-
00C0F0283628}
220 NumPanels = 2
221 BeginProperty Panel1 {8E3867AB-8586-11D1-B16A-
00C0F0283628}
222 Alignment = 1
223 AutoSize = 2
224 Object.Width = 1773
225 MinWidth = 1764
226 Text = "0 Bohrungen"
227 TextSave = "0 Bohrungen"
228 Key = "Bohrungen"
229 Object.ToolTipText = "Anzahl der Bohrungen in
der Datenbank"
230 EndProperty
231 BeginProperty Panel2 {8E3867AB-8586-11D1-B16A-
00C0F0283628}
232 AutoSize = 1
233 Bevel = 0
234 Object.Width = 16801
235 Key = "DatabaseName"
236 EndProperty
237 EndProperty
238 End
239 Begin VB.Menu m Datei
240 Caption = "&Datei"
241 Begin VB.Menu md neu
242 Caption = "&neu..."
243 Shortcut = ^N
244 End
245 Begin VB.Menu md open
246 Caption = "&öffnen..."
247 Shortcut = ^O
248 End
249 Begin VB.Menu md_tr1
250 Caption = "- "
251 End
252 Begin VB.Menu md saveas
253 Caption = "speichern &unter"
254 End
255 Begin VB.Menu md_tr2
256 Caption = "- "
257 End
258 Begin VB.Menu md print
259 Caption = "&drucken..."
260 Enabled = 0 'False
261 End

```

```
262     Begin VB.Menu md_export
263         Caption      = "Exportieren"
264     Begin VB.Menu md_exportWMF
265         Caption      = "aktuelle Ansicht in WMF-
Datei..."
266     End
267     Begin VB.Menu md_exportExcel
268         Caption      = "aktuelle Bohrung nach MS Excel"
269     End
270 End
271 Begin VB.Menu md_filetr
272     Caption      = "-"
273     Visible      = 0 'False
274 End
275 Begin VB.Menu md_file
276     Caption      = ""
277     Index        = 1
278     Visible      = 0 'False
279 End
280 Begin VB.Menu md_file
281     Caption      = ""
282     Index        = 2
283     Visible      = 0 'False
284 End
285 Begin VB.Menu md_file
286     Caption      = ""
287     Index        = 3
288     Visible      = 0 'False
289 End
290 Begin VB.Menu md_file
291     Caption      = ""
292     Index        = 4
293     Visible      = 0 'False
294 End
295 Begin VB.Menu md_file
296     Caption      = ""
297     Index        = 5
298     Visible      = 0 'False
299 End
300 Begin VB.Menu md_file
301     Caption      = ""
302     Index        = 6
303     Visible      = 0 'False
304 End
305 Begin VB.Menu md_file
306     Caption      = ""
307     Index        = 7
308     Visible      = 0 'False
```

```
309     End
310     Begin VB.Menu md_file
311         Caption      = ""
312         Index        = 8
313         Visible      = 0 'False
314     End
315     Begin VB.Menu md_file
316         Caption      = ""
317         Index        = 9
318         Visible      = 0 'False
319     End
320     Begin VB.Menu md_tr3
321         Caption      = "-"
322     End
323     Begin VB.Menu md_Exit
324         Caption      = "&Beenden"
325     End
326 End
327 Begin VB.Menu m_edit
328     Caption      = "&Bearbeiten"
329     Begin VB.Menu mb_BohrungAdd
330         Caption      = "Bohrung einfügen"
331     End
332     Begin VB.Menu mb_BohrungEdit
333         Caption      = "Eigenschaften Bohrung..."
334         Shortcut      = {F8}
335     End
336     Begin VB.Menu mb_BohrungDel
337         Caption      = "Bohrung löschen..."
338     End
339     Begin VB.Menu mb_tr1
340         Caption      = "-"
341     End
342     Begin VB.Menu me_KernmarschAdd
343         Caption      = "Kernmarsch einfügen"
344     End
345     Begin VB.Menu me_KernmarschEdt
346         Caption      = "Eigenschaften Kernmarsch..."
347     End
348     Begin VB.Menu me_KernmarschDel
349         Caption      = "Kernmarsch löschen..."
350     End
351     Begin VB.Menu mb_tr2
352         Caption      = "-"
353     End
354     Begin VB.Menu mb_Verweise
355         Caption      = "Verweise: Kernauswerteverfahren"
356         Shortcut      = ^T
```

```

357     End
358 End
359 Begin VB.Menu m Ansicht
360     Caption = "&Ansicht"
361     Begin VB.Menu ma Refresh
362         Caption = "&neu Aufbauen"
363         Shortcut = {F5}
364     End
365 End
366 Begin VB.Menu m Help
367     Caption = "&?"
368     Begin VB.Menu mh über
369         Caption = "&Info zu Gebirgsklassifikation..."
370     End
371 End
372 End
373 Attribute VB Name = "frmMain"
374 Attribute VB GlobalNameSpace = False
375 Attribute VB_Creatable = False
376 Attribute VB_PredeclaredId = True
377 Attribute VB_Exposed = False
378 Option Explicit
379
380 Private Database As DAO.Database
381 Private DatabaseFileName As String
382
383 Private RSBohrungen As DAO.Recordset
384 Private RSKernmärsche As DAO.Recordset, BohrungKey As String
385 Private RSObjekte As DAO.Recordset
386 Private RSBohrverfahren As DAO.Recordset, RSBohrwerkzeug As
DAO.Recordset
387
388 Private Sub Form_Load()
389     'CommonDialog1.InitDir = ...
390
391     With Toolbar1
392         Set .ImageList = ImageList1
393
394         With .Buttons
395             .Add(, "new", , , "new").ToolTipText = "neue Bohrung
erstellen"
396             .Add(, "edt", , , "edt").ToolTipText = "Eigenschaften der
markierten Bohrung anzeigen"
397             .Add , , , tbrSeparator
398             .Add(, "del", , , "del").ToolTipText = "markierte Bohrung
löschen"
399         End With
400     End With

```

```

401
402     With Toolbar2
403         Set .ImageList = ImageList2
404
405         With .Buttons
406             .Add(, "new", "neue Datei ", , "new").ToolTipText = "neue
Datenbank erstellen"
407             .Add(, "opn", "Datei öffnen ", , "opn").ToolTipText =
"bestehende Datenbank öffnen"
408             .Add , , , tbrSeparator
409             With .Add(, "prn", "drucken ... ", , "prn")
410                 .ToolTipText = "Aktuelle Ansicht drucken"
411                 .Enabled = False
412             End With
413             .Add , , , tbrSeparator
414             .Add(, "rfr", "neu aufbauen", , "rfr").ToolTipText =
"Daten erneut von Datenbank abrufen (F5)"
415         End With
416     End With
417
418     OpenFile vbNullString
419 End Sub
420
421 Private Sub Form_Resize()
422     On Error Resume Next
423     Dim WH As Long, L As Long, T As Double
424
425     T = Toolbar2.Height + Picture1.Height + Screen.TwipsPerPixelY
426     WH = ScaleHeight - StatusBar1.Height - T
427
428     Picture6.Move 0, T, Picture6.Width, WH
429
430     L = Picture6.Left + Picture6.Width + 3 * Screen.TwipsPerPixelX
431     picBohrung.Move L, T, ScaleWidth - L, WH
432 End Sub
433
434 Private Sub ListView2_Click()
435     Call ShowBohrung
436 End Sub
437
438 Private Sub ListView2_ColumnClick(ByVal ColumnHeader As
MSComctlLib.ColumnHeader)
439     ListView2.ColumnClick ListView2, ColumnHeader.Index
440 End Sub
441
442 Private Sub ListView2_DblClick()
443     Toolbar1.ButtonClick Toolbar1.Buttons("edt")
444 End Sub

```

```
445
446 Private Sub ListView2_KeyDown(KeyCode As Integer, Shift As
Integer)
447     If KeyCode = vbKeyDelete Then
448         Toolbar1_ButtonClick Toolbar1.Buttons("del")
449     End If
450 End Sub
451
452 Private Sub ListView2_KeyPress(KeyAscii As Integer)
453     If KeyAscii = vbKeyReturn Then
454         ListView2_DblClick
455     End If
456 End Sub
457
458 Private Sub ListView2_KeyUp(KeyCode As Integer, Shift As
Integer)
459     ListView2_Click
460 End Sub
461
462 Private Sub ma_Refresh_Click()
463     ListBohrungen
464 End Sub
465
466 Private Sub mb_BohrungAdd_Click()
467     Toolbar1_ButtonClick Toolbar1.Buttons("new")
468 End Sub
469 Private Sub mb_BohrungDel_Click()
470     Toolbar1_ButtonClick Toolbar1.Buttons("del")
471 End Sub
472 Private Sub mb_BohrungEdit_Click()
473     Toolbar1_ButtonClick Toolbar1.Buttons("edt")
474 End Sub
475
476 Private Sub mb_Verweise_Click()
477     frmVerweise.Show 1, Me
478 End Sub
479
480 Private Sub md_Exit_Click()
481     Unload Me
482 End Sub
483
484 Private Sub md_exportExcel_Click()
485     Dim FirstNewWorksheet As Boolean
486
487     On Error Resume Next
488     If Not (RSKernmärsche Is Nothing Or RSBohrungen Is Nothing)
Then
489
```

```
490     Dim ExcelApp As Object, Workbook As Object, Worksheet As
Object
491     Dim Row As Long, N As Long, RCX As Collection, Obj As
Object, CC As RockClassX.clsClassification
492     Dim Parameter As Scripting.Dictionary
493
494     'Klassifikationen zusammenstellen
495     Set RCX = New Collection
496     For N = 1 To ClassificationCollection.Count
497         Set Obj = Nothing
498         Set Obj = ClassificationCollection.Classification(N)
499         If Not Obj Is Nothing Then
500             RCX.Add Obj
501         End If
502     Next
503
504     FirstNewWorksheet = True
505
506     '1. Excel-Objekt erstellen & vorbereiten
507     Set ExcelApp = CreateObject("Excel.Application")
508     Set Workbook = ExcelApp.Workbooks.Add
509     Set Worksheet = prvGetNewWorksheet(Workbook, "Bohrung " &
ctlBohrung1.BohrungName, FirstNewWorksheet)
510
511     If ExcelApp Is Nothing Then
512         MsgBox "Es konnte keine Verbindung zu Microsoft Excel97
oder höher aufgebaut werden. Bitte installieren Sie das Produkt
erneut.", vbExclamation, "Export der Daten"
513     Else
514         '2. mit Daten füllen
515         With RSKernmärsche.Clone
516             Worksheet.range("A1") = "Bohrung"
517             Worksheet.range("B1") = ctlBohrung1.BohrungName
518             Worksheet.range("A2") = "Teufe"
519             Worksheet.range("B2") = ctlBohrung1.Teufe
520
521             Row = 4
522             Worksheet.range("A" & Row) = "Teufe"
523             Worksheet.range("B" & Row) = "Länge"
524             Worksheet.range("C" & Row) = "Parameter"
525             For N = 1 To RCX.Count
526                 Set CC = RCX(N)
527                 Worksheet.Cells(Row, N + 3).Value =
CC.ClassificationName
528             Next
529
530             .MoveFirst
531             Do Until .EOF
```

```
532         If .Fields("BohrungID") = BohrungKey Then
533             Row = Row + 1
534
535             Worksheet.range("A" & Row) = CMe(.Fields("Teufe") &
vbNullString)
536             Worksheet.range("B" & Row) = CMe(.Fields("Länge") &
vbNullString)
537             Worksheet.range("C" & Row) = .Fields("Parameter") &
vbNullString
538
539             Set Parameter = New Scripting.Dictionary
540             Set Parameter =
RockClassX.String2Dictionary(.Fields("Parameter") &
vbNullString)
541             Parameter.Add "pieces", .Fields("Bruchstücke") &
vbNullString
542             Parameter.Add "kernmarsch", .Fields("Länge") * 100
543
544             For N = 1 To RCX.Count
545                 Set CC = RCX(N)
546                 Err.Clear
547                 With Worksheet.Cells(Row, N + 3)
548                     .Value = CC.Calculate(Parameter)
549                     If Err.Number <> 0 Then
550                         .Value = Err.Description
551                     End If
552                 End With
553             Next
554         End If
555
556         .MoveNext
557     Loop
558 End With
559
560     '3. aktivieren
561     ExcelApp.Visible = True
562     Worksheet.Activate
563 End If
564 End If
565 End Sub
566
567 Private Function prvGetNewWorksheet(Workbook As Object, Name As
String, ByRef FirstNewWorksheet As Boolean) As Object
568     Dim tempWorksheet As Object
569
570     Const xlSheetVeryHidden As Long = 2
571
```

```
572     Set prvGetNewWorksheet = Workbook.Worksheets.Add(,
Workbook.Worksheets(Workbook.Worksheets.Count))
573     With prvGetNewWorksheet
574         .Name = Name
575     End With
576
577     If FirstNewWorksheet Then
578         For Each tempWorksheet In Workbook.Worksheets
579             If Not tempWorksheet Is prvGetNewWorksheet Then
580                 tempWorksheet.Visible = xlSheetVeryHidden
581             End If
582         Next
583         FirstNewWorksheet = False
584     End If
585 End Function
586 Private Sub md_exportWMF Click()
587     ctlBohrung1.Export2WMF
588 End Sub
589
590 Private Sub md_file_Click(Index As Integer)
591     OpenFile md_file(Index).Tag
592 End Sub
593
594 Private Sub md_neu_Click()
595     Dim sFile As String
596
597     On Error Resume Next
598     With CommonDialog1
599         .CancelError = True
600         .Flags = cdLOFNOverwritePrompt Or cdLOFNLongNames Or
cdLOFNNoChangeDir
601         .Filter = "Datenbanken (*.mdb)|*.mdb|alle Dateien (*.*)|*.*"
602         .ShowSave
603
604         If Err = 0 Then
605             'new File
606             sFile = FileCommands.GetFullFilePath(App.Path,
"source.mdb")
607             FileCopy sFile, .FileName
608             OpenFile .FileName
609         End If
610     End With
611 End Sub
612
613 Private Sub md_open_Click()
614     On Error Resume Next
615     With CommonDialog1
616         .CancelError = True
```

```
617 .Flags = cdlOFNFileMustExist Or cdlOFNLongNames Or
cdlOFNNoChangeDir
618 .Filter = "Datenbanken (*.mdb)|*.mdb|alle Dateien (*.*)|*.*"
619
620 .ShowOpen
621
622 If Err = 0 Then
623     'open File
624     OpenFile .FileName
625 End If
626 End With
627 End Sub
628
629 Private Sub md_saveas_Click()
630     If Not Database Is Nothing Then
631         On Error Resume Next
632         With CommonDialog1
633             .CancelError = True
634             .Flags = cdlOFNOverwritePrompt Or cdlOFNLongNames Or
cdlOFNNoChangeDir
635             .Filter = "Datenbanken (*.mdb)|*.mdb|alle Dateien
(*.*)|*.*"
636             .ShowSave
637
638             If Err = 0 Then
639                 'copy File
640                 Database.Close
641                 FileCopy DatabaseFileName, .FileName
642                 OpenFile .FileName
643             End If
644         End With
645     End If
646 End Sub
647
648 Private Sub me KernmarschAdd_Click()
649     On Error Resume Next
650     ctlBohrung1.KernmarschAdd
651 End Sub
652
653 Private Sub me KernmarschDel_Click()
654     On Error Resume Next
655     ctlBohrung1.KernmarschDel
656 End Sub
657
658 Private Sub me KernmarschEdt_Click()
659     On Error Resume Next
660     ctlBohrung1.KernmarschEdit
661 End Sub
```

```
662
663 Private Sub mh_über_Click()
664     frmInfo.Show 1, Me
665 End Sub
666
667 Private Sub picBohrung_Resize()
668     ctlBohrung1.Move 0, 0, picBohrung.ScaleWidth,
picBohrung.ScaleHeight
669     With ctlBohrung1
670         picBohrungLabel.Move .Left, .Top, .Width, .Height
671     End With
672 End Sub
673
674 Private Sub Picture6_Resize()
675     On Error Resume Next
676     With ListView2
677         ListView2.Move 0, ListView2.Top, Picture6.ScaleWidth,
Picture6.ScaleHeight - ListView2.Top
678         ListView2Label.Move .Left, .Top, .Width, .Height
679     End With
680 End Sub
681
682 Private Sub OpenFile(FileName As String)
683     DatabaseFileName = FileName
684
685     On Error Resume Next
686
687     Dim DBE As DAO.DBEngine
688     Set DBE = New DAO.DBEngine
689     Set Database = Nothing
690     If FileName <> vbNullString Then
691         Set Database = DBE.OpenDatabase(DatabaseFileName)
692     End If
693
694     If Err <> 0 Then
695         MsgBox Err.Description & " (Fehlercode: " & Err.Number &
")", vbExclamation, "Fehler"
696     End If
697
698     If Database Is Nothing Then
699         FileName = vbNullString
700     End If
701
702     SetCaption
703
704     ListBohrungen
705
706     RecentAdd FileName, md_file, md_filetr
```

```

707 End Sub
708
709 Private Sub SetCaption()
710     Dim T As String
711
712     If Not Database Is Nothing Then
713         T = ExtractFileName(DatabaseFileName, False) & " - "
714     End If
715     Caption = T & "Gebirgsklassifikation"
716
717     md saveas.Enabled = Not Database Is Nothing
718 End Sub
719
720 Private Sub ListBohrungen(Optional SelectItem As String)
721     Dim RS As DAO.Recordset
722     Dim s As String
723
724     Dim OldBKey As String
725     OldBKey = vbNullString
726     If Not ListView2.SelectedItem Is Nothing Then
727         OldBKey = ListView2.SelectedItem.Key
728     End If
729     If SelectItem <> vbNullString Then
730         OldBKey = SelectItem
731     End If
732
733     On Error Resume Next
734
735     Set RSBohrungen = Nothing
736     If Database Is Nothing Then
737         ListView2.Visible = False
738         ListView2.Label = "keine Datenbank geöffnet" & vbCrLf &
vbCrLf & "Bitte öffnen Sie eine Datenbank oder legen Sie eine
neue Datenbank an (Menü Datei)."
739         StatusBar1.Panels("Bohrungen") = "keine Datenbank geöffnet"
740         StatusBar1.Panels("DatabaseName") = vbNullString
741     Else
742         Set RS = Database.TableDefs("Bohrungen").OpenRecordset
743         Set RSBohrungen = RS
744         Set RSObjekte = Database.TableDefs("Objekte").OpenRecordset
745         Set RSBohrverfahren =
Database.TableDefs("Bohrverfahren").OpenRecordset
746         Set RSBohrwerkzeug =
Database.TableDefs("Bohrwerkzeug").OpenRecordset
747
748         ListView2.ListItems.Clear
749         If Err <> 0 Then

```

```

750         MsgBox Err.Description & " (Fehlercode: " & Err.Number &
")", vbExclamation, "Fehler"
751         ListView2.Visible = False
752         ListView2.Label = "Tabelle gesperrt"
753         StatusBar1.Panels("Bohrungen") = "Bohrungen gesperrt"
754         StatusBar1.Panels("DatabaseName") = vbNullString
755     Else
756         RS.MoveFirst
757         Do Until RS.EOF
758             With ListView2.ListItems.Add(,
RS.Fields("BohrungID").Value, RS.Fields("Name").Value)
759                 s = Format(RS.Fields("Teufe").Value, "0.00")
760                 s = String(7 - Len(s), " ") & s
761                 .SubItems(1) = s & " m"
762
763                 If OldBKey = .Key Then
764                     .Selected = True
765                 End If
766             End With
767             RS.MoveNext
768         Loop
769         ListView2.Visible = True
770     End If
771     StatusBar1.Panels("Bohrungen") = RS.RecordCount & "
Bohrungen"
772     StatusBar1.Panels("DatabaseName") = Database.Name
773     End If
774
775     Call ShowBohrung
776 End Sub
777
778 Public Sub ShowBohrung()
779     Set RSKernmärsche = Nothing
780     If Not RSBohrungen Is Nothing Then
781         If Not ListView2.SelectedItem Is Nothing Then
782             BohrungKey = ListView2.SelectedItem.Key
783             Set RSKernmärsche =
Database.TableDefs("Kernmärsche").OpenRecordset
784         End If
785     End If
786
787     ctlBohrung1.Visible = Not RSKernmärsche Is Nothing
788     If Not RSKernmärsche Is Nothing Then
789         ctlBohrung1.SetBohrung RSBohrungen, RSKernmärsche,
RSObjekte, BohrungKey, DatabaseFileName
790     End If
791 End Sub

```

```
792
793 Public Sub RecentAdd(ByVal FileName As String, RecentMenu As
Object, RecentMenuTr As Menu)
794
795     Dim C As Collection, a As Long, s As String
796     Set C = New Collection
797
798     C.Add FileName
799     For a = RecentMenu.LBound To RecentMenu.UBound
800         s = GetSetting(App.ProductName, "Recent", "File" & a,
vbNullString)
801         If s <> vbNullString And LCase(Trim(s)) <>
LCase(Trim(FileName)) Then
802             C.Add s
803         End If
804     Next
805
806     For a = RecentMenu.LBound To RecentMenu.UBound
807         If a > C.Count Then
808             s = vbNullString
809         Else
810             s = C(a)
811         End If
812         SaveSetting App.ProductName, "Recent", "File" & a, s
813     Next
814
815     RecentShow RecentMenu, RecentMenuTr
816 End Sub
817
818
819 Public Sub RecentShow(RecentMenu As Object, RecentMenuTr As
Menu)
820     Dim C As Collection, a As Long, s As String, B As Boolean
821     Set C = New Collection
822
823     For a = 1 To 10
824         s = GetSetting(App.ProductName, "Recent", "File" & a,
vbNullString)
825         If s <> vbNullString Then
826             C.Add s
827         End If
828     Next
829
830     B = False
831     For a = RecentMenu.LBound To RecentMenu.UBound
832         If a > C.Count Then
833             s = vbNullString
```

```
834     Else
835         s = C(a)
836     End If
837     With RecentMenu.Item(a)
838         .Caption = "&" & a & ": " & ExtractFileName(s, True)
839         .Tag = s
840         .Visible = s <> vbNullString
841         B = B Or .Visible
842     End With
843 Next
844 If Not RecentMenuTr Is Nothing Then
845     RecentMenuTr.Visible = B
846 End If
847 End Sub
848
849 Private Sub Toolbar1 ButtonClick(ByVal Button As
MSComctlLib.Button)
850     Dim KKey As String, frm As frmEditBohrung
851     Dim ShowAssistent As Boolean
852
853     On Error Resume Next
854
855     If Not RSBohrungen Is Nothing Then
856         Select Case Button.Key
857             Case "new"
858                 Dim NewName As String, NewNameOk As Boolean,
NewNameIndex As Long
859                 With RSBohrungen.Clone
860                     NewNameIndex = 0
861                     Do
862                         NewNameIndex = NewNameIndex + 1
863                         NewName = "Bohrung " & NewNameIndex
864                         NewNameOk = True
865                         .MoveFirst
866                         Do Until .EOF
867                             If .Fields("Name") = NewName Then
868                                 NewNameOk = False
869                                 Exit Do
870                             End If
871                             .MoveNext
872                         Loop
873                         Loop Until NewNameOk
874
875                     End With
876                 With RSBohrungen.Clone
877                     .AddNew
878                     .Fields("Name") = NewName
879
```

```

880         KKey = .Fields("BohrungID")
881
882         .Update
883
884         ListBohrungen KKey
885     End With
886     Toolbar1_ButtonClick Toolbar1.Buttons("edt")
887
888
889     Case "del"
890         If Not ListView2.SelectedItem Is Nothing Then
891             KKey = ListView2.SelectedItem.Key
892             With RSBohrungen
893                 .MoveFirst
894                 Do Until .EOF
895                     If .Fields("BohrungID") = KKey Then
896                         If MsgBox("Soll die Bohrung "" &
.Fields("Name") & "" gelöcht werden?", vbQuestion Or vbYesNo Or
vbDefaultButton2, "Bohrung löschen") = vbYes Then
897                             .Delete
898
899                             If Err <> 0 Then
900                                 MsgBox "Bei dem Vorgang trat ein Fehler
auf:" & vbCrLf & Err.Description & " (Nummer: " & Err.Number &
")", vbExclamation, "Bohrung löschen"
901                             End If
902
903                             Call ma Refresh Click
904                         End If
905                     Exit Do
906                 End If
907                 .MoveNext
908             Loop
909         End With
910     End If
911
912     Case "edt"
913         If Not ListView2.SelectedItem Is Nothing Then
914             KKey = ListView2.SelectedItem.Key
915
916             Set frm = frmEditBohrung
917             frm.Hide
918
919             With RSBohrungen.Clone
920                 .MoveFirst
921                 Do Until .EOF

```

```

923         If .Fields("BohrungID") = KKey Then
924
925             With RSBohrverfahren.Clone
926                 .MoveFirst
927                 frm.Combol.Clear
928                 Do Until .EOF
929                     frm.Combol.AddItem .Fields("Bohrverfahren")
930                 .MoveNext
931             Loop
932         End With
933
934         With RSBohrwerkzeug.Clone
935             .MoveFirst
936             frm.Combo2.Clear
937             Do Until .EOF
938                 frm.Combo2.AddItem .Fields("Bohrwerkzeug")
939             .MoveNext
940         Loop
941     End With
942
943     .Edit
944     PutTextBox .Fields("Name"), frm.Text3
945     PutTextBox .Fields("Teufe"), frm.Text5
946     PutTextBox .Fields("Durchmesser"), frm.Text6
947     PutTextBox .Fields("KoordinatenX"), frm.Text1
948     PutTextBox .Fields("KoordinatenY"), frm.Text4
949     PutTextBox .Fields("KoordinatenZ"), frm.Text7
950     PutTextBox .Fields("Bohrverfahren"), frm.Combol
951     PutTextBox .Fields("Bohrwerkzeug"), frm.Combo2
952     PutTextBox .Fields("Bemerkungen"), frm.Text2
953
954     frm.Show 1, Me
955
956     If frmEditBohrung.LastExitOK Then
957         'Werte in die Datenbank übernehmen
958         GetTextBox .Fields("Name"), frm.Text3
959         GetTextBox .Fields("Teufe"), frm.Text5
960         GetTextBox .Fields("Durchmesser"), frm.Text6
961         GetTextBox .Fields("KoordinatenX"), frm.Text1
962         GetTextBox .Fields("KoordinatenY"), frm.Text4
963         GetTextBox .Fields("KoordinatenZ"), frm.Text7
964         GetTextBox .Fields("Bohrverfahren"),
frm.Combol
965         GetTextBox .Fields("Bohrwerkzeug"), frm.Combo2
966         GetTextBox .Fields("Bemerkungen"), frm.Text2
967         .Update
968
969         Call ma Refresh Click

```

```

970
971         'Assistent anzeigen?
972         With RSKernmärsche.Clone
973             ShowAssistent = True
974             .MoveFirst
975             Do Until .EOF
976                 If .Fields("BohrungID") = KKey Then
977                     ShowAssistent = False
978                     Exit Do
979                 End If
980                 .MoveNext
981             Loop
982
983             If ShowAssistent And Val(frm.Text5) <> 0
984
985             Then
986                 frmAssistentKernmärsche.BohrungKey = KKey
987                 Set frmAssistentKernmärsche.RSBohrungen =
988                 RSBohrungen.Clone
989                 Set frmAssistentKernmärsche.RSKernmärsche
990                 = RSKernmärsche.Clone
991                 frmAssistentKernmärsche.Show 1, Me
992                 Call ma Refresh Click
993                 End If
994             End With
995             End If
996             Exit Do
997             End If
998             .MoveNext
999             Loop
1000             End With
1001         End Select
1002     End If
1003 End Sub
1004 Private Sub Toolbar2 ButtonClick(ByVal Button As
1005 MSComctlLib.Button)
1006     Select Case Button.Key
1007         Case "new": Call md neu Click
1008         Case "opn": Call md_open_Click
1009         Case "rfr": Call ma_Refresh_Click
1010         'Case "prn": Call md print Click
1011     End Select
1012 End Sub

```

Datei: frmIntro.frm (VB Form)

```

01 VERSION 5.00
02 Begin VB.Form frmIntro
03     BackColor        = &H00FFFFFF&
04     BorderStyle      = 0 'Kein
05     Caption          = "Form1"
06     ClientHeight     = 2760
07     ClientLeft       = 0
08     ClientTop        = 0
09     ClientWidth      = 5205
10     LinkTopic        = "Form1"
11     ScaleHeight      = 184
12     ScaleMode        = 3 'Pixel
13     ScaleWidth       = 347
14     ShowInTaskbar    = 0 'False
15     StartupPosition  = 2 'Bildschirmmitte
16     Begin VB.Label Label4
17         Alignment     = 1 'Rechts
18         BackStyle     = 0 'Transparent
19         Caption       = """"Vergleichende Untersuchungen zur
20         Repräsentanz des Lm-Wertes""""
21     EndProperty
22     BeginProperty Font
23         Name          = "Tahoma"
24         Size          = 9.75
25         Charset       = 0
26         Weight        = 700
27         Underline     = 0 'False
28         Italic        = 0 'False
29         Strikethrough = 0 'False
30     EndProperty
31     Height           = 585
32     Left             = 315
33     TabIndex         = 4
34     Top              = 1590
35     Width            = 4605
36 End
37 Begin VB.Label Label2
38     Alignment     = 1 'Rechts
39     BackStyle     = 0 'Transparent
40     Caption       = "Studienarbeit zum Thema:"
41     BeginProperty Font
42         Name          = "Tahoma"
43         Size          = 9.75
44         Charset       = 0
45         Weight        = 400
46         Underline     = 0 'False

```

```
45         Italic           = 0 'False
46         Strikethrough    = 0 'False
47     EndProperty
48     Height               = 345
49     Left                 = 1815
50     TabIndex            = 3
51     Top                  = 1260
52     Width                = 3105
53 End
54 Begin VB.Label Label3
55     Alignment            = 1 'Rechts
56     BackStyle            = 0 'Transparent
57     Caption              = "2001, Jörg Meier"
58     BeginProperty Font
59         Name              = "Tahoma"
60         Size              = 9.75
61         Charset           = 0
62         Weight            = 700
63         Underline         = 0 'False
64         Italic            = 0 'False
65         Strikethrough     = 0 'False
66     EndProperty
67     Height               = 345
68     Left                 = 1815
69     TabIndex            = 2
70     Top                  = 2160
71     Width                = 3105
72 End
73 Begin VB.Label VersionLabel
74     Alignment            = 1 'Rechts
75     BackStyle            = 0 'Transparent
76     Caption              = "Version"
77     Height               = 270
78     Left                 = 1995
79     TabIndex            = 1
80     Top                  = 720
81     Width                = 2925
82 End
83 Begin VB.Label Label1
84     Alignment            = 1 'Rechts
85     BackStyle            = 0 'Transparent
86     Caption              = "Gebirgsklassifikation"
87     BeginProperty Font
88         Name              = "Tahoma"
89         Size              = 20.25
90         Charset           = 0
91         Weight            = 700
```

```
92         Underline        = 0 'False
93         Italic            = 0 'False
94         Strikethrough     = 0 'False
95     EndProperty
96     Height               = 750
97     Left                 = 165
98     TabIndex            = 0
99     Top                  = 255
100    Width                = 4755
101 End
102 Begin VB.Shape Shape2
103     BorderStyle          = 0 'Transparent
104     DrawMode              = 12 'Keine Operation
105     FillColor             = &H00808080&
106     FillStyle             = 0 'Ausgefüllt
107     Height                = 2130
108     Left                  = 105
109     Top                   = 585
110     Width                 = 1485
111 End
112 Begin VB.Image Image1
113     Height                = 2100
114     Left                  = -180
115     Picture                = "frmIntro.frx":0000
116     Stretch                = -1 'True
117     Top                   = 585
118     Width                 = 2100
119 End
120 Begin VB.Shape Shape1
121     BorderWidth           = 3
122     Height                = 2730
123     Left                  = 15
124     Top                   = 15
125     Width                 = 5175
126 End
127 End
128 Attribute VB_Name = "frmIntro"
129 Attribute VB_GlobalNameSpace = False
130 Attribute VB_Creatable = False
131 Attribute VB_PredeclaredId = True
132 Attribute VB_Exposed = False
133 Option Explicit
134
135 Private Sub Form_Load()
136     VersionLabel = "Version " & App.Major & "." & App.Minor & "."
137     & App.Revision
138 End Sub
```

Datei: frmInfo.frm (VB Form)

```

01 VERSION 5.00
02 Begin VB.Form frmInfo
03     BorderStyle       = 3 'Fester Dialog
04     Caption           = "Info zu Gebirgsklassifikation"
05     ClientHeight      = 3225
06     ClientLeft       = 45
07     ClientTop        = 330
08     ClientWidth      = 5295
09     ControlBox       = 0 'False
10     LinkTopic        = "Form1"
11     LockControls     = -1 'True
12     MaxButton        = 0 'False
13     MinButton        = 0 'False
14     ScaleHeight      = 215
15     ScaleMode        = 3 'Pixel
16     ScaleWidth       = 353
17     ShowInTaskbar   = 0 'False
18     StartUpPosition = 1 'Fenstermitte
19     Begin VB.CommandButton Command1
20         Cancel       = -1 'True
21         Caption     = "&Ok"
22         Default     = -1 'True
23         Height      = 345
24         Left        = 3495
25         TabIndex   = 5
26         Top        = 2700
27         Width      = 1590
28     End
29     Begin VB.Image Image1
30         Height      = 960
31         Left        = 15
32         Picture     = "frmInfo.frx":0000
33         Stretch     = -1 'True
34         Top        = 135
35         Width      = 960
36     End
37     Begin VB.Label Label1
38         Alignment   = 1 'Rechts
39         BackStyle   = 0 'Transparent
40         Caption     = "Gebirgsklassifikation"
41         BeginProperty Font
42             Name      = "Tahoma"
43             Size     = 20.25

```

```

44         Charset     = 0
45         Weight      = 700
46         Underline   = 0 'False
47         Italic      = 0 'False
48         Strikethrough = 0 'False
49     EndProperty
50     Height        = 750
51     Left         = 345
52     TabIndex     = 4
53     Top          = 240
54     Width        = 4755
55 End
56 Begin VB.Label VersionLabel
57     Alignment     = 1 'Rechts
58     BackStyle     = 0 'Transparent
59     Caption       = "Version"
60     Height        = 270
61     Left         = 2175
62     TabIndex     = 3
63     Top          = 705
64     Width        = 2925
65 End
66 Begin VB.Label Label3
67     Alignment     = 1 'Rechts
68     BackStyle     = 0 'Transparent
69     Caption       = "2001, Jörg Meier"
70     BeginProperty Font
71         Name      = "Tahoma"
72         Size     = 9.75
73         Charset  = 0
74         Weight   = 700
75         Underline = 0 'False
76         Italic   = 0 'False
77         Strikethrough = 0 'False
78     EndProperty
79     Height        = 345
80     Left         = 1995
81     TabIndex     = 2
82     Top          = 2145
83     Width        = 3105
84 End
85 Begin VB.Label Label2
86     Alignment     = 1 'Rechts
87     BackStyle     = 0 'Transparent
88     Caption       = "Studienarbeit zum Thema:"
89     BeginProperty Font
90         Name      = "Tahoma"
91         Size     = 9.75

```

```

92         Charset           = 0
93         Weight           = 400
94         Underline        = 0 'False
95         Italic           = 0 'False
96         Strikethrough     = 0 'False
97     EndProperty
98     Height               = 345
99     Left                 = 1995
100    TabIndex             = 1
101    Top                   = 1245
102    Width                 = 3105
103 End
104 Begin VB.Label Label4
105     Alignment            = 1 'Rechts
106     BackStyle            = 0 'Transparent
107     Caption               = ""Vergleichende Untersuchungen zur
108     Repräsentanz des Lm-Wertes""
109     BeginProperty Font
110         Name               = "Tahoma"
111         Size                = 9.75
112         Charset             = 0
113         Weight              = 700
114         Underline          = 0 'False
115         Italic              = 0 'False
116         Strikethrough       = 0 'False
117     EndProperty
118     Height                = 585
119     Left                  = 495
120     TabIndex              = 0
121     Top                   = 1575
122     Width                 = 4605
123 End
124 Attribute VB Name = "frmInfo"
125 Attribute VB GlobalNameSpace = False
126 Attribute VB_Creatable = False
127 Attribute VB_PredeclaredId = True
128 Attribute VB_Exposed = False
129 Option Explicit
130
131 Private Sub Command1_Click()
132     Unload Me
133 End Sub
134
135 Private Sub Form_Load()
136     VersionLabel = "Version " & App.Major & "." & App.Minor & "."
137     & App.Revision

```

```

137 End Sub
138
139

```

Datei: frmVerweise.frm (VB Form)

```

01 VERSION 5.00
02 Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0";
03 "MSCOMCTL.OCX"
04 Object = "{5E9E78A0-531B-11CF-91F6-C2863C385E30}#1.0#0";
05 "MSFLXGRD.OCX"
06 Begin VB.Form frmVerweise
07     BorderStyle          = 3 'Fester Dialog
08     Caption               = "Verweise: Kernauswerteverfahren"
09     ClientHeight          = 4650
10     ClientLeft            = 45
11     ClientTop             = 330
12     ClientWidth           = 6825
13     ControlBox            = 0 'False
14     LinkTopic             = "Form1"
15     MaxButton             = 0 'False
16     MinButton             = 0 'False
17     ScaleHeight           = 4650
18     ScaleWidth            = 6825
19     ShowInTaskbar         = 0 'False
20     StartUpPosition       = 1 'Fenstermitte
21     Begin VB.Frame Frame1
22         Caption           = "Verbindung prüfen"
23         Height            = 1215
24         Left              = 960
25         TabIndex          = 7
26         Top               = 1680
27         Visible           = 0 'False
28         Width             = 4335
29         Begin VB.Label Label2
30             Caption       = "Verbindung wird geprüft..."
31             Height        = 255
32             Left          = 960
33             TabIndex      = 8
34             Top           = 480
35             Width         = 2775
36         End
37         Begin VB.Image Image1
38             Height        = 480
39             Left          = 240
40             Picture       = "frmVerweise.frx":0000

```

```

39         Top           = 360
40         Width        = 480
41     End
42 End
43 Begin VB.TextBox Text1
44     BorderStyle      = 0 'Kein
45     Height           = 195
46     Left             = 1320
47     TabIndex        = 6
48     Text             = "Text1"
49     Top              = 1440
50     Width            = 1575
51 End
52 Begin VB.CommandButton Command2
53     Caption          = "zurücksetzen"
54     Height           = 375
55     Left             = 5280
56     TabIndex        = 5
57     Top              = 3840
58     Width            = 1455
59 End
60 Begin VB.CommandButton Command1
61     Caption          = "&Abbrechen"
62     Height           = 375
63     Index            = 1
64     Left             = 5280
65     TabIndex        = 4
66     Top              = 840
67     Width            = 1455
68 End
69 Begin VB.CommandButton Command1
70     Caption          = "&Ok"
71     Height           = 375
72     Index            = 0
73     Left             = 5280
74     TabIndex        = 3
75     Top              = 360
76     Width            = 1455
77 End
78 Begin MSFlexGridLib.MSFlexGrid MSFlexGrid1
79     Height           = 3855
80     Left             = 120
81     TabIndex        = 2
82     Top              = 360
83     Width            = 5055
84     _ExtentX        = 8916
85     _ExtentY        = 6800

```

```

86     _Version        = 393216
87     Cols            = 3
88     FixedCols       = 0
89     FocusRect       = 0
90 End
91 Begin MSComctlLib.ImageList ImageList1
92     Left            = 5280
93     Top             = 4080
94     ExtentX         = 1005
95     ExtentY         = 1005
96     BackColor       = -2147483643
97     ImageWidth      = 16
98     ImageHeight     = 16
99     MaskColor       = 12632256
100    Version          = 393216
101    BeginProperty Images {2C247F25-8591-11D1-B16A-
00C0F0283628}
102        NumListImages = 2
103    BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-
00C0F0283628}
104        Picture        = "frmVerweise.frx":1272
105        Key            = "del"
106    EndProperty
107    BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-
00C0F0283628}
108        Picture        = "frmVerweise.frx":16C4
109        Key            = "chk"
110    EndProperty
111    EndProperty
112 End
113 Begin MSComctlLib.Toolbar Toolbar1
114     Height           = 330
115     Left             = 120
116     TabIndex        = 1
117     Top              = 4200
118     Width            = 5040
119     _ExtentX        = 8890
120     ExtentY         = 582
121     ButtonWidth     = 609
122     ButtonHeight    = 582
123     AllowCustomize  = 0 'False
124     Wrappable       = 0 'False
125     Style           = 1
126     TextAlignment  = 1
127     Version          = 393216
128 End
129 Begin VB.Label Label1
130     Caption          = "eingebundene Kernauswerteverfahren:"

```

```
131     Height      = 255
132     Left        = 120
133     TabIndex    = 0
134     Top         = 120
135     Width       = 2895
136 End
137 End
138 Attribute VB_Name = "frmVerweise"
139 Attribute VB_GlobalNameSpace = False
140 Attribute VB_Creatable = False
141 Attribute VB_PredeclaredId = True
142 Attribute VB_Exposed = False
143 Option Explicit
144
145 Dim WithEvents EditGrid1 As clsEditGrid
146 Attribute EditGrid1.VB_VarHelpID = -1
147 Dim LastEditRow As Long
148
149 Private Sub Command1_Click(Index As Integer)
150     If Index = 0 Then
151         Dim N As Long
152         With ClassificationCollection
153             .ChancelChange = True
154             .Clear
155             For N = 1 To MSFlexGrid1.Rows - 1
156                 .Add MSFlexGrid1.TextMatrix(N, 0),
MSFlexGrid1.TextMatrix(N, 1)
157             Next
158             .ChancelChange = False
159             .RaiseChange
160
161             .Save App.ProductName, "Classifications"
162         End With
163     End If
164
165     Unload Me
166 End Sub
167
168 Private Sub Command2_Click()
169     Dim TC As clsClassCol
170     Set TC = New clsClassCol
171     TC.Reset
172
173     FillGrid TC
174 End Sub
175
```

```
176 Private Sub EditGrid1_AfterCellEdit(ByVal Row As Long, ByVal Col
As Long, NewValue As String, Cancel As Boolean)
177     LastEditRow = Row
178 End Sub
179
180 Private Sub EditGrid1_BeforeCellEdit(ByVal Row As Long, ByVal
Col As Long, Prompt As String, Cancel As Boolean)
181     Cancel = Col > 1
182 End Sub
183
184 Private Sub EditGrid1_ProcessReady()
185     Call CheckRow(LastEditRow)
186 End Sub
187
188 Private Sub Form_Load()
189     With Toolbar1
190         Set .ImageList = ImageList1
191
192         With .Buttons
193             .Add(, "del", "Verweis löschen", , "del").ToolTipText =
"markiertes Kernauswerteverfahren löschen"
194             .Add(, "chk", "Verweise prüfen", , "chk").ToolTipText =
"prüft alle eingegebenen Verweise"
195         End With
196     End With
197
198     Set EditGrid1 = New clsEditGrid
199     With EditGrid1
200         Set .MSFlexGrid = MSFlexGrid1
201         Set .InputObject = Text1
202         .GridIgnoreClipboardKeys = True
203         .GridAutoRowAdd = True
204
205         .GridAutoFocus ColStart = 0
206         .GridAutoFocus ColEnd = 0
207     End With
208
209     With MSFlexGrid1
210         .TextMatrix(0, 0) = "Verweis"
211         .TextMatrix(0, 1) = "Server"
212         .TextMatrix(0, 2) = "Verbindung"
213
214         .ColWidth(0) = 2500
215         .ColWidth(1) = 1000
216         .ColWidth(2) = 6000
217     End With
218
219     FillGrid ClassificationCollection

```

```

220 End Sub
221
222 Private Sub Form Unload(Cancel As Integer)
223     Set EditGrid1 = Nothing
224 End Sub
225
226 Private Sub CheckRow(Row As Long)
227     Dim T As String, s As String, Obj As Object, clsClassification
228     As RockClassX.clsClassification
229
230     On Error GoTo ErrHandler
231     Err.Clear
232
233     Frame1.Visible = True
234     Refresh
235
236     T = MSFlexGrid1.TextMatrix(Row, 0)
237     s = MSFlexGrid1.TextMatrix(Row, 1)
238     If Not (s = vbNullString And T = vbNullString) Then
239         If s <> vbNullString Then
240             Set Obj = CreateObject(T, s)
241         Else
242             Set Obj = CreateObject(T)
243         End If
244         Set clsClassification = Obj
245         MSFlexGrid1.TextMatrix(Row, 2) = "Ok (Typ: " & TypeName(Obj)
246         & ")"
247     Else
248         MSFlexGrid1.TextMatrix(Row, 2) = ""
249     End If
250
251     Frame1.Visible = False
252 End Sub
253
254 ErrHandler:
255     If Err <> 0 Then
256         MSFlexGrid1.TextMatrix(Row, 2) = Err.Description & " (" &
257         Err.Number & ")"
258     End If
259     Frame1.Visible = False
260 End Sub
261
262 Private Sub Toolbar1_ButtonClick(ByVal Button As
263 MSComctlLib.Button)
264     Select Case Button.Key
265     Case "chk"
266         Dim N As Long

```

```

263     For N = 1 To MSFlexGrid1.Rows - 1
264         Call CheckRow(N)
265     Next
266
267     Case "del"
268         If MsgBox("Soll die markierte Zeile gelöscht werden?",
269 vbYesNo Or vbQuestion, "Zeile löschen") = vbYes Then
270             EditGrid1.RowsDel MSFlexGrid1.Row, MSFlexGrid1.Row
271         End If
272     End Select
273 End Sub
274
275 Private Sub AddGridLine(Name As String, Optional Server As
276 String)
277     With MSFlexGrid1
278         EditGrid1.RowsAdd .Rows - 1
279
280         Dim R As Long
281         R = .Rows - 2
282
283         .TextMatrix(R, 0) = Name
284         .TextMatrix(R, 1) = Server
285     End With
286
287     CheckRow R
288 End Sub
289
290 Private Sub FillGrid(ClassificationCollection As clsClassCol)
291     Dim N As Long
292     With MSFlexGrid1
293         .Rows = .FixedRows
294         .Rows = .FixedRows + 1
295     End With
296     With ClassificationCollection
297         For N = 1 To .Count
298             AddGridLine .ClassificationClass(N),
299             .ClassificationServer(N)
300         Next
301     End With
302 End Sub

```

Datei: frmEditBohrung.frm (VB Form)

```

01 VERSION 5.00
02 Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0";
"MSCOMCTL.OCX"

```

```

03 Object = "{86CF1D34-0C5F-11D2-A9FC-0000F8754DA1}#2.0#0";
   "MSCOMCT2.OCX"
04 Object = "{BDC217C8-ED16-11CD-956C-0000C04E4C0A}#1.1#0";
   "TABCTL32.OCX"
05 Begin VB.Form frmEditBohrung
06     Caption           = "Bohrung bearbeiten"
07     ClientHeight      = 5460
08     ClientLeft       = 60
09     ClientTop        = 345
10     ClientWidth      = 7275
11     ControlBox       = 0 'False
12     LinkTopic        = "Form1"
13     LockControls     = -1 'True
14     ScaleHeight      = 5460
15     ScaleWidth       = 7275
16     StartUpPosition = 1 'Fenstermitte
17     Begin VB.Frame Frame1
18         BorderStyle   = 0 'Kein
19         Caption       = "Frame1"
20         Height        = 375
21         Left          = 3840
22         TabIndex     = 31
23         Top           = 4920
24         Width         = 3015
25         Begin VB.CommandButton Command1
26             Caption   = "&Ok"
27             Default   = -1 'True
28             Height    = 375
29             Index     = 1
30             Left      = 0
31             TabIndex  = 22
32             Top       = 0
33             Width     = 1455
34         End
25         Begin VB.CommandButton Command1
36             Cancel    = -1 'True
37             Caption   = "&Abbrechen"
38             Height    = 375
39             Index     = 0
40             Left      = 1560
41             TabIndex  = 23
42             Top       = 0
43             Width     = 1455
44         End
45     End
46     Begin TabDlg.SSTab SSTab1
47         Height       = 4695

```

```

48     Left           = 120
49     TabIndex      = 24
50     Top           = 120
51     Width         = 6735
52     ExtentX       = 11880
53     _ExtentY     = 8281
54     _Version      = 393216
55     Style         = 1
56     Tabs          = 2
57     TabHeight     = 520
58     TabCaption(0) = "Allgemeine &Daten"
59     TabPicture(0) = "frmEditBohrung.frx":0000
60     Tab(0).ControlEnabled= -1 'True
61     Tab(0).Control(0)= "Picture1"
62     Tab(0).Control(0).Enabled= 0 'False
63     Tab(0).ControlCount= 1
64     TabCaption(1) = "&Bemerkungen"
65     TabPicture(1) = "frmEditBohrung.frx":001C
66     Tab(1).ControlEnabled= 0 'False
67     Tab(1).Control(0)= "Text2"
68     Tab(1).ControlCount= 1
69     Begin VB.PictureBox Picture1
70         BorderStyle   = 0 'Kein
71         Height        = 4095
72         Left          = 120
73         ScaleHeight   = 4095
74         ScaleWidth    = 6495
75         TabIndex     = 26
76         TabStop      = 0 'False
77         Top           = 480
78         Width         = 6495
79         Begin VB.ComboBox Combo2
80             Height    = 315
81             Left      = 1800
82             TabIndex  = 21
83             Text      = "Combo1"
84             Top       = 3720
85             Width     = 4695
86         End
87         Begin VB.ComboBox Combo1
88             Height    = 315
89             Left      = 1800
90             TabIndex  = 19
91             Text      = "Combo1"
92             Top       = 3360
93             Width     = 4695
94         End
95     Begin MSComCtl12.UpDown UpDown2

```

```
96      Height      = 285
97      Left        = 3735
98      TabIndex    = 11
99      TabStop     = 0 'False
100     Top         = 2520
101     Width       = 240
102     _ExtentX    = 423
103     _ExtentY    = 503
104     _Version     = 393216
105     BuddyControl = "Text1"
106     BuddyDispid  = 196622
107     OrigLeft    = 2520
108     OrigTop     = 4080
109     OrigRight   = 2760
110     OrigBottom  = 4335
111     Max         = 10000
112     Min         = -10000
113     SyncBuddy   = -1 'True
114     BuddyProperty = 0
115     Enabled     = -1 'True
116     End
117     Begin VB.TextBox Text7
118         Height      = 285
119         Left        = 2040
120         TabIndex    = 16
121         Text        = "Text1"
122         Top         = 2880
123         Width       = 1695
124     End
125     Begin VB.TextBox Text4
126         Height      = 285
127         Left        = 4440
128         TabIndex    = 13
129         Text        = "Text1"
130         Top         = 2520
131         Width       = 1695
132     End
133     Begin MSComCtl2.UpDown UpDown1
134         Height      = 285
135         Left        = 6256
136         TabIndex    = 4
137         TabStop     = 0 'False
138         Top         = 480
139         Width       = 240
140         _ExtentX    = 423
141         _ExtentY    = 503
142         _Version     = 393216
```

```
143     Value        = 1
144     BuddyControl  = "Text5"
145     BuddyDispid   = 196619
146     OrigLeft     = 4560
147     OrigTop      = 3600
148     OrigRight    = 4800
149     OrigBottom   = 3855
150     Max         = 10000
151     Min         = 1
152     SyncBuddy    = -1 'True
153     BuddyProperty = 0
154     Enabled     = -1 'True
155     End
156     Begin VB.TextBox Text1
157         Height      = 285
158         Left        = 2040
159         TabIndex    = 10
160         Text        = "Text1"
161         Top         = 2520
162         Width       = 1695
163     End
164     Begin VB.TextBox Text3
165         Height      = 285
166         Left        = 1800
167         TabIndex    = 1
168         Text        = "Text2"
169         Top         = 0
170         Width       = 4695
171     End
172     Begin VB.TextBox Text5
173         Height      = 285
174         Left        = 1800
175         TabIndex    = 3
176         Text        = "Text2"
177         Top         = 480
178         Width       = 4455
179     End
180     Begin VB.TextBox Text6
181         Height      = 285
182         Left        = 1800
183         TabIndex    = 6
184         Text        = "Text2"
185         Top         = 1440
186         Width       = 4455
187     End
188     Begin VB.Frame Tip1
189         BackColor   = &H80000018&
190         BorderStyle = 0 'Kein
```

```
191 Caption = "Frame2"
192 Height = 495
193 Left = 1800
194 TabIndex = 29
195 Top = 840
196 Width = 4695
197 Begin VB.Image Tip1Image
198 Height = 240
199 Left = 50
200 Picture = "frmEditBohrung.frx":0038
201 Stretch = -1 'True
202 Top = 0
203 Width = 240
204 End
205 Begin VB.Label Tip1Label
206 AutoSize = -1 'True
207 BackStyle = 0 'Transparent
208 Caption = "Die Summe der Bruchstücke
übersteigt die Kernmarschlänge."
209 Height = 390
210 Left = 360
211 TabIndex = 30
212 Top = 0
213 Width = 4215
214 WordWrap = -1 'True
215 End
216 End
217 Begin VB.Frame Tip2
218 BackColor = &H80000018&
219 BorderStyle = 0 'Kein
220 Caption = "Frame2"
221 Height = 495
222 Left = 1800
223 TabIndex = 27
224 Top = 1800
225 Width = 4695
226 Begin VB.Label Tip2Label
227 AutoSize = -1 'True
228 BackStyle = 0 'Transparent
229 Caption = "Der Kernverlust übersteigt
die Differenz zwischen der Kernmarschlänge und den
Bruchstücken."
230 Height = 390
231 Left = 360
232 TabIndex = 28
233 Top = 0
234 Width = 4215
```

```
235 WordWrap = -1 'True
236 End
237 Begin VB.Image Tip2Image
238 Height = 240
239 Left = 45
240 Picture = "frmEditBohrung.frx":047A
241 Stretch = -1 'True
242 Top = 0
243 Width = 240
244 End
245 End
246 Begin MSComCtl2.UpDown UpDown3
247 Height = 285
248 Left = 6255
249 TabIndex = 7
250 TabStop = 0 'False
251 Top = 1440
252 Width = 240
253 _ExtentX = 423
254 _ExtentY = 503
255 Version = 393216
256 Value = 120
257 BuddyControl = "Text6"
258 BuddyDispid = 196618
259 OrigLeft = 5880
260 OrigTop = 2280
261 OrigRight = 6120
262 OrigBottom = 2535
263 Increment = 10
264 Max = 400
265 Min = 50
266 SyncBuddy = -1 'True
267 BuddyProperty = 0
268 Enabled = -1 'True
269 End
270 Begin MSComCtl2.UpDown UpDown4
271 Height = 285
272 Left = 6135
273 TabIndex = 14
274 TabStop = 0 'False
275 Top = 2520
276 Width = 240
277 _ExtentX = 423
278 _ExtentY = 503
279 Version = 393216
280 BuddyControl = "Text4"
281 BuddyDispid = 196626
282 OrigLeft = 2760
```

```
283 OrigTop = 4080
284 OrigRight = 3000
285 OrigBottom = 4335
286 Max = 10000
287 Min = -10000
288 SyncBuddy = -1 'True
289 BuddyProperty = 0
290 Enabled = -1 'True
291 End
292 Begin MSComCtl2.UpDown UpDown5
293 Height = 285
294 Left = 3735
295 TabIndex = 17
296 TabStop = 0 'False
297 Top = 2880
298 Width = 240
299 ExtentX = 423
300 ExtentY = 503
301 _Version = 393216
302 BuddyControl = "Text7"
303 BuddyDispid = 196636
304 OrigLeft = 3000
305 OrigTop = 4080
306 OrigRight = 3240
307 OrigBottom = 4335
308 Max = 10000
309 Min = -10000
310 SyncBuddy = -1 'True
311 BuddyProperty = 0
312 Enabled = -1 'True
313 End
314 Begin VB.Label Label9
315 Alignment = 1 'Rechts
316 Caption = "Bohr&werkzeug:"
317 Height = 255
318 Left = 0
319 TabIndex = 20
320 Top = 3720
321 Width = 1695
322 End
323 Begin VB.Label Label8
324 Alignment = 1 'Rechts
325 Caption = "Bohr&verfahren:"
326 Height = 255
327 Left = 0
328 TabIndex = 18
329 Top = 3360
```

```
330 Width = 1695
331 End
332 Begin VB.Line Line3
333 BorderColor = &H80000010&
334 X1 = 0
335 X2 = 6480
336 Y1 = 3240
337 Y2 = 3240
338 End
339 Begin VB.Label Label7
340 Alignment = 2 'Zentriert
341 Caption = "&Z"
342 Height = 255
343 Left = 1800
344 TabIndex = 15
345 Top = 2880
346 Width = 195
347 End
348 Begin VB.Label Label3
349 Alignment = 2 'Zentriert
350 Caption = "&Y"
351 Height = 255
352 Left = 4200
353 TabIndex = 12
354 Top = 2520
355 Width = 195
356 End
357 Begin VB.Label Label1
358 Alignment = 2 'Zentriert
359 Caption = "&X"
360 Height = 255
361 Left = 1800
362 TabIndex = 9
363 Top = 2520
364 Width = 195
365 End
366 Begin VB.Line Line2
367 BorderColor = &H80000010&
368 X1 = 0
369 X2 = 6480
370 Y1 = 2400
371 Y2 = 2400
372 End
373 Begin VB.Label Label2
374 Alignment = 1 'Rechts
375 Caption = "&Name der Bohrung:"
376 Height = 255
377 Left = 0
```

```

378         TabIndex      = 0
379         Top           = 0
380         Width        = 1695
381     End
382     Begin VB.Label Label4
383         Alignment     = 1 'Rechts
384         Caption       = "&Teufe [m]:"
385         Height        = 255
386         Left          = 0
387         TabIndex     = 2
388         Top           = 480
389         Width        = 1695
390     End
391     Begin VB.Label Label5
392         Alignment     = 1 'Rechts
393         Caption       = "Kern&durchmesser [mm]:"
394         Height        = 255
395         Left          = 0
396         TabIndex     = 5
397         Top           = 1440
398         Width        = 1695
399     End
400     Begin VB.Line Line1
401         BorderColor   = &H80000010&
402         X1            = 0
403         X2            = 6480
404         Y1            = 360
405         Y2            = 360
406     End
407     Begin VB.Label Label6
408         Alignment     = 1 'Rechts
409         Caption       = "&Koordinaten:"
410         Height        = 255
411         Left          = 0
412         TabIndex     = 8
413         Top           = 2520
414         Width        = 1695
415     End
416     End
417     Begin VB.TextBox Text2
418         Height        = 5175
419         Left          = -74880
420         MultiLine    = -1 'True
421         ScrollBars   = 2 'Vertikal
422         TabIndex     = 25
423         Text         = "frmEditBohrung.frx":08BC
424         Top           = 480

```

```

425         Width        = 6495
426     End
427     End
428     Begin MSComctlLib.StatusBar StatusBar1
429         Align        = 2 'Unten ausrichten
430         Height       = 375
431         Left         = 0
432         TabIndex    = 32
433         Top          = 5085
434         Width       = 7275
435         ExtentX     = 12832
436         ExtentY     = 661
437         _Version    = 393216
438         BeginProperty Panels {8E3867A5-8586-11D1-B16A-
00C0F0283628}
439             NumPanels = 1
440             BeginProperty Panel1 {8E3867AB-8586-11D1-B16A-
00C0F0283628}
441                 AutoSize = 1
442                 Bevel = 0
443                 Object.Width = 12330
444             EndProperty
445         EndProperty
446     End
447     End
448     Attribute VB Name = "frmEditBohrung"
449     Attribute VB GlobalNameSpace = False
450     Attribute VB Creatable = False
451     Attribute VB PredeclaredId = True
452     Attribute VB_Exposed = False
453     Option Explicit
454     Public LastExitOK As Boolean
455
456     Private ShowTip1 As Boolean
457     Private ShowTip2 As Boolean
458
459     Private Sub SelTextBox(TextBox As TextBox)
460         With TextBox
461             .SelStart = 0
462             .SelLength = Len(.Text)
463         End With
464     End Sub
465
466     Private Sub Command1 Click(Index As Integer)
467         SSTab1.Tab = 0
468         SSTab1 Click 0
469
470         Text3.SetFocus

```

```
471
472     LastExitOK = (Index = 1)
473     Hide
474 End Sub
475
476
477 Private Sub Form_Resize()
478     On Error Resume Next
479
480     Dim R As Double
481     R = SStabl.Left
482
483     SStabl.Move R, R, ScaleWidth - 2 * R, ScaleHeight -
484     StatusBar1.Height - Frame1.Height - R
485     Select Case SStabl.Tab
486     Case 0
487         Picture1.Move R, Picture1.Top, SStabl.Width - 2 * R,
488         SStabl.Height - Picture1.Top - R
489         Picture1.Enabled = True
490         Text2.Enabled = False
491     Case 1
492         Text2.Move R, Picture1.Top, SStabl.Width - 2 * R,
493         SStabl.Height - Picture1.Top - R
494         Picture1.Enabled = False
495         Text2.Enabled = True
496     End Select
497
498     Frame1.Move ScaleWidth - R - Frame1.Width, SStabl.Top +
499     SStabl.Height + R / 2
500     Picture1.Resize
501 End Sub
502
503 Private Sub CheckIntegrity()
504     ShowTip1 = False
505     If CMe(Text5) = 0 Then
506         Tip1Label.Caption = "Bitte geben Sie eine Teufe an."
507         ShowTip1 = True
508     End If
509     If CMe(Text5) < 0 Then
510         Tip1Label.Caption = "Eine Teufe kann keinen negativen Wert
511         annehmen"
512         ShowTip1 = True
513     End If
514     If CMe(Text5) > 500 Then
515         Tip1Label.Caption = "Bitte prüfen Sie die Einganbe der
516         Teufe. Die Teufe wird in Metern angegeben."
```

```
513     ShowTip1 = True
514 End If
515
516     ShowTip2 = False
517     If CMe(Text6) = 0 Then
518         Tip2Label.Caption = "Bitte geben Sie einen Kerndurchmesser
519         an."
520         ShowTip2 = True
521     End If
522     If CMe(Text6) < 0 Then
523         Tip2Label.Caption = "Ein Kerndurchmesser kann nicht negativ
524         werden."
525         ShowTip2 = True
526     End If
527     If CMe(Text6) < 50 Or CMe(Text6) > 400 Then
528         Tip2Label.Caption = "Bitte prüfen Sie den Kerndurchmesser.
529         Der Kerndurchmesser wird in Millimetern angegeben."
530         ShowTip2 = True
531     End If
532
533     Form.Resize
534 End Sub
535 Private Sub Picture1_Resize()
536     On Error Resume Next
537
538     Dim R As Double, L As Double, W As Double
539     R = Screen.TwipsPerPixelX * 10
540     L = Label2.Left + Label2.Width + R
541     W = Picture1.ScaleWidth - R
542     If W < L + 20 * R Then W = L + 20 * R
543     W = W - L
544
545     Dim T As Double, H As Double
546     T = 0
547
548     '1. Abschnitt
549     Label2.Top = T
550     Text3.Move L, T, W
551
552     T = T + Text3.Height + R
553
554     With Line1
555     .X1 = R
556     .X2 = L + W
557     .Y1 = T - R / 2
558     .Y2 = .Y1
559 End With
```

```
558 '2. Abschnitt
559
560 Label4.Top = T
561 Text5.Move L, T, W - UpDown1.Width
562 UpDown1.Move L + W - UpDown1.Width, T
563 T = T + Text5.Height
564
565 If ShowTip1 Then
566     Tip1Label.Width = W - 4 * Tip1Image.Left - Tip1Image.Width
567     Tip1Label.AutoSize = True
568     Tip1.Move L, T, W, IIf(Tip1Label.Height > Tip1Image.Height +
Screen.TwipsPerPixelY, Tip1Label.Height, Tip1Image.Height +
Screen.TwipsPerPixelY)
569     T = T + Tip1.Height
570 End If
571 Tip1.Visible = ShowTip1
572
573 T = T + R / 2
574
575 Label5.Top = T
576 Text6.Move L, T, W - UpDown3.Width
577 UpDown3.Move L + W - UpDown3.Width, T
578 T = T + Text6.Height
579
580 If ShowTip2 Then
581     Tip2Label.Width = W - 4 * Tip2Image.Left - Tip2Image.Width
582     Tip2Label.AutoSize = True
583     Tip2.Move L, T, W, IIf(Tip2Label.Height > Tip2Image.Height +
Screen.TwipsPerPixelY, Tip2Label.Height, Tip2Image.Height +
Screen.TwipsPerPixelY)
584     T = T + Tip2.Height
585 End If
586 Tip2.Visible = ShowTip2
587
588 T = T + R
589
590 With Line2
591     .X1 = R
592     .X2 = L + W
593     .Y1 = T - R / 2
594     .Y2 = .Y1
595 End With
596
597 '3. Abschnitt
598 Label6.Top = T
599 Label11.Move L, T
600 Text1.Move L + Label11.Width, T, W / 2 - UpDown2.Width -
Label13.Width * 1.5
```

```
601 UpDown2.Move Text1.Left + Text1.Width, T
602
603 Label13.Move UpDown2.Left + UpDown2.Width + Label13.Width, T
604 Text4.Move Label13.Left + Label13.Width, T, Text1.Width
605 UpDown4.Move Text4.Left + Text4.Width, T
606 T = T + Text1.Height + R / 2
607
608 Label7.Move Label11.Left, T
609 Text7.Move Text1.Left, T, Text1.Width
610 UpDown5.Move Text7.Left + Text7.Width, T
611 T = T + Text7.Height + R
612
613 With Line3
614     .X1 = R
615     .X2 = L + W
616     .Y1 = T - R / 2
617     .Y2 = .Y1
618 End With
619
620 '4. Abschnitt
621 Label8.Top = T
622 Combo1.Move L, T, W
623 T = T + Combo1.Height + R / 2
624
625 Label9.Top = T
626 Combo2.Move L, T, W
627
628 End Sub
629
630 Private Sub SStab1_Click(PreviousTab As Integer)
631     Form_Resize
632 End Sub
633
634 Private Sub Text3 GotFocus()
635     SelTextBox Text3
636 End Sub
637 Private Sub Text5_GotFocus()
638     SelTextBox Text5
639 End Sub
640 Private Sub Text6 GotFocus()
641     SelTextBox Text6
642 End Sub
643 Private Sub Text1_GotFocus()
644     SelTextBox Text1
645 End Sub
646 Private Sub Text4 GotFocus()
647     SelTextBox Text4
648 End Sub
```

```

649 Private Sub Text7_GotFocus()
650     SelTextBox Text7
651 End Sub
652
653 Private Sub Text5_Change()
654     Call CheckIntegrity
655 End Sub
656
657 Private Sub Text6_Change()
658     Call CheckIntegrity
659 End Sub
660
661

```

Datei: frmEditKernmarsch.frm (VB Form)

```

01 VERSION 5.00
02 Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0";
03   "MSCOMCTL.OCX"
04 Object = "{86CF1D34-0C5F-11D2-A9FC-0000F8754DA1}#2.0#0";
05   "MSCOMCTL2.OCX"
06 Object = "{BDC217C8-ED16-11CD-956C-0000C04E4C0A}#1.1#0";
07   "TABCTL32.OCX"
08
09 Begin VB.Form frmEditKernmarsch
10     Caption           = "Kernmarsch bearbeiten"
11     ClientHeight      = 6915
12     ClientLeft        = 60
13     ClientTop         = 345
14     ClientWidth       = 6855
15     ControlBox        = 0 'False
16     LinkTopic         = "Form1"
17     LockControls      = -1 'True
18     ScaleHeight       = 6915
19     ScaleWidth        = 6855
20     StartUpPosition  = 1 'Fenstermitte
21     Begin TabDlg.SSTab SSTab1
22         Height        = 5775
23         Left          = 120
24         TabIndex      = 22
25         Top           = 120
26         Width         = 6735
27         ExtentX       = 11880
28         ExtentY       = 10186
29         _Version      = 393216
30         Style         = 1
31         Tabs          = 2
32         TabHeight     = 520

```

```

29     TabCaption(0)    = "Allgemeine &Daten"
30     TabPicture(0)    = "frmEditKernmarsch.frx":0000
31     Tab(0).ControlEnabled= -1 'True
32     Tab(0).Control(0)= "Picture1"
33     Tab(0).Control(0).Enabled= 0 'False
34     Tab(0).ControlCount= 1
35     TabCaption(1)    = "&Bemerkungen"
36     TabPicture(1)    = "frmEditKernmarsch.frx":001C
37     Tab(1).ControlEnabled= 0 'False
38     Tab(1).Control(0)= "Text2"
39     Tab(1).ControlCount= 1
40     Begin VB.TextBox Text2
41         Height        = 5175
42         Left          = -74880
43         MultiLine     = -1 'True
44         ScrollBars    = 2 'Vertikal
45         TabIndex      = 17
46         Text          = "frmEditKernmarsch.frx":0038
47         Top           = 480
48         Width         = 6495
49     End
50     Begin VB.PictureBox Picture1
51         BorderStyle    = 0 'Kein
52         Height         = 5175
53         Left           = 120
54         ScaleHeight    = 5175
55         ScaleWidth     = 6495
56         TabIndex      = 0
57         TabStop        = 0 'False
58         Top           = 480
59         Width         = 6495
60     Begin VB.Frame Tip2
61         BackColor      = &H80000018&
62         BorderStyle    = 0 'Kein
63         Caption        = "Frame2"
64         Height         = 495
65         Left           = 1800
66         TabIndex      = 24
67         Top           = 2160
68         Visible       = 0 'False
69         Width         = 4695
70     Begin VB.Image Tip2Image
71         Height         = 240
72         Left          = 50
73         Picture        = "frmEditKernmarsch.frx":003E
74         Stretch        = -1 'True
75         Top           = 0
76         Width         = 240

```

```
77         End
78         Begin VB.Label Tip2Label
79             AutoSize           = -1 'True
80             BackStyle          = 0 'Transparent
81             Caption             = "Der Kernverlust übersteigt
die Differenz zwischen der Kernmarschlänge und den
Bruchstücken."
82             Height              = 390
83             Left                 = 360
84             TabIndex            = 13
85             Top                  = 0
86             Width                = 4215
87             WordWrap            = -1 'True
88         End
89     End
90     Begin VB.Frame Tip1
91         BackColor              = &H80000018&
92         BorderStyle            = 0 'Kein
93         Caption                 = "Frame2"
94         Height                  = 495
95         Left                    = 1800
96         TabIndex                = 23
97         Top                     = 1200
98         Width                   = 4695
99     Begin VB.Label Tip1Label
100        AutoSize                = -1 'True
101        BackStyle                = 0 'Transparent
102        Caption                   = "Die Summe der Bruchstücke
übersteigt die Kernmarschlänge."
103        Height                    = 390
104        Left                      = 360
105        TabIndex                  = 9
106        Top                      = 0
107        Width                     = 4215
108        WordWrap                  = -1 'True
109    End
110    Begin VB.Image Tip1Image
111        Height                    = 240
112        Left                      = 50
113        Picture                   = "frmEditKernmarsch.frx":0480
114        Stretch                    = -1 'True
115        Top                      = 0
116        Width                     = 240
117    End
118    End
119    Begin VB.TextBox Text6
120        Height                    = 285
```

```
121        Left                    = 1800
122        TabIndex                  = 11
123        Text                      = "Text2"
124        Top                      = 1800
125        Visible                    = 0 'False
126        Width                     = 4455
127    End
128    Begin VB.TextBox Text5
129        Height                    = 285
130        Left                      = 1800
131        TabIndex                  = 8
132        Text                      = "Text2"
133        Top                      = 840
134        Width                     = 4695
135    End
136    Begin VB.TextBox Text4
137        Height                    = 285
138        Left                      = 1800
139        TabIndex                  = 5
140        Text                      = "Text2"
141        Top                      = 360
142        Width                     = 4455
143    End
144    Begin VB.TextBox Text3
145        Height                    = 285
146        Left                      = 1800
147        TabIndex                  = 2
148        Text                      = "Text2"
149        Top                      = 0
150        Width                     = 4455
151    End
152    Begin VB.TextBox Text1
153        Height                    = 285
154        Left                      = 1800
155        TabIndex                  = 15
156        Text                      = "Text1"
157        Top                      = 2760
158        Width                     = 4695
159    End
160    Begin Gebirgsklassifikation.ct1Parameter ctlParameter1
161        Height                    = 2055
162        Left                      = 1800
163        TabIndex                  = 16
164        Top                      = 3120
165        Width                     = 4695
166        ExtentX                   = 8281
167        ExtentY                   = 3625
168    End
```

```

169     Begin MSComCtl2.UpDown UpDown3
170         Height      = 285
171         Left        = 6255
172         TabIndex    = 12
173         TabStop     = 0 'False
174         Top         = 1800
175         Visible     = 0 'False
176         Width       = 240
177         ExtentX     = 423
178         ExtentY     = 503
179         Version     = 393216
180         BuddyControl = "Tip2Label"
181         BuddyDispid  = 196613
182         OrigLeft    = 5880
183         OrigTop     = 2280
184         OrigRight   = 6120
185         OrigBottom  = 2535
186         Max         = 1000
187         SyncBuddy   = -1 'True
188         BuddyProperty = 0
189         Enabled     = -1 'True
190     End
191     Begin MSComCtl2.UpDown UpDown2
192         Height      = 285
193         Left        = 6255
194         TabIndex    = 6
195         TabStop     = 0 'False
196         Top         = 360
197         Width       = 240
198         _ExtentX   = 423
199         _ExtentY   = 503
200         Version     = 393216
201         Value       = 1
202         BuddyControl = "Text6"
203         BuddyDispid  = 196617
204         OrigLeft    = 5280
205         OrigTop     = 840
206         OrigRight   = 5520
207         OrigBottom  = 1095
208         Max         = 5
209         Min         = 1
210         SyncBuddy   = -1 'True
211         BuddyProperty = 0
212         Enabled     = -1 'True
213     End
214     Begin MSComCtl2.UpDown UpDown1
215         Height      = 285
216         Left        = 6255

```

```

217         TabIndex    = 3
218         TabStop     = 0 'False
219         Top         = 0
220         Width       = 240
221         ExtentX     = 423
222         _ExtentY   = 503
223         _Version   = 393216
224         BuddyControl = "Text3"
225         BuddyDispid  = 196620
226         OrigLeft    = 4560
227         OrigTop     = 720
228         OrigRight   = 4800
229         OrigBottom  = 975
230         Max         = 5000
231         SyncBuddy   = -1 'True
232         BuddyProperty = 0
233         Enabled     = -1 'True
234     End
235     Begin VB.Label Label6
236         Alignment    = 1 'Rechts
237         Caption      = "&Parameter:"
238         Height       = 255
239         Left         = 0
240         TabIndex    = 14
241         Top         = 2760
242         Width       = 1695
243     End
244     Begin VB.Line Line1
245         BorderColor  = &H80000010&
246         X1           = 0
247         X2           = 6480
248         Y1           = 720
249         Y2           = 720
250     End
251     Begin VB.Label Label5
252         Alignment    = 1 'Rechts
253         Caption      = "Kern&verlust [cm]:"
254         Height       = 255
255         Left         = 0
256         TabIndex    = 10
257         Top         = 1800
258         Visible     = 0 'False
259         Width       = 1695
260     End
261     Begin VB.Label Label4
262         Alignment    = 1 'Rechts
263         Caption      = "&Bruchstücke [cm]:"
264         Height       = 255

```

```
265         Left           = 0
266         TabIndex      = 7
267         Top           = 840
268         Width         = 1695
269     End
270     Begin VB.Label Label3
271         Alignment      = 1 'Rechts
272         Caption        = "Kernmarsch&länge [m]:"
273         Height         = 255
274         Left           = 0
275         TabIndex      = 4
276         Top           = 360
277         Width         = 1695
278     End
279     Begin VB.Label Label2
280         Alignment      = 1 'Rechts
281         Caption        = "&Teufe [m]:"
282         Height         = 255
283         Left           = 0
284         TabIndex      = 1
285         Top           = 0
286         Width         = 1695
287     End
288 End
289 Begin VB.Frame Frame1
290     BorderStyle      = 0 'Kein
291     Caption          = "Frame1"
292     Height           = 375
293     Left             = 3600
294     TabIndex        = 21
295     Top              = 6120
296     Width            = 3015
297 End
298 Begin VB.CommandButton Command1
299     Cancel           = -1 'True
300     Caption          = "&Abbrechen"
301     Height           = 375
302     Index            = 0
303     Left             = 1560
304     TabIndex        = 19
305     Top              = 0
306     Width            = 1455
307 End
308 Begin VB.CommandButton Command1
309     Caption          = "&Ok"
310     Default          = -1 'True
311     Height           = 375
```

```
312         Index         = 1
313         Left           = 0
314         TabIndex      = 18
315         Top           = 0
316         Width         = 1455
317     End
318 End
319 Begin MSComctlLib.StatusBar StatusBar1
320     Align            = 2 'Unten ausrichten
321     Height           = 375
322     Left             = 0
323     TabIndex        = 20
324     Top              = 6540
325     Width            = 6855
326     ExtentX          = 12091
327     ExtentY          = 661
328     Version          = 393216
329     BeginProperty Panels {8E3867A5-8586-11D1-B16A-
00C0F0283628}
330         NumPanels      = 1
331         BeginProperty Panel1 {8E3867AB-8586-11D1-B16A-
00C0F0283628}
332             AutoSize      = 1
333             Bevel         = 0
334             Object.Width   = 11589
335         EndProperty
336     EndProperty
337 End
338 End
339 Attribute VB_Name = "frmEditKernmarsch"
340 Attribute VB_GlobalNameSpace = False
341 Attribute VB_Creatable = False
342 Attribute VB_PredeclaredId = True
343 Attribute VB_Exposed = False
344 Option Explicit
345
346 Public LastExitOK As Boolean
347
348 Private ShowTip1 As Boolean
349 Private ShowTip2 As Boolean
350
351 Private Sub GetParameter()
352     Text1 = RockClassX.Dictionary2String(ctlParameter1.Parameter)
353 End Sub
354
355 Private Sub PutParameters()
356     Set ctlParameter1.Parameter =
RockClassX.String2Dictionary(Text1)
```

```
357 End Sub
358
359 Private Sub Command1 Click(Index As Integer)
360     On Error Resume Next
361
362     SSTab1.Tab = 0
363     SSTab1_Click 0
364
365     Text3.SetFocus
366
367     LastExitOK = (Index = 1)
368     Hide
369 End Sub
370
371 Private Sub ctlParameter1 Change()
372     Call GetParameter
373 End Sub
374
375 Private Sub ctlParameter1_GotFocus()
376     Command1(1).Default = False
377     Command1(0).Cancel = False
378 End Sub
379
380 Private Sub ctlParameter1_LostFocus()
381     Command1(1).Default = True
382     Command1(0).Cancel = True
383 End Sub
384
385 Private Sub Form Load()
386     Dim StdParameter As Collection
387
388     Set StdParameter = New Collection
389     With StdParameter
390         .Add "Qu"
391         .Add "Ra"
392         .Add "Rb"
393         .Add "Rc"
394         .Add "Rd"
395         .Add "Alpha"
396         .Add "Beta"
397         .Add "F3"
398         .Add "F4"
399         .Add "Jn"
400         .Add "Jr"
401         .Add "Ja"
402         .Add "Jw"
403         .Add "SRF"
404         .Add "p"
```

```
405     .Add "H"
406 End With
407
408 With ctlParameter1
409     Set .StandardParameter = StdParameter
410     Set .ClassificationCollection = ClassificationCollection
411
412     Set .Parameter = Nothing
413 End With
414
415 SSTab1.Tab = 0
416 SSTab1_Click 0
417 End Sub
418
419 Private Sub Form Resize()
420     On Error Resume Next
421
422     Dim R As Double
423     R = SSTab1.Left
424
425     SSTab1.Move R, R, ScaleWidth - 2 * R, ScaleHeight -
426     StatusBar1.Height - Frame1.Height - R
427     Select Case SSTab1.Tab
428     Case 0
429         Picture1.Move R, Picture1.Top, SSTab1.Width - 2 * R,
430         SSTab1.Height - Picture1.Top - R
431         Picture1.Enabled = True
432         Text2.Enabled = False
433
434     Case 1
435         Text2.Move R, Picture1.Top, SSTab1.Width - 2 * R,
436         SSTab1.Height - Picture1.Top - R
437         Picture1.Enabled = False
438         Text2.Enabled = True
439
440     End Select
441
442     Frame1.Move ScaleWidth - R - Frame1.Width, SSTab1.Top +
443     SSTab1.Height + R / 2
444     Picture1.Resize
445 End Sub
446
447 Private Sub Picture1_Resize()
448     On Error Resume Next
449
450     Dim R As Double, L As Double, W As Double
451     R = Screen.TwipsPerPixelX * 10
452     L = Label2.Left + Label2.Width + R
```

```
449 W = Picture1.ScaleWidth - R
450 If W < L + 20 * R Then W = L + 20 * R
451 W = W - L
452
453 Dim T As Double, H As Double
454 T = 0
455
456 '1. Abschnitt
457 Label2.Top = T
458 Text3.Move L, T, W - UpDown1.Width
459 UpDown1.Move L + W - UpDown1.Width, T
460
461 T = T + Text3.Height + R / 2
462
463 Label3.Top = T
464 Text4.Move L, T, W - UpDown2.Width
465 UpDown2.Move L + W - UpDown2.Width, T
466
467 T = T + Text4.Height + R
468 With Line1
469     .X1 = R
470     .X2 = L + W
471     .Y1 = T - R / 2
472     .Y2 = .Y1
473 End With
474
475 '2. Abschnitt
476
477 Label4.Top = T
478 Text5.Move L, T, W
479
480 T = T + Text5.Height
481
482 If ShowTip1 Then
483     Tip1Label.Width = W - 4 * Tip1Image.Left - Tip1Image.Width
484     Tip1Label.AutoSize = True
485     Tip1.Move L, T, W, IIf(Tip1Label.Height > Tip1Image.Height +
Screen.TwipsPerPixelY, Tip1Label.Height, Tip1Image.Height +
Screen.TwipsPerPixelY)
486     T = T + Tip1.Height
487 End If
488 Tip1.Visible = ShowTip1
489
490 T = T + R / 2
491
492 ' Label5.Top = T
493 ' Text6.Move L, T, W - UpDown3.Width
494 ' UpDown3.Move L + W - UpDown3.Width, T
```

```
495 '
496 ' T = T + Text6.Height
497 '
498 ' If ShowTip2 Then
499 '     Tip2Label.Width = W - 4 * Tip2Image.Left - Tip2Image.Width
500 '     Tip2Label.AutoSize = True
501 '     Tip2.Move L, T, W, IIf(Tip2Label.Height > Tip2Image.Height
+ Screen.TwipsPerPixelY, Tip2Label.Height, Tip2Image.Height +
Screen.TwipsPerPixelY)
502 '     T = T + Tip2.Height
503 ' End If
504 ' Tip2.Visible = ShowTip2
505 '
506 ' T = T + R / 2
507
508 Label6.Top = T
509 Text1.Move L, T, W
510
511 T = T + Text1.Height
512
513 H = Picture1.ScaleHeight - T - StatusBar1.Height + R
514 If H < 10 * R Then H = 10 * R
515 ctlParameter1.Move L, T, W, H
516 End Sub
517
518 Private Sub SStab1_Click(PreviousTab As Integer)
519     Form Resize
520 End Sub
521
522 Private Sub Text1_GotFocus()
523     With Text1
524         .SelStart = 0
525         .SelLength = Len(.Text)
526     End With
527 End Sub
528 Private Sub Text3_GotFocus()
529     With Text3
530         .SelStart = 0
531         .SelLength = Len(.Text)
532     End With
533 End Sub
534 Private Sub Text4_GotFocus()
535     With Text4
536         .SelStart = 0
537         .SelLength = Len(.Text)
538     End With
539 End Sub
540 Private Sub Text5_GotFocus()
```

```

541     With Text5
542         .SelStart = 0
543         .SelLength = Len(.Text)
544     End With
545 End Sub
546 Private Sub Text6_GotFocus()
547     With Text6
548         .SelStart = 0
549         .SelLength = Len(.Text)
550     End With
551 End Sub
552
553 Public Sub Text1_KeyUp(KeyCode As Integer, Shift As Integer)
554     Call PutParameters
555 End Sub
556
557 Private Sub Text4_Change()
558     Call CheckIntegrity
559 End Sub
560
561 Private Sub Text5_Change()
562     Call CheckIntegrity
563 End Sub
564
565 Private Sub Text6_Change()
566     Call CheckIntegrity
567 End Sub
568
569 Private Sub CheckIntegrity()
570     Dim L As Double, Lx As Double
571
572     L = RockClassX.AddPieceLength(Text5.Text, True, True, True,
573     True, True, True)
574     Lx = RockClassX.AddPieceLength(Text5.Text, True, True, True,
575     True, False, True)
576
577     ShowTip1 = False
578     ShowTip2 = False
579     If Lx + CMe(Text6.Text) > CMe(Text4) * 100 Then
580         Tip2Label.Caption = "Der Kernverlust übersteigt die
581         Differenz zwischen der Kernmarschlänge und den eingegebenen
582         Bruchstücken."
583         ShowTip2 = True
584     End If
585
586     If L = 0 Then

```

```

584         Tip1Label.Caption = "Bitte geben Sie die erhaltenen
585         Kernbruchstücke ein."
586         ShowTip1 = True
587     End If
588     If L > CMe(Text4.Text) * 100 Then
589         Tip1Label.Caption = "Die Summe der Längen der eingegebenen
590         Kernbruchstücke übersteigt die Kernmarschlänge."
591         ShowTip1 = True
592         ShowTip2 = False
593     End If
594
595     If CMe(Text6) > CMe(Text4) * 100 Then
596         Tip2Label.Caption = "Der Kernverlust übersteigt 100%."
597         ShowTip2 = True
598     End If
599     If CMe(Text6) < 0 Then
600         Tip2Label.Caption = "Der Kernverlust ist negativ."
601         ShowTip2 = True
602     End If
603
604     Form Resize
605 End Sub

```

Datei: frmAssistentKernmärsche.frm (VB Form)

```

01 VERSION 5.00
02 Object = "{86CF1D34-0C5F-11D2-A9FC-0000F8754DA1}#2.0#0";
03 "MSCOMCT2.OCX"
04 Begin VB.Form frmAssistentKernmärsche
05     BorderStyle = 4 'Festes Werkzeugfenster
06     Caption = "Assistent: Kernmärsche hinzufügen"
07     ClientHeight = 4845
08     ClientLeft = 45
09     ClientTop = 285
10     ClientWidth = 7935
11     ControlBox = 0 'False
12     LinkTopic = "Form1"
13     LockControls = -1 'True
14     MaxButton = 0 'False
15     MinButton = 0 'False
16     ScaleHeight = 4845
17     ScaleWidth = 7935
18     ShowInTaskbar = 0 'False
19     StartUpPosition = 1 'Fenstermitte
20     Begin VB.CommandButton Command1

```

```

20     Cancel      = -1 'True
21     Caption     = "&Abbruch"
22     Height      = 375
23     Index       = 2
24     Left        = 5040
25     TabIndex    = 16
26     Top         = 4320
27     Width       = 1335
28 End
29 Begin VB.CommandButton Command1
30     Caption     = "&Ok"
31     Default     = -1 'True
32     Height      = 375
33     Index       = 0
34     Left        = 6480
35     TabIndex    = 15
36     Top         = 4320
37     Width       = 1335
38 End
39 Begin VB.Frame Frame1
40     BorderStyle = 0 'Kein
41     Caption     = "Frame1"
42     Height      = 3255
43     Left        = 120
44     TabIndex    = 17
45     Top         = 960
46     Width       = 7695
47 Begin VB.Frame Frame3
48     Caption     = "2. Vorbelegung definieren"
49     BeginProperty Font
50         Name      = "MS Sans Serif"
51         Size      = 8.25
52         Charset   = 0
53         Weight    = 700
54         Underline = 0 'False
55         Italic    = 0 'False
56         Strikethrough = 0 'False
57     EndProperty
58     Height      = 975
59     Left        = 0
60     TabIndex    = 13
61     Top         = 2280
62     Width       = 7695
63 Begin VB.CommandButton Command2
64     Caption     = "&Vorbelegung anzeigen"
65     Height      = 375
66     Left        = 1800

```

```

67         TabIndex = 14
68         Top      = 360
69         Width    = 3855
70     End
71 End
72 Begin VB.Frame Frame2
73     Caption     = "1. Bereich eingrenzen"
74     BeginProperty Font
75         Name      = "MS Sans Serif"
76         Size      = 8.25
77         Charset   = 0
78         Weight    = 700
79         Underline = 0 'False
80         Italic    = 0 'False
81         Strikethrough = 0 'False
82     EndProperty
83     Height      = 1455
84     Left        = 0
85     TabIndex    = 3
86     Top         = 720
87     Width       = 7695
88 Begin MSComCtl2.UpDown UpDown1
89     Height      = 285
90     Left        = 3136
91     TabIndex    = 18
92     Top         = 240
93     Width       = 240
94     ExtentX     = 423
95     ExtentY     = 503
96     _Version    = 393216
97     BuddyControl = "Text1"
98     BuddyDispid = 196618
99     OrigLeft    = 3720
100    OrigTop     = 960
101    OrigRight   = 3960
102    OrigBottom  = 1215
103    SyncBuddy   = -1 'True
104    BuddyProperty = 0
105    Enabled     = -1 'True
106 End
107 Begin VB.CheckBox Check1
108     Caption     = "Kernmarschlänge mit
109     Schrittweite &gleichsetzen"
110     Height      = 375
111     Left        = 5280
112     TabIndex    = 12
113     Top         = 960
114     Width       = 2295

```

```
114 End
115 Begin VB.TextBox Text4
116     Height = 285
117     Left = 5280
118     TabIndex = 11
119     Text = "Text1"
120     Top = 600
121     Width = 1575
122 End
123 Begin VB.TextBox Text3
124     Height = 285
125     Left = 5280
126     TabIndex = 7
127     Text = "Text1"
128     Top = 240
129     Width = 1575
130 End
131 Begin VB.TextBox Text2
132     Height = 285
133     Left = 1560
134     TabIndex = 9
135     Text = "Text1"
136     Top = 600
137     Width = 1575
138 End
139 Begin VB.TextBox Text1
140     Height = 285
141     Left = 1560
142     TabIndex = 5
143     Text = "Text1"
144     Top = 240
145     Width = 1575
146 End
147 Begin MSComCtl2.UpDown UpDown2
148     Height = 285
149     Left = 3136
150     TabIndex = 19
151     Top = 600
152     Width = 240
153     ExtentX = 423
154     ExtentY = 503
155     _Version = 393216
156     BuddyControl = "Text2"
157     BuddyDispid = 196617
158     OrigLeft = 3960
159     OrigTop = 960
160     OrigRight = 4200
```

```
161     OrigBottom = 1215
162     SyncBuddy = -1 'True
163     BuddyProperty = 0
164     Enabled = -1 'True
165 End
166 Begin MSComCtl2.UpDown UpDown3
167     Height = 285
168     Left = 6856
169     TabIndex = 20
170     Top = 240
171     Width = 240
172     ExtentX = 423
173     ExtentY = 503
174     _Version = 393216
175     Value = 1
176     BuddyControl = "Text3"
177     BuddyDispid = 196616
178     OrigLeft = 4200
179     OrigTop = 960
180     OrigRight = 4440
181     OrigBottom = 1215
182     Min = 1
183     SyncBuddy = -1 'True
184     BuddyProperty = 0
185     Enabled = -1 'True
186 End
187 Begin MSComCtl2.UpDown UpDown4
188     Height = 285
189     Left = 6856
190     TabIndex = 21
191     Top = 600
192     Width = 240
193     ExtentX = 423
194     ExtentY = 503
195     Version = 393216
196     Value = 1
197     BuddyControl = "Text4"
198     BuddyDispid = 196615
199     OrigLeft = 4440
200     OrigTop = 960
201     OrigRight = 4680
202     OrigBottom = 1215
203     Max = 5
204     SyncBuddy = -1 'True
205     BuddyProperty = 0
206     Enabled = -1 'True
207 End
208 Begin VB.Label Label8
```

```

209         Alignment      = 1 'Rechts
210         Caption        = "Kernmarsch&länge [m]:"
211         Height         = 255
212         Left           = 3360
213         TabIndex       = 10
214         Top            = 600
215         Width          = 1815
216     End
217     Begin VB.Label Label7
218         Alignment      = 1 'Rechts
219         Caption        = "Schritt&weite [m]:"
220         Height         = 255
221         Left           = 3360
222         TabIndex       = 6
223         Top            = 240
224         Width          = 1815
225     End
226     Begin VB.Label Label6
227         Alignment      = 1 'Rechts
228         Caption        = "Teufe &Ende [m]:"
229         Height         = 255
230         Left           = 120
231         TabIndex       = 8
232         Top            = 600
233         Width          = 1335
234     End
235     Begin VB.Label Label5
236         Alignment      = 1 'Rechts
237         Caption        = "Teufe &Start [m]:"
238         Height         = 255
239         Left           = 120
240         TabIndex       = 4
241         Top            = 240
242         Width          = 1335
243     End
244     End
245     Begin VB.Label Label2
246         Caption        = $"frmAssistentKernmärsche.frx":0000
247         Height         = 855
248         Left           = 0
249         TabIndex       = 2
250         Top            = 0
251         Width          = 5535
252     End
253     End
254     Begin VB.Label Label1
255         BackStyle       = 0 'Transparent

```

```

256         Caption        = "Kernmärsche hinzufügen"
257     BeginProperty Font
258         Name           = "Arial"
259         Size           = 20.25
260         CharSet        = 0
261         Weight         = 700
262         Underline      = 0 'False
263         Italic         = 0 'False
264         Strikethrough  = 0 'False
265     EndProperty
266     Height            = 495
267     Index            = 0
268     Left             = 360
269     TabIndex         = 1
270     Top              = 360
271     Width            = 5295
272     End
273     Begin VB.Label Label1
274         BackStyle      = 0 'Transparent
275         Caption        = "Assistent"
276     BeginProperty Font
277         Name           = "Arial"
278         Size           = 36
279         CharSet        = 0
280         Weight         = 700
281         Underline      = 0 'False
282         Italic         = 0 'False
283         Strikethrough  = 0 'False
284     EndProperty
285     ForeColor         = &H80000011&
286     Height            = 1215
287     Index            = 1
288     Left             = 120
289     TabIndex         = 0
290     Top              = 0
291     Width            = 3735
292     End
293     End
294     Attribute VB Name = "frmAssistentKernmärsche"
295     Attribute VB GlobalNameSpace = False
296     Attribute VB Creatable = False
297     Attribute VB_PredeclaredId = True
298     Attribute VB_Exposed = False
299     Option Explicit
300
301     Public RSBohrungen As DAO.Recordset
302     Public RSKernmärsche As DAO.Recordset
303     Public BohrungKey As String

```

```
304
305 Private preBruchstücke As String
306 Private preParameter As String
307 Private preBemerkungen As String
308
309 Private Sub Check1_Click()
310     Dim B As Boolean
311
312     B = (Check1.Value = vbChecked)
313     Text4.Enabled = Not B
314     UpDown4.Enabled = Not B
315
316     If B Then
317         Text4.Text = Text3.Text
318     End If
319 End Sub
320
321 Private Sub Command1_Click(Index As Integer)
322     If Index = 0 Then 'Ok
323         Dim ns As Double, ne As Double, nw As Double, nl As Double
324         ns = CMe(Text1)
325         ne = CMe(Text2)
326         nw = CMe(Text3)
327         nl = CMe(Text4)
328
329         If ns > ne Then
330             MsgBox "Das Ende befindet sich vor dem Start.",
vbExclamation, Caption
331             Exit Sub
332         End If
333         If ns = ne Then
334             MsgBox "Das Ende entspricht dem Start. Es würden keine
Kernmärsche hinzugefügt werden.", vbExclamation, Caption
335             Exit Sub
336         End If
337         If nw = 0 Then
338             MsgBox "Die Schrittweite ist Null.", vbExclamation,
Caption
339             Exit Sub
340         End If
341         If nw < 0 Then
342             MsgBox "Die Schrittweite ist negativ.", vbExclamation,
Caption
343             Exit Sub
344         End If
345         If nl = 0 Then
```

```
346             MsgBox "Die Kernmarschlänge ist Null.", vbExclamation,
Caption
347             Exit Sub
348         End If
349         If nl < 0 Then
350             MsgBox "Die Kernmarschlänge ist negativ.", vbExclamation,
Caption
351             Exit Sub
352         End If
353
354         With RSKernmärsche
355             Dim D As Double, M As Long
356             M = 0
357             For D = ns To ne - nw Step nw
358                 M = M + 1
359                 .AddNew
360                 .Fields("BohrungID") = BohrungKey
361
362                 .Fields("Teufe") = D
363                 .Fields("Länge") = nl
364                 If preBruchstücke <> vbNullString Then
365                     .Fields("Bruchstücke") = preBruchstücke
366                 End If
367                 .Fields("Parameter") = preParameter
368                 If preBemerkungen <> vbNullString Then
369                     .Fields("Bemerkungen") = preBemerkungen
370                 End If
371
372                 .Update
373             Next
374             MsgBox "Es wurden " & M & " Kernmärsche hinzugefügt.",
vbInformation, Caption
375             End With
376         End If
377         Unload Me
378     End Sub
379
380 Private Sub Command2_Click()
381     Dim frm As frmEditKernmarsch
382     Set frm = New frmEditKernmarsch
383
384     With frm
385         .Text3.Text = "(automatisch)"
386         .Text3.Enabled = False
387         .Text3.BackColor = vb3DFace
388         .UpDown1.Enabled = False
389         .Text4.Text = "(automatisch)"
390         .Text4.Enabled = False
```

```

391 .Text4.BackColor = vb3DFace
392 .UpDown2.Enabled = False
393
394 .Text5.Text = preBruchstücke
395 .Text1.Text = preParameter: .Text1 KeyUp 0, 0
396 .Text2.Text = preBemerkungen
397
398 .Show 1, Me
399
400 If .LastExitOK Then
401     'Werte übernehmen
402     preBruchstücke = .Text5
403     preParameter = .Text1
404     preBemerkungen = .Text2
405 End If
406 End With
407 End Sub
408
409 Private Sub Form_Load()
410     On Error Resume Next
411     Dim D As Double
412
413     preBruchstücke = vbNullString
414     preParameter = vbNullString
415     preBemerkungen = vbNullString
416
417     RSBohrungen.MoveFirst
418     Do Until RSBohrungen.EOF
419         If RSBohrungen.Fields("BohrungID") = BohrungKey Then
420             D = Val(RSBohrungen.Fields("Teufe") & vbNullString)
421             Text1.Text = "0,00"
422             Text2.Text = Format(D, "0.00")
423             UpDown1.Max = D
424             UpDown2.Max = D
425         End If
426         RSBohrungen.MoveNext
427     Loop
428
429     Text3.Text = "1,00"
430     Check1.Value = vbChecked
431     Call Check1 Click
432 End Sub
433
434 Private Sub Text3_Change()
435     Call Check1 Click
436 End Sub
437

```

Datei: frmOLEObjekt.frm (VB Form)

```

01 VERSION 5.00
02 Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0";
03 "MSCOMCTL.OCX"
04 Object = "{BDC217C8-ED16-11CD-956C-0000C04E4C0A}#1.1#0";
05 "TABCTL32.OCX"
06
07 Begin VB.Form frmOLEObjekt
08     BorderStyle = 5 'Änderbares Werkzeugfenster
09     Caption = "Objekteigenschaften"
10     ClientHeight = 5670
11     ClientLeft = 60
12     ClientTop = 300
13     ClientWidth = 6960
14     ControlBox = 0 'False
15     LinkTopic = "Form1"
16     LockControls = -1 'True
17     MaxButton = 0 'False
18     MinButton = 0 'False
19     ScaleHeight = 5670
20     ScaleWidth = 6960
21     ShowInTaskbar = 0 'False
22     StartupPosition = 1 'Fenstermitte
23
24 Begin VB.CommandButton Command1
25     Cancel = -1 'True
26     Caption = "&Ok"
27     Default = -1 'True
28     Height = 375
29     Left = 5400
30     TabIndex = 9
31     Top = 5040
32     Width = 1455
33
34 End
35
36 Begin TabDlg.SSTab SSTab1
37     Height = 4815
38     Left = 120
39     TabIndex = 5
40     Top = 120
41     Width = 6735
42     ExtentX = 11880
43     ExtentY = 8493
44     Version = 393216
45     Style = 1
46     Tabs = 2
47     TabHeight = 520
48     TabCaption(0) = "All&gemein"
49     TabPicture(0) = "frmOLEObjekt.frx":0000

```

```

44 Tab(0).ControlEnabled= -1 'True
45 Tab(0).Control(0)= "Picture1"
46 Tab(0).Control(0).Enabled= 0 'False
47 Tab(0).ControlCount= 1
48 TabCaption(1) = "&Beschreibung"
49 TabPicture(1) = "frmOLEObjekt.frx":001C
50 Tab(1).ControlEnabled= 0 'False
51 Tab(1).Control(0)= "Text2"
52 Tab(1).ControlCount= 1
53 Begin VB.TextBox Text2
54     Height = 4215
55     Left = -74880
56     TabIndex = 6
57     Text = "Text2"
58     Top = 480
59     Width = 6495
60 End
61 Begin VB.PictureBox Picture1
62     BorderStyle = 0 'Kein
63     Height = 3975
64     Left = 120
65     ScaleHeight = 3975
66     ScaleWidth = 6375
67     TabIndex = 8
68     TabStop = 0 'False
69     Top = 480
70     Width = 6375
71 Begin VB.Data Data1
72     Caption = "Data1"
73     Connect = "Access"
74     DatabaseName =
"E:\JMeier\Dokumente\Studium\Studienarbeit\Kernauswertung
'97.mdb"
75     DefaultCursorType= 0 'Standard-Cursor
76     DefaultType = 2 'ODBC verwenden
77     Exclusive = 0 'False
78     Height = 375
79     Left = 480
80     Options = 0
81     ReadOnly = 0 'False
82     RecordsetType = 1 'Dynaset
83     RecordSource = "Objekte"
84     Top = 1800
85     Visible = 0 'False
86     Width = 1815
87 End
88 Begin VB.TextBox Text1

```

```

89     Height = 285
90     Left = 840
91     TabIndex = 1
92     Text = "Text1"
93     Top = 0
94     Width = 4935
95 End
96 Begin MSComctlLib.ImageList ImageList1
97     Left = 5040
98     Top = 2760
99     ExtentX = 1005
100     ExtentY = 1005
101     BackColor = -2147483643
102     ImageWidth = 16
103     ImageHeight = 16
104     MaskColor = 12632256
105     Version = 393216
106     BeginProperty Images {2C247F25-8591-11D1-B16A-
00C0F0283628}
107         NumListImages = 3
108         BeginProperty ListImage1 {2C247F27-8591-11D1-
B16A-00C0F0283628}
109             Picture = "frmOLEObjekt.frx":0038
110             Key = "new"
111         EndProperty
112         BeginProperty ListImage2 {2C247F27-8591-11D1-
B16A-00C0F0283628}
113             Picture = "frmOLEObjekt.frx":048A
114             Key = "edt"
115         EndProperty
116         BeginProperty ListImage3 {2C247F27-8591-11D1-
B16A-00C0F0283628}
117             Picture = "frmOLEObjekt.frx":05E4
118             Key = "del"
119         EndProperty
120     EndProperty
121 End
122 Begin MSComctlLib.Toolbar Toolbar1
123     Height = 330
124     Left = 840
125     TabIndex = 4
126     Top = 3120
127     Width = 4920
128     ExtentX = 8678
129     ExtentY = 582
130     ButtonWidth = 1138
131     ButtonHeight = 582
132     AllowCustomize = 0 'False

```

```

133     Wrappable      = 0 'False
134     Style          = 1
135     TextAlignment  = 1
136     Version       = 393216
137   End
138   Begin VB.Label Label2
139     Alignment      = 1 'Rechts
140     Caption        = "&Objekt:"
141     Height         = 255
142     Left           = 0
143     TabIndex       = 2
144     Top            = 360
145     Width          = 735
146   End
147   Begin VB.Label Label1
148     Alignment      = 1 'Rechts
149     Caption        = "&Name:"
150     Height         = 255
151     Left           = 0
152     TabIndex       = 0
153     Top            = 0
154     Width          = 735
155   End
156   Begin VB.OLE OLE1
157     DataField      = "Objekt"
158     DataSource     = "Data1"
159     Height         = 2655
160     Left           = 840
161     TabIndex       = 3
162     Top            = 360
163     Width          = 4935
164   End
165 End
166 End
167 Begin MSComctlLib.StatusBar StatusBar1
168   Align           = 2 'Unten ausrichten
169   Height          = 375
170   Left            = 0
171   TabIndex        = 7
172   Top             = 5295
173   Width           = 6960
174   _ExtentX        = 12277
175   _ExtentY        = 661
176   Version         = 393216
177   BeginProperty Panels {8E3867A5-8586-11D1-B16A-
00C0F0283628}
178     NumPanels      = 1

```

```

179     BeginProperty Panel1 {8E3867AB-8586-11D1-B16A-
00C0F0283628}
180       AutoSize      = 1
181       Bevel         = 0
182       Object.Width   = 11774
183     EndProperty
184   EndProperty
185   End
186 End
187 Attribute VB Name = "frmOLEObjekt"
188 Attribute VB GlobalNameSpace = False
189 Attribute VB Creatable = False
190 Attribute VB_PredeclaredId = True
191 Attribute VB_Exposed = False
192 Option Explicit
193
194 Private Sub Command1 Click()
195   Hide
196 End Sub
197
198 Private Sub Form_Load()
199   Set Toolbar1.ImageList = ImageList1
200   With Toolbar1.Buttons
201     .Add , "new", "neues Objekt... ", , "new"
202     .Add , "opn", "Objekt bearbeiten", , "edt"
203     .Add , , , tbrSeparator
204     .Add , "del", "Objekt löschen ", , "del"
205   End With
206 End Sub
207
208 Private Sub Form_Resize()
209   On Error Resume Next
210
211   Dim R As Double
212   R = SSTab1.Left
213
214   SSTab1.Move R, R, ScaleWidth - 2 * R, ScaleHeight -
StatusBar1.Height - Command1.Height - R
215   Select Case SSTab1.Tab
216     Case 0
217       Picture1.Move R, Picture1.Top, SSTab1.Width - 2 * R,
SSTab1.Height - Picture1.Top - R
218       Picture1.Enabled = True
219       Text2.Enabled = False
220
221     Case 1
222       Text2.Move R, Picture1.Top, SSTab1.Width - 2 * R,
SSTab1.Height - Picture1.Top - R

```

```
223     Picture1.Enabled = False
224     Text2.Enabled = True
225
226     End Select
227
228     Command1.Move ScaleWidth - R - Command1.Width, SStab1.Top +
SStab1.Height + R / 2
229     Picture1_Resize
230
231 End Sub
232
233 Private Sub Picture1_Resize()
234     On Error Resume Next
235
236     Text1.Width = Picture1.ScaleWidth - Text1.Left
237     OLE1.Move OLE1.Left, OLE1.Top, Text1.Width,
Picture1.ScaleHeight - OLE1.Top - Toolbar1.Height * 1.5
238     Toolbar1.Move OLE1.Left, OLE1.Top + OLE1.Height, OLE1.Width
239 End Sub
240
241 Private Sub SStab1_Click(PreviousTab As Integer)
242     Form Resize
243 End Sub
244
245 Private Sub Toolbar1_ButtonClick(ByVal Button As
MSComctlLib.Button)
246     On Error Resume Next
247     Select Case Button.Key
248         Case "new"
249             If Not OLE1.Class = vbNullString Then
250                 Toolbar1_ButtonClick Toolbar1.Buttons("del")
251             End If
252             If OLE1.Class = vbNullString Then
253                 OLE1.InsertObjDlg
254             End If
255         Case "opn"
256             OLE1_DblClick
257
258         Case "del"
259             If Not OLE1.Class = vbNullString Then
260                 If MsgBox("Soll das verknüpfte Objekt gelöscht werden?",
vbQuestion Or vbYesNo, "Objekt löschen") = vbYes Then
261                     On Error Resume Next
262
263                     OLE1.Close
264                     OLE1.Delete
265                     OLE1.Class = ""
266                 End If
```

```
267     End If
268     End Select
269
270     OLE1_Updated 0
271 End Sub
272
273 Private Sub OLE1_DblClick()
274     On Error Resume Next
275
276     If OLE1.Class <> vbNullString Then
277         OLE1.DoVerb vbOLEOpen
278     End If
279
280     If Err <> 0 Then MsgBox Err.Description
281 End Sub
282
283 Private Sub OLE1_LostFocus()
284     OLE1_Updated 0
285 End Sub
286
287 Private Sub OLE1_MouseMove(Button As Integer, Shift As Integer,
X As Single, Y As Single)
288     OLE1_Updated 0
289 End Sub
290
291 Private Sub OLE1_Updated(Code As Integer)
292     On Error Resume Next
293     With Toolbar1
294         .Buttons("del").Enabled = (OLE1.Class <> vbNullString)
295         .Buttons("opn").Enabled = (OLE1.Class <> vbNullString)
296
297         'Label2 = OLE1.Class
298     End With
299 End Sub
300
```

Quellcode: ActiveX-Komponente RockClassX

Datei: RockClassX.vbp (VB Project)

```

01 Type=OleDll
02 Reference=*\\G{00020430-0000-0000-C000-
03 00000000046}#2.0#0#C:\WINNT\System32\stdole2.tlb#OLE Automation
04 Reference=*\\G{420B2830-E718-11CF-893D-
05 00A0C9054228}#1.0#0#C:\WINNT\System32\scrrun.dll#Microsoft
06 Scripting Runtime
07 Class=clsClassification; clsKlassifikation.cls
08 Class=clsLmWert; clsLmWert.cls
09 Class=clsRockClassX; clsRockClassX.cls
10 Module=basRockClassX; RockClassX.bas
11 Module=Standard; ..\..\..\..\Programmierung\BAS\Standard.bas
12 Class=clsRQDIndex; clsRQDIndex.cls
13 Class=clsZerrüttungsgradZG; clsZerrüttungsgradZG.cls
14 Class=clsKlüftigkeitsziffer; clsKlüftigkeitsziffer.cls
15 Class=clsRMR; clsRMR.cls
16 Class=clsMittlereKernlänge; clsMittlereKernlänge.cls
17 Class=clsGebirgsfaktorC; clsGebirgsfaktorC.cls
18 Class=clsSMR; clsSMR.cls
19 Class=clsRockMassQualityQ; clsRockMassQualityQ.cls
20 Class=clsRockMassQualityQTBM; clsRockMassQualityQTBM.cls
21 Form=frmParameters.frm
22 Object={5E9E78A0-531B-11CF-91F6-C2863C385E30}#1.0#0;
23 MSFLXGRD.OCX
24 UserControl=ctlParameterItem.ctl
25 Object={6FBA474E-43AC-11CE-9A0E-00AA0062BB4C}#1.0#0; SYSINFO.OCX
26 IconForm="frmParameters"
27 Startup="(None)"
28 HelpFile=""
29 Title="Gebirgsklassifikation"
30 ExeName32="RockClassX.dll"
31 Command32=""
32 Name="RockClassX"
33 HelpContextID="0"
34 Description="Gebirgsklassifikation ActiveX-DLL (Studienarbeit
35 Jörg Meier, 2001)"
36 CompatibleMode="1"
37 CompatibleEXE32="RockClassX.dll"
38 MajorVer=1
39 MinorVer=0
40 RevisionVer=16
41 AutoIncrementVer=0

```

```

37 ServerSupportFiles=0
38 VersionCompanyName="-"
39 VersionFileDescription="ActiveX-DLL der Komponente RockClassX
40 (Studienarbeit Jörg Meier, TU Bergakademie Freiberg)"
41 VersionLegalCopyright="2001, Jörg Meier"
42 VersionProductName="RockClassX"
43 CompilationType=-1
44 OptimizationType=0
45 FavorPentiumPro(tm)=0
46 CodeViewDebugInfo=0
47 NoAliasing=0
48 BoundsCheck=0
49 OverflowCheck=0
50 FlPointCheck=0
51 FDIVCheck=0
52 UnroundedFP=0
53 StartMode=1
54 Unattended=0
55 Retained=0
56 ThreadPerObject=0
57 MaxNumberOfThreads=1
58 ThreadingModel=1
59 DebugStartupOption=0
60
61 [MS Transaction Server]
62 AutoRefresh=1

```

Datei: basRockClassX.bas (VB Module)

```

01 Attribute VB_Name = "basRockClassX"
02 Option Explicit
03
04 Public Const rcxPieces As String = "Pieces"
05 Public Const rcxCLength As String = "Kernmarsch"
06 Public Const rcxCdiameter As String = "Kerndurchmesser"
07
08 Public Type PieceType
09     Length As Double
10     PreservationDegree As PreservationDegreeEnum
11 End Type
12 Public Type Pieces
13     Piece() As PieceType
14 End Type
15
16

```

```

17 Public Function PhrasePiecesExt(PiecesString As String,
Kenmarschlänge As Double, ErrSource As String) As Pieces
18 Dim L As Double
19 PhrasePiecesExt = PhrasePieces(PiecesString, L)
20
21 If L > Kenmarschlänge Then
22 'eingegebene Kernstücken länger als Kernmarsch!
23 Err.Raise vbObjectError + 3, ErrSource, "Die Summe der
eingegebenen Daten ist größer als die Berechnungslänge."
24 Exit Function
25 End If
26 End Function
27 Public Function PhrasePieces(PiecesString As String, Optional
ByRef PiecesLength As Double) As Pieces
28 'Zerlegt den String PiecesString in die einzelnen
29 'Kernbruchstücken. Liefert ein Array und die Summe
30 'aller Kernbruchstücken zurück.
31
32 Dim str As String, N As Long, a As Long
33 Dim sp As String, pl As Double, pc As PreservationDegreeEnum
34 str = PiecesString
35
36 ReDim PhrasePieces.Piece(0 To 0)
37
38 PiecesLength = 0
39 N = 1
40 Do
41 a = InStr(N, str, "/")
42 If a = 0 Then
43 sp = Mid$(str, N)
44 Else
45 sp = Mid$(str, N, a - N)
46 N = a + 1
47 End If
48 sp = Trim(sp)
49
50 If Len(sp) >= 1 Then
51 pl = Val(sp)
52 Select Case LCase(Right(sp, 1))
53 Case "a": pc = pdA
54 Case "b": pc = pdB
55 Case "c": pc = pdC
56 Case "d": pc = pdD
57 Case "e": pc = pdE
58 Case Else: pc = pdNone
59 End Select
60
61 If pl > 0 Then

```

```

62 If UBound(PhrasePieces.Piece) = 0 Then
63 ReDim PhrasePieces.Piece(1 To 1)
64 Else
65 ReDim Preserve PhrasePieces.Piece(1 To
UBound(PhrasePieces.Piece) + 1)
66 End If
67 With PhrasePieces.Piece(UBound(PhrasePieces.Piece))
68 .Length = CDbI(pl)
69 .PreservationDegree = pc
70 PiecesLength = PiecesLength + .Length
71 End With
72 End If
73 End If
74 Loop Until a = 0
75 End Function
76
77
78 Public Function GetItem(Parameter As Scripting.Dictionary, Key
As String, StandardValue As Variant, RaiseError As Boolean)
79 'Sucht nach einem Eintrag in dem Dictionary
80 'löst einen Fehler aus, falls nicht gefunden (für
RaiseError=True)
81
82 Dim K As Variant, b As Boolean
83
84 GetItem = StandardValue
85
86 With Parameter
87 b = False
88 For Each K In .Keys
89 If LCase(Trim(Key)) = LCase(Trim(K)) Then
90 b = True
91 GetItem = .Item(K)
92 Exit For
93 End If
94 Next
95 End With
96
97 If Not b Then
98 If RaiseError Then
99 Err.Raise RockClassXErr.RCXErr MissingParameter,
App.ProductName, "Parameter nicht gefunden: "" & Key & """"
100 End If
101 End If
102 End Function
103
104

```

Datei: clsKlassifikation.cls (VB Class Module)

```

01 VERSION 1.0 CLASS
02 BEGIN
03     MultiUse = -1 'True
04     Persistable = 0 'NotPersistable
05     DataBindingBehavior = 0 'vbNone
06     DataSourceBehavior = 0 'vbNone
07     MTSTransactionMode = 0 'NotAnMTSObject
08 END
09 Attribute VB Name = "clsClassification"
10 Attribute VB GlobalNameSpace = False
11 Attribute VB_Creatable = True
12 Attribute VB_PredeclaredId = False
13 Attribute VB_Exposed = True
14 Option Explicit
15
16 Public Function Calculate(Parameter As Scripting.Dictionary) As Double
17     'Berechnet den eigentlichen Zahlenwert der Klassifikation
18 End Function
19
20 Public Function ClassDesignation(ByVal Value As Double) As String
21     'Gibt die Benennung der angegebenen Klasse zurück.
22     'Value: den mit der Funktion Calculate berechneten Wert
23
24     ClassDesignation =
25     ClassDesignationByIndex(ClassIndexByValue(Value))
26 End Function
27
28 Public Function ClassDesignationByIndex(ByVal Index As Long) As String
29     'Gibt die Benennung der angegebenen Klasse zurück.
30     'Index: Index der Klasse
31 End Function
32
33 Public Function ClassIndexByValue(ByVal Value As Double) As Long
34     'Liefert den Index der Klasse, in der der Wert Value
35     'liegt.
36 End Function
37
38 Public Property Get ClassCount() As Long
39     'Liefert die Anzahl der Klassen dieser Klassifikation
40 End Property
41
42 Public Property Get ClassificationName() As String

```

```

42     'Liefert den Namen der Klassifikation
43 End Property
44
45 Public Property Get ClassificationAutorYear() As String
46     'Liefert den Autor und Jahr der Entwicklung
47 End Property
48
49 Public Property Get ValueRangeMin() As Double
50     'Liefert die Untergrenze des Wertebereiches der Klassifikation
51 End Property
52
53 Public Property Get ValueRangeMax(Berechnungslänge As Double) As Double
54     'Liefert die Obergrenze des Wertebereiches der Klassifikation
55 End Property
56
57 Public Property Get Unit() As String
58     'Liefert die Einheit der zurückgegebenen Werte
59 End Property
60
61 Public Property Get FormatString() As String
62     'Liefert das zur Formatierung des Ergebnisses
63     'nötige Zahlenformat
64 End Property
65
66 Public Property Get ParameterList() As Scripting.Dictionary
67     'Liefert ein Dictionary mit allen für die
68     'Klassifikation nötigen Schlüsseln und Beschreibungen
69 End Property
70
71 Public Function ShowParameterWindow(Parameter As String,
72     Optional StartValue As Variant, Optional ParentWindow As Object,
73     Optional TestPresence As Boolean = False) As Variant
74     'zeigt (falls gegeben) ein Fenster mit einer
75     'näheren Beschreibung und Wahlmöglichkeit
76     'der Parameter an.
77     'Falls TestPresence = True ist, wird nur TRUE oder FALSE
78     'zurückgegeben, ob ein Fenster angezeigt werden kann.
79 End Function

```

Datei: clsRockClassX.cls (VB Class Module)

```

01 VERSION 1.0 CLASS
02 BEGIN
03     MultiUse = -1 'True
04     Persistable = 0 'NotPersistable

```

```

05     DataBindingBehavior = 0 'vbNone
06     DataSourceBehavior = 0 'vbNone
07     MTSTransactionMode = 0 'NotAnMTSObject
08 END
09 Attribute VB Name = "clsRockClassX"
10 Attribute VB_GlobalNameSpace = True
11 Attribute VB_Creatable = True
12 Attribute VB_PredeclaredId = False
13 Attribute VB_Exposed = True
14 Option Explicit
15
16 'Fehlerkonstanten
17 Public Enum RockClassXErr
18     RCXErr_MissingParameter = vbObjectError + 512 + 1
19 End Enum
20
21 Public Enum PreservationDegreeEnum
22     pdNone = 0
23     pdA = 1
24     pdB = 2
25     pdC = 3
26     pdD = 4
27     pdE = 5
28 End Enum
29
30 Public Function String2Dictionary(ByVal strValue As String) As
Scripting.Dictionary
31     'wandelt eines String wie "L=100 B=200 Name="Hallo Welt, sagt
""Leo"" F=5,6"
32     'in ein Dictionary um
33
34     Dim Dict As Scripting.Dictionary
35     Set Dict = New Scripting.Dictionary
36
37     Dim s As String, t1 As String, N As Long, a As Long
38     Dim Key As String, Value As Variant
39
40     Const SetChar = "="
41
42     s = Trim(strValue)
43     Do Until s = vbNullString
44         Key = vbNullString
45
46         'nächstes SetChar finden
47         t1 = InStr(s, SetChar)
48         If t1 = 0 Then
49             'SetChar nicht gefunden

```

```

50     Value = s
51     s = vbNullString
52 Else
53     'Key heraussuchen
54     Key = Left(s, t1 - 1)
55     s = Trim(Mid(s, t1 + Len(SetChar)))
56
57     If Left(s, 1) = "" Then
58         'es folgt ein String
59         a = 0
60         For N = 2 To Len(s) - 1
61             If Mid(s, N, 1) = "" Then
62                 If Mid(s, N + 1, 1) = "" Then
63                     N = N + 1
64                 Else
65                     a = N
66                     Exit For
67                 End If
68             End If
69         Next
70         If a = 0 Then
71             If Mid(s, Len(s), 1) = "" Then
72                 a = Len(s)
73             End If
74         End If
75
76         'a = InStr(2, s, "")
77         If a = 0 Then
78             Value = Mid(s, 2)
79             s = vbNullString
80         Else
81             If a < 2 Then
82                 Value = vbNullString
83             Else
84                 Value = Mid(s, 2, a - 2)
85             End If
86             s = Mid(s, a + 1)
87         End If
88         Value = Replace(Value, "", "")
89     Else
90         'es folgt irgendwas
91
92         'nächstes SetChar finden
93         t1 = InStr(s, SetChar)
94         If t1 = 0 Then
95             'Reststring gehört zu Key
96             Value = s
97             s = vbNullString

```

```

98     Else
99         'na wo trennen wir denn...
100        a = InStrRev(Left(s, t1), " ") 'letztes Leerzeichen
vor SetChar
101        If a = 0 Then
102            'Ups!
103            For N = 1 To t1 - 1
104                If Not IsNumeric(Mid(s, N, 1)) Then
105                    a = N
106                    Exit For
107                End If
108            Next
109        End If
110        If a <> 0 Then
111            'beim Leerzeichen
112            Value = Left(s, a - 1)
113            s = Mid(s, a)
114        Else
115            s = Mid(s, t1 + 1)
116        End If
117    End If
118 End If
119
120    'unnötige Leerzeichen entfernen
121    s = Trim(s)
122    Key = Trim(Key)
123
124    'hinzufügen
125    If Key <> vbNullString And Value <> vbNullString Then 'nur
sinnvolle Einträge zulassen!
126        If Not IsNumeric(Left(Key, 1)) Then 'Keys dürfen mit
keiner Zahl anfangen!
127            Dict.Add Key, Value 'hinzufügen
128        End If
129    End If
130
131 End If
132 Loop
133
134 Set String2Dictionary = Dict
135 End Function
136
137 Public Function Dictionary2String(Dict As Scripting.Dictionary)
As String
138     Dim K As Variant, str As String, Value As Variant
139
140     str = vbNullString
141     For Each K In Dict.Keys

```

```

142     If Not str = vbNullString Then
143         str = str & " "
144     End If
145
146     Value = Dict.Item(K)
147     If InStr(Value, "\"") <> 0 Or InStr(Value, "=") <> 0 Then
148         'falls Gleichheitszeichen oder Anführungszeichen
vorkommen...
149         Value = "\"" & Replace(Value, "\"", "\"\"") & "\""
150     End If
151
152     str = str & K & "=" & Value
153 Next
154
155 Dictionary2String = str
156 End Function
157
158 Public Function AddPieceLength(PiecesString As String,
AddPreservationDegreeA As Boolean, AddPreservationDegreeB As
Boolean, AddPreservationDegreeC As Boolean,
AddPreservationDegreeD As Boolean, AddPreservationDegreeE As
Boolean, AddPreservationDegreeUnknown As Boolean)
159     Dim Pieces As Pieces
160     Pieces = PhrasePieces(PiecesString)
161
162     Dim N As Long, L As Double
163     For N = 1 To UBound(Pieces.Piece)
164         With Pieces.Piece(N)
165             Select Case .PreservationDegree
166                 Case pdA: If AddPreservationDegreeA Then L = L + .Length
167                 Case pdB: If AddPreservationDegreeB Then L = L + .Length
168                 Case pdC: If AddPreservationDegreeC Then L = L + .Length
169                 Case pdD: If AddPreservationDegreeD Then L = L + .Length
170                 Case pdE: If AddPreservationDegreeE Then L = L + .Length
171                 Case Else: If AddPreservationDegreeUnknown Then L = L +
.Length
172             End Select
173         End With
174     Next
175
176     AddPieceLength = L
177 End Function
178
179 Public Property Get ClassificationList() As Collection
180     'liefert eine List aller in dem Projekt
181     'RockClassX eingebundenen Klassifikationen
182
183     Set ClassificationList = New Collection

```

```

184 With ClassificationList
185     .Add New clsGebirgsfaktorC
186     .Add New clsKlüftigkeitsziffer
187     .Add New clsLmWert
188     .Add New clsMittlereKernlänge
189     .Add New clsRMR
190     .Add New clsRockMassQualityQ
191     .Add New clsRockMassQualityQTBM
192     .Add New clsRQDIndex
193     .Add New clsSMR
194     .Add New clsZerrüttungsgradZG
195 End With
196 End Property
197

```

Datei: clsGebirgsfaktorC.cls (VB Class Module)

```

01 VERSION 1.0 CLASS
02 BEGIN
03     MultiUse = -1 'True
04     Persistable = 0 'NotPersistable
05     DataBindingBehavior = 0 'vbNone
06     DataSourceBehavior = 0 'vbNone
07     MTSTransactionMode = 0 'NotAnMTSObject
08 END
09 Attribute VB_Name = "clsGebirgsfaktorC"
10 Attribute VB_GlobalNameSpace = False
11 Attribute VB_Creatable = True
12 Attribute VB_PredeclaredId = False
13 Attribute VB_Exposed = True
14 Option Explicit
15 Implements clsClassification
16
17 Private Const rcxGFCPieces As String = rcxPieces
18 Private Const rcxGFCCLength As String = rcxCLength
19 Private Const rcxGFCCDiameter As String = rcxCDiameter
20 Private Const rcxGFCCSampleCount As String = "AnzahlProben"
21 Private Const rcxGFCCSampleHeight As String = "HöheProbe"
22 Private Const rcxGFCCSampleCountAlt As String = "p"
23 Private Const rcxGFCCSampleHeightAlt As String = "H"
24
25 Private Sub Class_Initialize()
26     Set clsClassification = New clsClassification
27 End Sub
28
29 Private Function clsClassification Calculate(Parameter As
Scripting.Dictionary) As Double

```

```

30 Dim Pieces As String, kml As Double
31 Dim p As Long, H As Double, D As Double
32 Pieces = GetItem(Parameter, rcxGFCPieces, vbNullString, True)
33 kml = Val(GetItem(Parameter, rcxGFCCLength, vbNullString,
True))
34
35 Dim v As Variant
36
37 v = GetItem(Parameter, rcxGFCCSampleCount, "leer", False)
38 If v = "leer" Then
39     v = GetItem(Parameter, rcxGFCCSampleCountAlt, vbNullString,
True)
40 End If
41 p = Val(v)
42
43 v = GetItem(Parameter, rcxGFCCSampleHeight, "leer", False)
44 If v = "leer" Then
45     v = GetItem(Parameter, rcxGFCCSampleHeightAlt, vbNullString,
True)
46 End If
47 H = Val(v)
48
49 D = GetItem(Parameter, rcxGFCCDiameter, 10, False)
50
51 clsClassification_Calculate =
prvGebirgsfaktorC(PhrasePiecesExt(Pieces, kml,
clsClassification ClassificationName), kml, p, H, D)
52 End Function
53
54 Private Property Get clsClassification_ClassCount() As Long
55     clsClassification_ClassCount = 5
56 End Property
57
58 Private Function clsClassification ClassDesignation(ByVal Value
As Double) As String
59     clsClassification_ClassDesignation =
clsClassification_ClassDesignationByIndex(clsClassification_Clas
sIndexByValue(Value))
60 End Function
61
62 Private Property Get clsClassification ClassificationAutorYear()
As String
63     clsClassification_ClassificationAutorYear = "Hansagi (1974)"
64 End Property
65
66 Private Property Get clsClassification ClassificationName() As
String
67     clsClassification_ClassificationName = "Gebirgsfaktor C"

```

```

68 End Property
69
70 Private Function clsClassification ClassIndexByValue(ByVal Value
As Double) As Long
71     Dim ValueBack As Long
72
73     Select Case Value
74         Case Is <= 0.15: ValueBack = 1
75         Case Is <= 0.3:  ValueBack = 2
76         Case Is <= 0.45: ValueBack = 3
77         Case Is <= 0.65: ValueBack = 4
78         Case Else:      ValueBack = 5
79     End Select
80     clsClassification_ClassIndexByValue = ValueBack
81 End Function
82 Private Function clsClassification ClassDesignationByIndex(ByVal
Index As Long) As String
83     Dim Value As String
84
85     Select Case Index
86         Case 1: Value = "sehr schwach"
87         Case 2: Value = "schwach"
88         Case 3: Value = "mittel"
89         Case 4: Value = "gut"
90         Case 5: Value = "ausgezeichnet"
91     End Select
92     clsClassification ClassDesignationByIndex = Value
93 End Function
94
95
96 Private Property Get clsClassification_FormatString() As String
97     clsClassification FormatString = "0.00"
98 End Property
99
100 Private Property Get clsClassification Unit() As String
101     clsClassification_Unit = ""
102 End Property
103
104 Private Property Get
clsClassification ValueRangeMax(Berechnungslänge As Double) As
Double
105     clsClassification_ValueRangeMax = 1
106 End Property
107
108 Private Property Get clsClassification ValueRangeMin() As Double
109     clsClassification ValueRangeMin = 0
110 End Property
111

```

```

112 Private Function prvGebirgsfaktorC(Pieces As Pieces, kml As
Double, p As Long, H As Double, D As Double) As Double
113     Dim N As Long, cK As Double, cN As Long 'Hilfsvariablen
114
115     cN = 0 'Null setzten (ist
eigentlich nicht nötig, sicher ist sicher)
116     For N = 1 To UBound(Pieces.Piece) 'kein Stück vergessen...
117         With Pieces.Piece(N) 'dann gehts ein wenig
schneller
118             If .Length > D Then 'min. Bruchstücklänge
119                 Select Case .PreservationDegree
120                     Case pdNone, pdA, pdB 'nur Erhaltungsgrad A, B &
NONE
121                         cK = cK + .Length
122                         cN = cN + 1
123                     Case Else 'falscher Erhaltungsgrad
124                         'nix
125                     End Select
126                 End If
127             End With
128         Next
129         If cN = 0 Then
130             prvGebirgsfaktorC = 0
131         Else
132             prvGebirgsfaktorC = 1 / (2 * kml) * (p * H + cK / cN)
'eigentliche Berechnung & zurückgeben
133         End If
134     End Function
135
136
137 Public Function GebirgsfaktorC(Pieces As String, CoreLength As
Double, SampleCount As Long, SampleHeight As Double, Optional
CoreDiameter As Double = 10)
138     Dim Dict As Scripting.Dictionary
139
140     Set Dict = New Scripting.Dictionary
141     Dict.Add rcxGFCCPieces, Pieces
142     Dict.Add rcxGFCCLength, CoreLength
143     Dict.Add rcxGFCCSampleCount, SampleCount
144     Dict.Add rcxGFCCSampleHeight, SampleHeight
145     Dict.Add rcxGFCCDiameter, CoreDiameter
146
147     GebirgsfaktorC = clsClassification_Calculate(Dict)
148 End Function
149 Public Property Get ClassificationInterface() As
clsClassification
150     Set ClassificationInterface = Me
151 End Property

```

```

152
153 Public Function SampleCountFromPieces(PiecesString As String,
SampleHeight As Double) As Long
154     Dim Pieces As Pieces
155     Pieces = PhrasePieces(PiecesString)
156
157     Dim N As Long, Count As Long
158     Count = 0
159     For N = 1 To UBound(Pieces.Piece)
160         With Pieces.Piece(N)
161             Select Case .PreservationDegree
162                 Case pdNone, pdA, pdB      'nur Erhaltungsgrad A, B &
NONE
163                     Count = Count + Fix(.Length / SampleHeight)
164             End Select
165         End With
166     Next
167
168     SampleCountFromPieces = Count
169 End Function
170
171 Private Property Get clsClassification ParameterList() As
Scripting.IDictionary
172     Dim RQD As clsRQDIndex
173     Set RQD = New clsRQDIndex
174     Set clsClassification ParameterList =
RQD.ClassificationInterface.ParameterList
175     With clsClassification ParameterList
176         .Add rcxGFCCSampleCount, "Anzahl zylindrischer Proben, die
aus intakten Bohrkernen herausgesägt werden können"
177         .Add rcxGFCCSampleHeight, "Höhe der zylindrischen
Probenkörper [m]"
178
179         .Add rcxGFCCSampleCountAlt, "Anzahl zylindrischer Proben,
die aus intakten Bohrkernen herausgesägt werden können"
180         .Add rcxGFCCSampleHeightAlt, "Höhe der zylindrischen
Probenkörper [m]"
181     End With
182 End Property
183
184 Private Function clsClassification ShowParameterWindow(Parameter
As String, Optional StartValue As Variant, Optional ParentWindow
As Object, Optional TestPresence As Boolean = False) As Variant
185     If TestPresence Then
186         clsClassification ShowParameterWindow = False
187     Else
188         clsClassification ShowParameterWindow = StartValue
189     End If

```

```

190 End Function
191
192

```

Datei: clsKlüftigkeitsziffer.cls (VB Class Module)

```

01 VERSION 1.0 CLASS
02 BEGIN
03     MultiUse = -1 'True
04     Persistable = 0 'NotPersistable
05     DataBindingBehavior = 0 'vbNone
06     DataSourceBehavior = 0 'vbNone
07     MTSTransactionMode = 0 'NotAnMTSObject
08 END
09 Attribute VB_Name = "clsKlüftigkeitsziffer"
10 Attribute VB_GlobalNameSpace = False
11 Attribute VB_Creatable = True
12 Attribute VB_PredeclaredId = False
13 Attribute VB_Exposed = True
14 Option Explicit
15 Implements clsClassification
16
17 Private Const rcxKKPieces As String = rcxPieces
18 Private Const rcxKKCLength As String = rcxCLength
19
20 Private Sub Class_Initialize()
21     Set clsClassification = New clsClassification
22 End Sub
23
24 Private Function clsClassification Calculate(Parameter As
Scripting.Dictionary) As Double
25     Dim Pieces As String, kml As Double
26     Pieces = GetItem(Parameter, rcxKKPieces, vbNullString, True)
27     kml = Val(GetItem(Parameter, rcxKKCLength, vbNullString,
True))
28
29     clsClassification_Calculate =
prvKlüftigkeitsziffer(PhrasePiecesExt(Pieces, kml,
clsClassification_ClassificationName), kml)
30 End Function
31
32 Private Property Get clsClassification ClassCount() As Long
33     clsClassification ClassCount = 5
34 End Property
35
36 Private Function clsClassification ClassDesignation(ByVal Value
As Double) As String

```

```
37     clsClassification_ClassDesignation =  
clsClassification_ClassDesignationByIndex(clsClassification_Clas  
sIndexByValue(Value))  
38 End Function  
39  
40 Private Property Get clsClassification_ClassificationAutorYear()  
As String  
41     clsClassification_ClassificationAutorYear = "Müller (1963,  
Möbus (1982))"  
42 End Property  
43  
44 Private Property Get clsClassification_ClassificationName() As  
String  
45     clsClassification_ClassificationName = "Klüftigkeitsziffer k"  
46 End Property  
47  
48 Private Function clsClassification_ClassIndexByValue(ByVal Value  
As Double) As Long  
49     Dim ValueBack As Long  
50  
51     Select Case Value  
52         Case Is <= 1:     ValueBack = 1  
53         Case Is <= 10:   ValueBack = 2  
54         Case Is <= 100: ValueBack = 3  
55         Case Else:      ValueBack = 4  
56     End Select  
57     clsClassification_ClassIndexByValue = ValueBack  
58 End Function  
59 Private Function clsClassification_ClassDesignationByIndex(ByVal  
Index As Long) As String  
60     Dim Value As String  
61  
62     Select Case Index  
63         Case 1: Value = "weitmaschig"  
64         Case 2: Value = "mitteldicht"  
65         Case 3: Value = "engständig"  
66         Case 4: Value = "zerdrückt bis mylonisiert"  
67     End Select  
68     clsClassification_ClassDesignationByIndex = Value  
69 End Function  
70  
71 Private Property Get clsClassification_FormatString() As String  
72     clsClassification_FormatString = "0.0"  
73 End Property  
74  
75 Private Property Get clsClassification_Unit() As String  
76     clsClassification_Unit = "1/m"  
77 End Property
```

```
78  
79 Private Property Get  
clsClassification_ValueRangeMax(Berechnungslänge As Double) As  
Double  
80     clsClassification_ValueRangeMax = 10  
81 End Property  
82  
83 Private Property Get clsClassification_ValueRangeMin() As Double  
84     clsClassification_ValueRangeMin = 0.1  
85 End Property  
86  
87 Private Function prvKlüftigkeitsziffer(Pieces As Pieces, kml As  
Double) As Double  
88     Dim N As Long, v As Double           'Hilfsvariablen  
89  
90     v = 0                                'Null setzten (ist  
eigentlich nicht nötig, sicher ist sicher)  
91     For N = 1 To UBound(Pieces.Piece) 'kein Stück vergessen...  
92         With Pieces.Piece(N)           'dann gehts ein wenig  
schneller  
93             Select Case .PreservationDegree  
94                 Case pdNone, pdA, pdB 'nur Erhaltungsgrad A, B &  
NONE  
95                     v = v + 1  
96                 Case Else              'falscher Erhaltungsgrad  
97                     'nix  
98             End Select  
99         End With  
100     Next  
101     prvKlüftigkeitsziffer = v / kml * 100           'eigentliche  
Berechnung & zurückgeben  
102 End Function  
103  
104 Public Function Klüftigkeitsziffer(Pieces As String, CoreLength  
As Double)  
105     Dim Dict As Scripting.Dictionary  
106  
107     Set Dict = New Scripting.Dictionary  
108     Dict.Add rcxKKPieces, Pieces  
109     Dict.Add rcxKKLength, CoreLength  
110  
111     Klüftigkeitsziffer = clsClassification_Calculate(Dict)  
112 End Function  
113 Public Property Get ClassificationInterface() As  
clsClassification  
114     Set ClassificationInterface = Me  
115 End Property
```

```

116
117 Private Property Get clsClassification_ParameterList() As
Scripting.IDictionary
    Dim RQD As clsRQDIndex
118     Set RQD = New clsRQDIndex
119     Set clsClassification_ParameterList =
120     RQD.ClassificationInterface.ParameterList
121 End Property
122
123 Private Function clsClassification ShowParameterWindow(Parameter
As String, Optional StartValue As Variant, Optional ParentWindow
As Object, Optional TestPresence As Boolean = False) As Variant
124     If TestPresence Then
125         clsClassification_ShowParameterWindow = False
126     Else
127         clsClassification ShowParameterWindow = StartValue
128     End If
129 End Function
130
131

```

Datei: clsLmWert.cls (VB Class Module)

```

01 VERSION 1.0 CLASS
02 BEGIN
03     MultiUse = -1 'True
04     Persistable = 0 'NotPersistable
05     DataBindingBehavior = 0 'vbNone
06     DataSourceBehavior = 0 'vbNone
07     MTSTransactionMode = 0 'NotAnMTSObject
08 END
09 Attribute VB_Name = "clsLmWert"
10 Attribute VB_GlobalNameSpace = False
11 Attribute VB_Creatable = True
12 Attribute VB_PredeclaredId = False
13 Attribute VB_Exposed = True
14 Option Explicit
15 Implements clsClassification
16
17 Private Const rcxLmPieces As String = rcxPieces
18 Private Const rcxLmCLength As String = rcxCLength
19 Private Const rcxLmClass As String = "klasse"
20
21
22 Private Sub Class_Initialize()
23     Set clsClassification = New clsClassification
24 End Sub

```

```

25
26 Private Function clsClassification_Calculate(Parameter As
Scripting.Dictionary) As Double
27     Dim Pieces As String, kml As Double, klasse As Double
28     Pieces = GetItem(Parameter, rcxLmPieces, vbNullString, True)
29     kml = Val(GetItem(Parameter, rcxLmCLength, vbNullString,
True))
30     klasse = Val(GetItem(Parameter, rcxLmClass, 5, False))
31
32     clsClassification Calculate =
prvLmWert(PhrasePiecesExt(Pieces, kml,
clsClassification ClassificationName), kml, klasse)
33 End Function
34
35 Private Property Get clsClassification ClassCount() As Long
36     clsClassification ClassCount = 4
37 End Property
38
39 Private Function clsClassification_ClassDesignation(ByVal Value
As Double) As String
40     clsClassification ClassDesignation =
clsClassification ClassDesignationByIndex(clsClassification Clas
sIndexByValue(Value))
41 End Function
42
43 Private Function clsClassification ClassDesignationByIndex(ByVal
Index As Long) As String
44     Dim Value As String
45
46     Select Case Index
47         Case 1: Value = "sehr gering"           'mürbe, teils
zersetzt, sehr stark verwittert und aufgelockert, sehr
kleinklüftig"
48         Case 2: Value = "gering - mittel"       'stark geklüftet und
aufgelockert, teils mürbe, angewittert, lagig"
49         Case 3: Value = "mittel - gut"         'teils kompakt,
frisch, quarzreich, auf Klüften wenig angewittert, gering
entfestigt"
50         Case 4: Value = "gut - sehr gut"       'kompakt, fest bis
sehr fest, frisch, quarzreich (Beton B25 > 17,5 cm)"
51     End Select
52     clsClassification_ClassDesignationByIndex = Value
53 End Function
54
55 Private Property Get clsClassification ClassificationAutorYear()
As String
56     clsClassification ClassificationAutorYear = "Meier, G. (1974)"
57 End Property

```

```

58
59 Private Property Get clsClassification_ClassificationName() As
String
60     clsClassification_ClassificationName = "Lm-Wert"
61 End Property
62
63 Private Function clsClassification_ClassIndexByValue(ByVal Value
As Double) As Long
64     Dim ValueBack As Long
65
66     Select Case Value
67         Case Is < 5: ValueBack = 1
68         Case Is <= 10: ValueBack = 2
69         Case Is <= 15: ValueBack = 3
70         Case Else: ValueBack = 4
71     End Select
72     clsClassification_ClassIndexByValue = ValueBack
73 End Function
74
75 Private Property Get clsClassification_FormatString() As String
76     clsClassification_FormatString = "0.0"
77 End Property
78
79 Private Property Get clsClassification_ParameterList() As
Scripting.IDictionary
80     Dim RQD As clsRQDIndex
81     Set RQD = New clsRQDIndex
82     Set clsClassification_ParameterList =
RQD.ClassificationInterface.ParameterList
83 End Property
84
85 Private Function clsClassification_ShowParameterWindow(Parameter
As String, Optional StartValue As Variant, Optional ParentWindow
As Object, Optional TestPresence As Boolean = False) As Variant
86     If TestPresence Then
87         clsClassification_ShowParameterWindow = False
88     Else
89         clsClassification_ShowParameterWindow = StartValue
90     End If
91 End Function
92
93 Private Property Get clsClassification_Unit() As String
94     clsClassification_Unit = "cm"
95 End Property
96

```

```

97 Private Property Get
clsClassification_ValueRangeMax(Berechnungslänge As Double) As
Double
98     clsClassification_ValueRangeMax = Berechnungslänge
99 End Property
100
101 Private Property Get clsClassification_ValueRangeMin() As Double
102     clsClassification_ValueRangeMin = 0
103 End Property
104
105 Private Function prvLmWert(Pieces As Pieces, ByVal kml As
Double, ByVal ClassLength As Double) As Double
106     'Berechnet den LmWert
107     Dim N As Long, v As Double           'Hilfsvariablen
108     Dim L As Double
109
110     v = 0                               'Null setzen (ist
eigentlich nicht nötig, sicher ist sicher)
111     For N = 1 To UBound(Pieces.Piece) 'kein Stück vergessen...
112         With Pieces.Piece(N)           'dann gehts ein wenig
schneller
113             Select Case .PreservationDegree
114                 Case pdNone, pdA, pdB     'nur Erhaltungsgrad A, B &
NONE
115                     v = v + .Length * PieceClass(.Length, ClassLength,
kml)
116                     L = L + .Length
117                 Case Else                 'falscher Erhaltungsgrad
118                     'nix
119             End Select
120         End With
121     Next
122     prvLmWert = ((kml - L) * ClassLength / 2 + v) / kml
123     'eigentliche Berechnung & zurückgeben
124 End Function
125
126 Public Function LmWert(Pieces As String, CoreLength As Double,
Optional ClassLength As Double = 5)
127     Dim Dict As Scripting.Dictionary
128
129     Set Dict = New Scripting.Dictionary
130     Dict.Add rcxLmPieces, Pieces
131     Dict.Add rcxLmCLength, CoreLength
132     Dict.Add rcxLmClass, ClassLength
133
134     LmWert = clsClassification_Calculate(Dict)
135 End Function

```

```

135 Public Property Get ClassificationInterface() As
    clsClassification
136     Set ClassificationInterface = Me
137 End Property
138
139 Public Function PieceClass(ByVal PieceLenght As Double, ByVal
    ClassLenght As Double, ByVal CoreLength As Double) As Double
140 Attribute PieceClass.VB_MemberFlags = "40"
141 'Berechnet für eine gegebene Kernbruchstücklänge
142 'und Klassengröße die resultierende Klasse
143
144 Dim MaxPieceClass As Double, X As Double
145
146 If ClassLenght = 0 Then
147     PieceClass = PieceLenght
148 Else
149     MaxPieceClass = (Fix(CoreLength / ClassLenght + 0.5) - 0.5)
    * ClassLenght
150     X = ClassLenght * (Fix((PieceLenght) / ClassLenght) + 0.5 +
    ((PieceLenght / ClassLenght) = Fix(PieceLenght / ClassLenght)))
151
152     If MaxPieceClass < X Then
153         PieceClass = MaxPieceClass
154     Else
155         PieceClass = X
156     End If
157 End If
158 End Function
159
160

```

Datei: clsMittlereKernlänge.cls (VB Class Module)

```

01 VERSION 1.0 CLASS
02 BEGIN
03     MultiUse = -1 'True
04     Persistable = 0 'NotPersistable
05     DataBindingBehavior = 0 'vbNone
06     DataSourceBehavior = 0 'vbNone
07     MTSTransactionMode = 0 'NotAnMTSObject
08 END
09 Attribute VB Name = "clsMittlereKernlänge"
10 Attribute VB GlobalNameSpace = False
11 Attribute VB_Creatable = True
12 Attribute VB_PredeclaredId = False
13 Attribute VB Exposed = True
14 Option Explicit

```

```

15 Implements clsClassification
16
17 Private Const rcxMKKPieces As String = rcxPieces
18 Private Const rcxMKKCLength As String = rcxCLength
19 Private Const rcxMKKClass As String = "klasse"
20
21 Private Sub Class_Initialize()
22     Set clsClassification = New clsClassification
23 End Sub
24
25 Private Function clsClassification Calculate(Parameter As
    Scripting.Dictionary) As Double
26     Dim Pieces As String, kml As Double, klasse As Double
27     Pieces = GetItem(Parameter, rcxMKKPieces, vbNullString, True)
28     kml = Val(GetItem(Parameter, rcxMKKCLength, vbNullString,
    True))
29
30     clsClassification Calculate = prvMKK(PhrasePiecesExt(Pieces,
    kml, clsClassification_ClassificationName), kml)
31 End Function
32
33 Private Property Get clsClassification ClassCount() As Long
34 End Property
35
36 Private Function clsClassification_ClassDesignation(ByVal Value
    As Double) As String
37     clsClassification ClassDesignation =
    clsClassification ClassDesignationByIndex(clsClassification Clas
    sIndexByValue(Value))
38 End Function
39
40 Private Function clsClassification ClassDesignationByIndex(ByVal
    Index As Long) As String
41     Dim Value As String
42
43     ' Select Case Index
44     ' Case 1: Value = "sehr gering"
45     ' Case 2: Value = "gering - mittel"
46     ' Case 3: Value = "mittel - gut"
47     ' Case 4: Value = "gut - sehr gut"
48     ' End Select
49     clsClassification_ClassDesignationByIndex = Value
50 End Function
51
52 Private Property Get clsClassification ClassificationAutorYear()
    As String
53     clsClassification ClassificationAutorYear = "Hansagi (1967)"
54 End Property

```

```

55
56 Private Property Get clsClassification_ClassificationName() As
String
57     clsClassification_ClassificationName = "mittlere Kernlänge k"
58 End Property
59
60 Private Function clsClassification_ClassIndexByValue(ByVal Value
As Double) As Long
61     Dim ValueBack As Long
62
63     ' Select Case Value
64     '     Case Is < 5: ValueBack = 1
65     '     Case Is <= 10: ValueBack = 2
66     '     Case Is <= 15: ValueBack = 3
67     '     Case Else: ValueBack = 4
68     ' End Select
69     clsClassification_ClassIndexByValue = ValueBack
70 End Function
71
72 Private Property Get clsClassification_FormatString() As String
73     clsClassification_FormatString = "0.0"
74 End Property
75
76 Private Property Get clsClassification_Unit() As String
77     clsClassification_Unit = "cm"
78 End Property
79
80 Private Property Get
81     clsClassification_ValueRangeMax(Berechnungslänge As Double) As
Double
82     clsClassification_ValueRangeMax = Berechnungslänge
83 End Property
84
85 Private Property Get clsClassification_ValueRangeMin() As Double
86     clsClassification_ValueRangeMin = 0
87 End Property
88
89 Private Function prvMKK(Pieces As Pieces, ByVal kml As Double)
As Double
90     'Berechnet den LmWert
91     Dim N As Long, v As Double 'Hilfsvariablen
92     Dim ClassLength As Double, L As Double
93     ClassLength = 10
94
95     v = 0: L = 0 'Null setzten (ist
eigentlich nicht nötig, sicher ist sicher)
96     For N = 1 To UBound(Pieces.Piece) 'kein Stück vergessen...

```

```

96     With Pieces.Piece(N) 'dann gehts ein wenig
schneller
97         If .Length > 10 Then 'min. Bruchstücklänge
98             Select Case .PreservationDegree
99                 Case pdNone, pdA, pdB 'nur Erhaltungsgrad A, B &
NONE
100                 v = v + .Length * PieceClass(.Length, ClassLength,
kml)
101                 L = L + .Length
102             Case Else 'falscher Erhaltungsgrad
103                 'nix
104             End Select
105         End If
106     End With
107 Next
108
109 If L = 0 Then
110     prvMKK = 0
111 Else
112     prvMKK = v / L 'eigentliche Berechnung & zurückgeben
113 End If
114 End Function
115
116 Public Function MittlereKernlänge(Pieces As String, CoreLength
As Double)
117     Dim Dict As Scripting.Dictionary
118
119     Set Dict = New Scripting.Dictionary
120     Dict.Add rcxMKKPieces, Pieces
121     Dict.Add rcxMKKLength, CoreLength
122
123     MittlereKernlänge = clsClassification Calculate(Dict)
124 End Function
125 Public Property Get ClassificationInterface() As
clsClassification
126     Set ClassificationInterface = Me
127 End Property
128
129 Public Function PieceClass(ByVal PieceLenght As Double, ByVal
ClassLength As Double, ByVal CoreLength As Double) As Double
130     'Berechnet für eine gegebene Kernbruchstücklänge
131     'und Klassengröße die resultierende Klasse
132
133     Dim MaxPieceClass As Double, X As Double
134
135     If ClassLength = 0 Then
136         PieceClass = PieceLenght
137     Else

```

```

138     MaxPieceClass = (Fix(CoreLength / ClassLength + 0.5) - 0.5)
* ClassLength
139     X = ClassLength * (Fix((PieceLength) / ClassLength) + 0.5 +
((PieceLength / ClassLength) = Fix(PieceLength / ClassLength)))
140
141     If MaxPieceClass < X Then
142         PieceClass = MaxPieceClass
143     Else
144         PieceClass = X
145     End If
146 End If
147 End Function
148
149 Private Property Get clsClassification_ParameterList() As
Scripting.IDictionary
150     Dim RQD As clsRQDIndex
151     Set RQD = New clsRQDIndex
152     Set clsClassification_ParameterList =
RQD.ClassificationInterface.ParameterList
153 End Property
154
155 Private Function clsClassification_ShowParameterWindow(Parameter
As String, Optional StartValue As Variant, Optional ParentWindow
As Object, Optional TestPresence As Boolean = False) As Variant
156     If TestPresence Then
157         clsClassification_ShowParameterWindow = False
158     Else
159         clsClassification_ShowParameterWindow = StartValue
160     End If
161 End Function
162
163

```

Datei: clsRMR.cls (VB Class Module)

```

01 VERSION 1.0 CLASS
02 BEGIN
03     MultiUse = -1 'True
04     Persistable = 0 'NotPersistable
05     DataBindingBehavior = 0 'vbNone
06     DataSourceBehavior = 0 'vbNone
07     MTSTransactionMode = 0 'NotAnMTSObject
08 END
09 Attribute VB_Name = "clsRMR"
10 Attribute VB_GlobalNameSpace = False
11 Attribute VB_Creatable = True
12 Attribute VB_PredeclaredId = False

```

```

13 Attribute VB_Exposed = True
14 Option Explicit
15 Implements clsClassification
16
17 Private Const rcxRMRPieces As String = rcxPieces
18 Private Const rcxRMRCLength As String = rcxCLength
19 Private Const rcxRMRRQu As String = "RQu"
20 Private Const rcxRMRQu As String = "Qu"
21 Private Const rcxRMRRa As String = "Ra"
22 Private Const rcxRMRRb As String = "Rb"
23 Private Const rcxRMRRc As String = "Rc"
24 Private Const rcxRMRRd As String = "Rd"
25 Private Const rcxRMRKluftabstand As String = "Kluftabstand"
26
27 Private clsRQDIndex As clsRQDIndex
28
29 Private Sub Class_Initialize()
30     Set clsClassification = New clsClassification
31     Set clsRQDIndex = New clsRQDIndex
32 End Sub
33
34 Private Sub Class_Terminate()
35     Set clsRQDIndex = Nothing
36 End Sub
37
38 Private Function clsClassification_Calculate(Parameter As
Scripting.Dictionary) As Double
39     Dim Pieces As String, kml As Double, v As Variant
40     Dim RQu As Double, Ra As Double, Rb As Double, Rc As Double,
Rd As Double
41
42     Pieces = GetItem(Parameter, rcxRMRPieces, vbNullString, True)
43     kml = Val(GetItem(Parameter, rcxRMRCLength, vbNullString,
True))
44
45     v = GetItem(Parameter, rcxRMRRQu, "leer", False)
46     If v = "leer" Then
47         v = GetItem(Parameter, rcxRMRQu, vbNullString, True)
48     End If
49     RQu = RatingQu(Cdbl(v))
50
51     v = Val(GetItem(Parameter, rcxRMRRa, "Leer", False))
52     If v = "Leer" Then
53         v = Val(GetItem(Parameter, rcxRMRKluftabstand, "Leer",
False))
54     End If
55     'aus Pieces
56     Dim clsKlüftigkeitsziffer As clsKlüftigkeitsziffer

```

```
57     Set clsKlüftigkeitsziffer = New clsKlüftigkeitsziffer
58     Ra = RatingRa(1000 /
clsKlüftigkeitsziffer.Klüftigkeitsziffer(Pieces, kml))
59     Else
60     Ra = RatingRa(Val(v)) 'über Kluftabstand (in [mm])
61     End If
62     Else
63     Ra = Val(v) 'explizit angegeben
64     End If
65
66     Rb = Val(GetItem(Parameter, rcxRMRRb, vbNullString, True))
67
68     Rc = Val(GetItem(Parameter, rcxRMRRc, vbNullString, True))
69
70     Rd = Val(GetItem(Parameter, rcxRMRRd, vbNullString, True))
71
72     clsClassification Calculate = prvRMR(PhrasePiecesExt(Pieces,
kml, clsClassification ClassificationName), Pieces, kml, RQu,
Ra, Rb, Rc, Rd)
73     End Function
74
75     Private Property Get clsClassification ClassCount() As Long
76     clsClassification ClassCount = 5
77     End Property
78
79     Private Function clsClassification ClassDesignation(ByVal Value
As Double) As String
80     clsClassification ClassDesignation =
clsClassification ClassDesignationByIndex(clsClassification Clas
sIndexByValue(Value))
81     End Function
82
83     Private Property Get clsClassification ClassificationAutorYear()
As String
84     clsClassification ClassificationAutorYear = "Bieniawski
(1973)"
85     End Property
86
87     Private Property Get clsClassification ClassificationName() As
String
88     clsClassification ClassificationName = "RMR-System"
89     End Property
90
91     Private Function clsClassification ClassIndexByValue(ByVal Value
As Double) As Long
92     Dim ValueBack As Long
93
```

```
94     Select Case Value
95     Case Is > 80: ValueBack = 1
96     Case Is > 60: ValueBack = 2
97     Case Is > 40: ValueBack = 3
98     Case Is > 20: ValueBack = 4
99     Case Else: ValueBack = 5
100    End Select
101    clsClassification_ClassIndexByValue = ValueBack
102    End Function
103    Private Function clsClassification ClassDesignationByIndex(ByVal
Index As Long) As String
104    Dim Value As String
105
106    Select Case Index
107    Case 1: Value = "sehr gut"
108    Case 2: Value = "gut"
109    Case 3: Value = "normal"
110    Case 4: Value = "schlecht"
111    Case 5: Value = "sehr schlecht"
112    End Select
113    clsClassification ClassDesignationByIndex = Value
114    End Function
115
116    Private Property Get clsClassification_FormatString() As String
117    clsClassification_FormatString = "0"
118    End Property
119
120    Private Property Get clsClassification Unit() As String
121    clsClassification Unit = ""
122    End Property
123
124    Private Property Get
clsClassification ValueRangeMax(Berechnungslänge As Double) As
Double
125    clsClassification ValueRangeMax = 100
126    End Property
127
128    Private Property Get clsClassification ValueRangeMin() As Double
129    clsClassification ValueRangeMin = 0
130    End Property
131
132    Private Function prvRMR(Pieces As Pieces, PiecesString As
String, kml As Double, RQu, Ra As Double, Rb As Double, Rc As
Double, Rd As Double) As Double
133    Dim RQD As Double
134    RQD = clsRQDIndex.RQDIndex(PiecesString, kml)
135
136    prvRMR = RQu + RatingRQD(RQD) + Ra + Rb + Rc + Rd
```

```

137 End Function
138
139 Public Function RMR(Pieces As String, CoreLength As Double, Qu
As Double, Ra As Double, Rb As Double, Rc As Double, Rd As
Double)
140     Dim Dict As Scripting.Dictionary
141
142     Set Dict = New Scripting.Dictionary
143     Dict.Add rcxRMRPieces, Pieces
144     Dict.Add rcxRMRCLength, CoreLength
145
146     Dict.Add rcxRMRRQu, Qu
147     Dict.Add rcxRMRRa, Ra
148     Dict.Add rcxRMRRb, Rb
149     Dict.Add rcxRMRRc, Rc
150     Dict.Add rcxRMRRd, Rd
151
152     RMR = clsClassification Calculate(Dict)
153 End Function
154 Public Property Get ClassificationInterface() As
clsClassification
155     Set ClassificationInterface = Me
156 End Property
157
158 Public Function RatingRQD(RQD As Double) As Double
159     'Wertung RQD-Index
160     Select Case RQD
161         Case Is > 90: RatingRQD = 20
162         Case Is > 75: RatingRQD = 17
163         Case Is > 50: RatingRQD = 13
164         Case Is > 25: RatingRQD = 8
165         Case Else:   RatingRQD = 3
166     End Select
167 End Function
168 Public Function RatingQu(Qu As Double) As Double
169     'einaxiale Druckfestigkeit [MPa]
170     Select Case Qu
171         Case Is > 250: RatingQu = 15
172         Case Is > 100: RatingQu = 12
173         Case Is > 50:  RatingQu = 7
174         Case Is > 25:  RatingQu = 4
175         Case Is > 5:   RatingQu = 2
176         Case Is > 1:   RatingQu = 1
177         Case Is <= 1:  RatingQu = 0
178     End Select
179 End Function
180 Public Function RatingRa(AbstandMM As Double) As Double
181     'Kluftabstand in [mm]

```

```

182     Select Case AbstandMM
183         Case Is > 2000: RatingRa = 20
184         Case Is > 600:  RatingRa = 15
185         Case Is > 200:  RatingRa = 10
186         Case Is > 60:   RatingRa = 8
187         Case Else:     RatingRa = 5
188     End Select
189 End Function
190
191 Private Property Get clsClassification ParameterList() As
Scripting.IDictionary
192     Dim RQD As clsRQDIndex
193     Set RQD = New clsRQDIndex
194     Set clsClassification_ParameterList =
RQD.ClassificationInterface.ParameterList
195     With clsClassification_ParameterList
196         .Add rcxRMRRQu, "Wertung für Einaxiale
Gesteinsdruckfestigkeit"
197         .Add rcxRMRRQu, "Einaxiale Gesteinsdruckfestigkeit [MPa]
(alternative Bestimmung der Wertung für Einaxiale
Gesteinsfestigkeit)"
198         .Add rcxRMRRa, "Wertung für Kluftabstand (wird über
Bruchstücke berechnet, wenn nicht angegeben)"
199         .Add rcxRMRKluftabstand, "Kluftabstand [mm] (alternative
Bestimmung der Wertung des Kluftabstandes, wird über Bruchstücke
berechnet, wenn nicht angegeben)"
200         .Add rcxRMRRb, "Wertung für Kluftbeschaffenheit"
201         .Add rcxRMRRc, "Wertung für Raumstellung der Klüfte"
202         .Add rcxRMRRd, "Wertung für Zutritt von Grundwasser"
203     End With
204 End Property
205
206 Private Function clsClassification ShowParameterWindow(Parameter
As String, Optional StartValue As Variant, Optional OwnerForm As
Object, Optional TestPresence As Boolean = False) As Variant
207     Select Case LCase(Parameter)
208         Case LCase(rcxRMRRQu)
209             If TestPresence Then
210                 clsClassification ShowParameterWindow = True
211             Else
212                 With frmParameters
213                     .Caption = "Parameter bestimmen: " &
clsClassification_ClassificationName
214                     .Labell = "RQu - Festigkeit des intakten Gesteins;
Einaxiale Druckfestigkeit"
215                     prvAddParameter frmParameters, "15", StartValue,
"Einaxiale Druckfestigkeit > 250 MPa"

```

```

216     prvAddParameter frmParameters, "12", StartValue,
"Einaxiale Druckfestigkeit 100...250 MPa"
217     prvAddParameter frmParameters, " 7", StartValue,
"Einaxiale Druckfestigkeit 50...100 MPa"
218     prvAddParameter frmParameters, " 4", StartValue,
"Einaxiale Druckfestigkeit 25...50 MPa"
219     prvAddParameter frmParameters, " 2", StartValue,
"Einaxiale Druckfestigkeit 5...25 MPa"
220     prvAddParameter frmParameters, " 1", StartValue,
"Einaxiale Druckfestigkeit 1...5 MPa"
221     prvAddParameter frmParameters, " 0", StartValue,
"Einaxiale Druckfestigkeit < 1 MPa"
222
223     .Show 1, OwnerForm
224     If IsEmpty(.ExitValue) Then
225         clsClassification ShowParameterWindow = StartValue
226     Else
227         clsClassification ShowParameterWindow = .ExitValue
228     End If
229     End With
230 End If
231
232 Case LCase(rcxRMRRa)
233     If TestPresence Then
234         clsClassification_ShowParameterWindow = True
235     Else
236         With frmParameters
237             .Caption = "Parameter bestimmen: " &
clsClassification ClassificationName
238             .Labell = "Ra - Kluftabstand"
239             prvAddParameter frmParameters, "20", StartValue,
"Kluftabstand > 2 m"
240             prvAddParameter frmParameters, "15", StartValue,
"Kluftabstand 0,6...2,0 m"
241             prvAddParameter frmParameters, "10", StartValue,
"Kluftabstand 200...600 mm"
242             prvAddParameter frmParameters, " 8", StartValue,
"Kluftabstand 60...200 mm"
243             prvAddParameter frmParameters, " 5", StartValue,
"Kluftabstand < 60 mm"
244
245             .Show 1, OwnerForm
246             If IsEmpty(.ExitValue) Then
247                 clsClassification ShowParameterWindow = StartValue
248             Else
249                 clsClassification ShowParameterWindow = .ExitValue
250             End If
251             End With

```

```

252     End If
253
254     Case LCase(rcxRMRRb)
255     If TestPresence Then
256         clsClassification ShowParameterWindow = True
257     Else
258         With frmParameters
259             .Caption = "Parameter bestimmen: " &
clsClassification ClassificationName
260             .Labell = "Rb - Kluftbeschaffenheit"
261             prvAddParameter frmParameters, "30", StartValue, "sehr
raue, unverwitterte Oberfläche. nicht kontinuierlich, keine
Durchtrennung"
262             prvAddParameter frmParameters, "25", StartValue,
"schwach raue Oberfläche, Durchtrennung < 1 mm leicht
angewittert"
263             prvAddParameter frmParameters, "20", StartValue,
"schwach raue Oberfläche, Durchtrennung < 1 mm, stark
verwittert"
264             prvAddParameter frmParameters, "10", StartValue,
"harnischartige Oberfl. oder Letten < 5 mm Dicke oder
Öffnungsweite 1-5 mm durchgehend"
265             prvAddParameter frmParameters, " 0", StartValue,
"weiche Letten > 5 mm Dicke oder Öffnungsweite > 5 mm
durchgehend"
266
267             .Show 1, OwnerForm
268             If IsEmpty(.ExitValue) Then
269                 clsClassification ShowParameterWindow = StartValue
270             Else
271                 clsClassification_ShowParameterWindow = .ExitValue
272             End If
273             End With
274         End If
275
276     Case LCase(rcxRMRRc)
277     If TestPresence Then
278         clsClassification ShowParameterWindow = True
279     Else
280         With frmParameters
281             .Caption = "Parameter bestimmen: " &
clsClassification ClassificationName
282             .Labell = "Rc - Raumstellung der Klüfte"
283             prvAddParameter frmParameters, " 0", StartValue,
"Tunnel & Bergwerke: sehr günstig"
284             prvAddParameter frmParameters, "-2", StartValue,
"Tunnel & Bergwerke: günstig"

```

```

285     prvAddParameter frmParameters, "-5", StartValue,
"Tunnel & Bergwerke: gut"
286     prvAddParameter frmParameters, "-10", StartValue,
"Tunnel & Bergwerke: ungünstig"
287     prvAddParameter frmParameters, "-12", StartValue,
"Tunnel & Bergwerke: sehr ungünstig"
288
289     prvAddParameter frmParameters, " 0", StartValue,
"Gründungen: sehr günstig"
290     prvAddParameter frmParameters, "-2", StartValue,
"Gründungen: günstig"
291     prvAddParameter frmParameters, "-7", StartValue,
"Gründungen: gut"
292     prvAddParameter frmParameters, "-15", StartValue,
"Gründungen: ungünstig"
293     prvAddParameter frmParameters, "-25", StartValue,
"Gründungen: sehr ungünstig"
294
295     prvAddParameter frmParameters, " 0", StartValue,
"Hänge & Böschungen: sehr günstig"
296     prvAddParameter frmParameters, "-5", StartValue,
"Hänge & Böschungen: günstig"
297     prvAddParameter frmParameters, "-25", StartValue,
"Hänge & Böschungen: gut"
298     prvAddParameter frmParameters, "-50", StartValue,
"Hänge & Böschungen: ungünstig"
299
300     .Show 1, OwnerForm
301     If IsEmpty(.ExitValue) Then
302         clsClassification_ShowParameterWindow = StartValue
303     Else
304         clsClassification_ShowParameterWindow = .ExitValue
305     End If
306     End With
307     End If
308
309     Case LCase(rcxRMRRd)
310     If TestPresence Then
311         clsClassification_ShowParameterWindow = True
312     Else
313         With frmParameters
314             .Caption = "Parameter bestimmen: " &
clsClassification_ClassificationName
315             .Label1 = "Rd - Zutritt von Grundwasser"
316             prvAddParameter frmParameters, "15", StartValue,
"vollständig trocken (kein Zufluss)"

```

```

317     prvAddParameter frmParameters, "10", StartValue,
"feucht (<10 l/min Zufluss pro 10 m Tunnellänge)"
318     prvAddParameter frmParameters, " 7", StartValue, "nass
(10-25 l/min Zufluss pro 10 m Tunnellänge)"
319     prvAddParameter frmParameters, " 4", StartValue,
"tropfend (25-125 l/min Zufluss pro 10 m Tunnellänge)"
320     prvAddParameter frmParameters, " 0", StartValue,
"fließend (>125 l/min Zufluss pro 10 m Tunnellänge)"
321
322     .Show 1, OwnerForm
323     If .ExitValue = vbNullString Then
324         clsClassification_ShowParameterWindow = StartValue
325     Else
326         clsClassification_ShowParameterWindow = .ExitValue
327     End If
328     End With
329     End If
330
331     Case Else
332     If TestPresence Then
333         clsClassification_ShowParameterWindow = False
334     Else
335         clsClassification_ShowParameterWindow = Empty
336     End If
337     End Select
338     End Function
339
340     Private Sub prvAddParameter(frm As frmParameters, Value As
String, StartValue As Variant, Coll As String)
341     With frm.AddParameterItem(Value)
342         .AddCol Coll, False, 4500, vbLeftJustify
343         .AddCol Value, False, 1000, vbRightJustify
344
345         If CMe(StartValue) = CMe(Value) Then
346             .Selected = True
347         End If
348     End With
349     End Sub
350

```

Datei: clsRockMassQualityQ.cls (VB Class Module)

```

01     VERSION 1.0 CLASS
02     BEGIN
03         MultiUse = -1 'True
04         Persistable = 0 'NotPersistable
05         DataBindingBehavior = 0 'vbNone

```

```
06     DataSourceBehavior = 0 'vbNone
07     MTSTransactionMode = 0 'NotAnMTSObject
08 END
09 Attribute VB Name = "clsRockMassQualityQ"
10 Attribute VB GlobalNameSpace = False
11 Attribute VB_Creatable = True
12 Attribute VB_PredeclaredId = False
13 Attribute VB_Exposed = True
14 Option Explicit
15 Implements clsClassification
16
17 Private Const rcxRMQPieces As String = rcxPieces
18 Private Const rcxRMQCLength As String = rcxCLength
19 Private Const rcxRMQJn As String = "Jn"
20 Private Const rcxRMQJr As String = "Jr"
21 Private Const rcxRMQJa As String = "Ja"
22 Private Const rcxRMQJw As String = "Jw"
23 Private Const rcxRMQSRF As String = "SRF"
24
25 Private clsRQDIndex As clsRQDIndex
26
27 Private Sub Class Initialize()
28     Set clsClassification = New clsClassification
29     Set clsRQDIndex = New clsRQDIndex
30 End Sub
31
32 Private Sub Class Terminate()
33     Set clsRQDIndex = Nothing
34 End Sub
35
36 Private Function clsClassification_Calculate(Parameter As
Scripting.Dictionary) As Double
37     Dim RQD As Double, Jn As Double, Jr As Double, Ja As Double,
Jw As Double, SRF As Double
38
39     RQD = clsRQDIndex.ClassificationInterface.Calculate(Parameter)
40
41     Jn = GetItem(Parameter, rcxRMQJn, vbNullString, True)
42     Jr = GetItem(Parameter, rcxRMQJr, vbNullString, True)
43     Ja = GetItem(Parameter, rcxRMQJa, vbNullString, True)
44     Jw = GetItem(Parameter, rcxRMQJw, vbNullString, True)
45     SRF = GetItem(Parameter, rcxRMQSRF, vbNullString, True)
46
47     clsClassification.Calculate = prvRockMassQualityQ(RQD, Jn, Jr,
Ja, Jw, SRF)
48 End Function
49
```

```
50 Private Property Get clsClassification_ClassCount() As Long
51     clsClassification_ClassCount = 9
52 End Property
53
54 Private Function clsClassification_ClassDesignation(ByVal Value
As Double) As String
55     clsClassification_ClassDesignation =
clsClassification_ClassDesignationByIndex(clsClassification_Clas
sIndexByValue(Value))
56 End Function
57
58 Private Property Get clsClassification_ClassificationAutorYear()
As String
59     clsClassification_ClassificationAutorYear = "Barton, Lien,
Lund (1974)"
60 End Property
61
62 Private Property Get clsClassification_ClassificationName() As
String
63     clsClassification_ClassificationName = "Rock Mass Quality Q"
64 End Property
65
66 Private Function clsClassification_ClassIndexByValue(ByVal Value
As Double) As Long
67     Dim ValueBack As Long
68
69     Select Case Value
70         Case Is > 400: ValueBack = 1
71         Case Is > 100: ValueBack = 2
72         Case Is > 40: ValueBack = 3
73         Case Is > 10: ValueBack = 4
74         Case Is > 4: ValueBack = 5
75         Case Is > 1: ValueBack = 6
76         Case Is > 0.1: ValueBack = 7
77         Case Is > 0.01: ValueBack = 8
78         Case Else: ValueBack = 9
79     End Select
80     clsClassification_ClassIndexByValue = ValueBack
81 End Function
82 Private Function clsClassification_ClassDesignationByIndex(ByVal
Index As Long) As String
83     Dim Value As String
84
85     Select Case Index
86         Case 1: Value = "ausgezeichnet"
87         Case 2: Value = "extrem gut"
88         Case 3: Value = "sehr gut"
89         Case 4: Value = "gut"
```

```

90     Case 5: Value = "mittel"
91     Case 6: Value = "gering"
92     Case 7: Value = "sehr gering"
93     Case 8: Value = "extrem gering"
94     Case 9: Value = "äußert gering"
95     End Select
96     clsClassification_ClassDesignationByIndex = Value
97 End Function
98
99 Private Property Get clsClassification_FormatString() As String
100     clsClassification_FormatString = "0.000"
101 End Property
102
103 Private Property Get clsClassification_Unit() As String
104     clsClassification_Unit = ""
105 End Property
106
107 Private Property Get
108     clsClassification_ValueRangeMax(Berechnungslänge As Double) As Double
109     clsClassification_ValueRangeMax = 1000
110 End Property
111
112 Private Property Get clsClassification_ValueRangeMin() As Double
113     clsClassification_ValueRangeMin = 0.001
114 End Property
115
116 Private Function prvRockMassQualityQ(ByVal RQD As Double, ByVal
117     Jn As Double, ByVal Jr As Double, ByVal Ja As Double, ByVal Jw
118     As Double, ByVal SRF As Double) As Double
119     If RQD < 10 Then RQD = 10
120     prvRockMassQualityQ = RQD / Jn * Jr / Ja * Jw / SRF
121 End Function
122
123 Public Function RockMassQualityQ(Pieces As String, CoreLength As
124     Double, Jn As Double, Jr As Double, Ja As Double, Jw As Double,
125     SRF As Double)
126     Dim Dict As Scripting.Dictionary
127
128     Set Dict = New Scripting.Dictionary
129     Dict.Add rcxRMQPieces, Pieces
130     Dict.Add rcxRMQCLength, CoreLength
131     Dict.Add rcxRMQJn, Jn
132     Dict.Add rcxRMQJr, Jr
133     Dict.Add rcxRMQJa, Ja
134     Dict.Add rcxRMQJw, Jw
135     Dict.Add rcxRMQSRF, SRF

```

```

131
132     RockMassQualityQ = clsClassification_Calculate(Dict)
133 End Function
134 Public Property Get ClassificationInterface() As
135     clsClassification
136     Set ClassificationInterface = Me
137 End Property
138
139 Private Property Get clsClassification_ParameterList() As
140     Scripting.IDictionary
141     Dim RQD As clsRQDIndex
142     Set RQD = New clsRQDIndex
143     Set clsClassification_ParameterList =
144     RQD.ClassificationInterface.ParameterList
145     With clsClassification_ParameterList
146         .Add rcxRMQJn, "Wertung für Anzahl der Klufscharen"
147         .Add rcxRMQJr, "Wertung für Rauigkeit der schwächsten
148         Spaltebene"
149         .Add rcxRMQJa, "Wertung für Umwandlungsgrad"
150         .Add rcxRMQJw, "Wertung für Klufwasserführung"
151         .Add rcxRMQSRF, "Spannungsniveau"
152     End With
153 End Property
154
155 Private Function clsClassification_ShowParameterWindow(Parameter
156     As String, Optional StartValue As Variant, Optional OwnerForm As
157     Object, Optional TestPresence As Boolean = False) As Variant
158     Select Case LCase(Parameter)
159     Case LCase(rcxRMQSRF)
160         If TestPresence Then
161             clsClassification_ShowParameterWindow = True
162         Else
163             With frmParameters
164                 .Caption = "Parameter bestimmen: " &
165                 clsClassification_ClassificationName
166                 .Label1 = "SRF - Spannungsniveau"
167                 prvAddParameter frmParameters, "10,0", StartValue,
168                 "A", "Vielfaches Vorkommen von schwachen Zonen, die Ton oder
169                 chemisch verwittertes Gestein enthalten. Sehr loses umgebendes
170                 Gestein. (jede Tiefe des Vortriebes)"
171                 prvAddParameter frmParameters, " 5,0", StartValue,
172                 "B", "Einzelne Schwächezonen, die Ton oder chemisch verwittertes
173                 Gestein enthalten. (Tiefe des Vortriebes = 50 m)"
174                 prvAddParameter frmParameters, " 2,5", StartValue,
175                 "C", "Einzelne Schwächezonen, die Ton oder chemisch verwittertes
176                 Gestein enthalten. (Tiefe des Vortriebes > 50 m)"

```

```

163     prvAddParameter frmParameters, " 7,5", StartValue,
    "D", "Mehrfache Scherzonen im festen Gestein (tonfrei), loses
    umgebendes Gestein (jede Tiefe des Vortriebes)"
164     prvAddParameter frmParameters, " 5,0", StartValue,
    "E", "Einzelne Scherzonen im festen Gestein (frei von Ton),
    (Tiefe des Vortriebes = 50 m)"
165     prvAddParameter frmParameters, " 2,5", StartValue,
    "F", "Einzelne Scherzonen im festen Gestein (frei von Ton),
    (Tiefe des Vortriebes > 50 m)"
166     prvAddParameter frmParameters, " 5,0", StartValue,
    "G", "Aufglockerte offene Klüfte, stark zerklüftet (blockig
    gewürfelt, jede Tiefe des Vortriebes)"
167
168     prvAddParameter frmParameters, " 2,5", StartValue,
    "H", "geringe Spannung, nahe der Oberfläche SigmaC/SigmaI > 200
    SigmaT/SigmaI > 13"
169     prvAddParameter frmParameters, " 1,0", StartValue,
    "J", "mittlere Spannung SigmaC/SigmaI = 200...10
    SigmaT/SigmaI=13...0,66"
170     prvAddParameter frmParameters, " 1,3", StartValue,
    "K", "hohe Spannungen, geschlossene Strukturen (gewöhnlich
    günstig für Stabilität, kann ungünstig sein für Standsicherheit
    der Ulme) SigmaC/SigmaI = 10...5  SigmaT/SigmaI = 0,66...0,33"
171     prvAddParameter frmParameters, " 7,5", StartValue,
    "L", "nicht explosive Gebirgsschlaggefahr (massives Gestein)
    SigmaC/SigmaI = 5...2,5  SigmaT/SigmaI = 0,33...0,16"
172     prvAddParameter frmParameters, "15,0", StartValue,
    "M", "explosive Gebirgsschlaggefahr (massives Gestein)
    SigmaC/SigmaI < 2,5  SigmaT/SigmaI < 1,6"
173
174     prvAddParameter frmParameters, " 7,5", StartValue,
    "N", "geringfügig druckhaftes Gebirge"
175     prvAddParameter frmParameters, "15,0", StartValue,
    "O", "sehr druckhaftes Gebirge"
176     prvAddParameter frmParameters, " 7,5", StartValue,
    "P", "schwach quellendes Gebirge"
177     prvAddParameter frmParameters, "12,5", StartValue,
    "R", "stark quellendes Gebirge"
178
179     .Show 1, OwnerForm
180     If IsEmpty(.ExitValue) Then
181         clsClassification_ShowParameterWindow = StartValue
182     Else
183         clsClassification_ShowParameterWindow = .ExitValue
184     End If
185     End With
186 End If

```

```

187
188     Case LCase(rcxRMQJn)
189     If TestPresence Then
190         clsClassification_ShowParameterWindow = True
191     Else
192         With frmParameters
193             .Caption = "Parameter bestimmen: " &
    clsClassification_ClassificationName
194             .Labell = "Jn - Anzahl der Klufscharen"
195             prvAddParameter frmParameters, " 0,7", StartValue,
    "A", "massiv, keine oder wenige Klüfte"
196             prvAddParameter frmParameters, " 2,0", StartValue,
    "B", "eine Kluftschar"
197             prvAddParameter frmParameters, " 3,0", StartValue,
    "C", "eine Kluftschar, zusätzliches Auftreten von ungerichteten
    Einzelklüften"
198             prvAddParameter frmParameters, " 4,0", StartValue,
    "D", "zwei Klufscharen"
199             prvAddParameter frmParameters, " 6,0", StartValue,
    "E", "zwei Klufscharen, zusätzliches Auftreten von
    ungerichteten Einzelklüften"
200             prvAddParameter frmParameters, " 9,0", StartValue,
    "F", "drei Klufscharen"
201             prvAddParameter frmParameters, "12,0", StartValue,
    "G", "drei Klufscharen, zusätzliches Auftreten von
    ungerichteten Einzelklüften"
202             prvAddParameter frmParameters, "15,0", StartValue,
    "H", "mehr als vier Klufscharen, ungerichtete Klüfte, stark
    zerklüftet"
203             prvAddParameter frmParameters, "20,0", StartValue,
    "J", "zerdrücktes Gebirge"
204
205             .Show 1, OwnerForm
206             If IsEmpty(.ExitValue) Then
207                 clsClassification_ShowParameterWindow = StartValue
208             Else
209                 clsClassification_ShowParameterWindow = .ExitValue
210             End If
211         End With
212     End If
213
214     Case LCase(rcxRMQJr)
215     If TestPresence Then
216         clsClassification_ShowParameterWindow = True
217     Else
218         With frmParameters
219             .Caption = "Parameter bestimmen: " &
    clsClassification_ClassificationName

```

```

220     .Labell = "Jr - Rauigkeit für die schwächste
Spaltebene"
221     prvAddParameter frmParameters, " 4,0", StartValue,
"A", "nicht durchstreichende Kluft"
222     prvAddParameter frmParameters, " 3,0", StartValue,
"B", "rau oder unregelmäßig, wellig"
223     prvAddParameter frmParameters, " 2,0", StartValue,
"C", "glatt, wellig"
224     prvAddParameter frmParameters, " 1,5", StartValue,
"D", "beharnischt, wellig"
225     prvAddParameter frmParameters, " 1,5", StartValue,
"E", "rau oder unregelmäßig, eben"
226     prvAddParameter frmParameters, " 1,0", StartValue,
"F", "glatt, eben"
227     prvAddParameter frmParameters, " 0,5", StartValue,
"G", "beharnischt, eben"
228     prvAddParameter frmParameters, " 1,0", StartValue, "H
& J", "kein Wand-Wand-Kontakt, Letten-, Ton- oder Sandfüllung"
229
230     .Show 1, OwnerForm
231     If IsEmpty(.ExitValue) Then
232         clsClassification ShowParameterWindow = StartValue
233     Else
234         clsClassification_ShowParameterWindow = .ExitValue
235     End If
236     End With
237 End If
238
239 Case LCase(rcxRMQJw)
240     If TestPresence Then
241         clsClassification_ShowParameterWindow = True
242     Else
243         With frmParameters
244             .Caption = "Parameter bestimmen: " &
clsClassification ClassificationName
245             .Labell = "Jw - Kluftwasserführung"
246             prvAddParameter frmParameters, " 1,0", StartValue,
"A", "Trockener Vortrieb oder örtlich geringer Zufluss (> 5
l/min)" & vbCrLf & "Wasserdruck <0,1 N/mm²"
247             prvAddParameter frmParameters, "0,66", StartValue,
"B", "Mittlerer Zufluss oder Druck, gelegentlich Ausspülung der
Kluftfüllungen" & vbCrLf & "Wasserdruck 0,10...0,25 N/mm²"
248             prvAddParameter frmParameters, " 0,5", StartValue,
"C", "starker Zufluss oder hoher Druck in festem Gestein mit
ungefüllten Klüften" & vbCrLf & "Wasserdruck 0,25...1,0 N/mm²"

```

```

249         prvAddParameter frmParameters, "0,33", StartValue,
"D", "starker Zufluss oder hoher Druck, bedeutende Ausspülungen
der Kluftfüllungen" & vbCrLf & "Wasserdruck 0,25...1,0 N/mm²"
250         prvAddParameter frmParameters, "0,15", StartValue,
"E", "Übermäßig hoher Zufluss oder Wasserdruck beim Sprengen,
mit der Zeit abnehmend" & vbCrLf & "Wasserdruck >1,0 N/mm²"
251         prvAddParameter frmParameters, " 0,1", StartValue,
"F", "Übermäßig hoher Zufluss oder Wasserdruck, gleichbleibend
ohne bemerkenswerte Abnahme" & vbCrLf & "Wasserdruck >1,0 N/mm²"
252
253     .Show 1, OwnerForm
254     If IsEmpty(.ExitValue) Then
255         clsClassification_ShowParameterWindow = StartValue
256     Else
257         clsClassification ShowParameterWindow = .ExitValue
258     End If
259     End With
260 End If
261
262 Case LCase(rcxRMQJa)
263     If TestPresence Then
264         clsClassification ShowParameterWindow = True
265     Else
266         With frmParameters
267             .Caption = "Parameter bestimmen: " &
clsClassification ClassificationName
268             .Labell = "Ja - Umwandlungsgrad"
269             prvAddParameter frmParameters, "0,75", StartValue,
"A", "fest verheilt, hart, nicht entfestigt, undurchlässige
Füllung (z.B. Quarz, Epidote)"
270             prvAddParameter frmParameters, " 1,0", StartValue,
"B", "nicht umgewandelte, verfärbte Kluftflächen" & vbCrLf &
"Phi = 25° - 35°"
271             prvAddParameter frmParameters, " 2,0", StartValue,
"C", "geringfügig verwitterte Kluftflächen, nicht entfestigte
Kluftbeläge, sandige Partikel, verwittertes Gestein" & vbCrLf &
"Phi = 25° - 30°"
272             prvAddParameter frmParameters, " 3,0", StartValue,
"D", "siltige oder sandige, tonige Kluftbeläge, geringe
Tonfraktion" & vbCrLf & "Phi = 20° - 25°"
273             prvAddParameter frmParameters, " 4,0", StartValue,
"E", "erweichbar oder mit geringer Reibung, Kluftbeläge (z.B.:
Kaolit, Chlorite, Talk, Gips, Graphit), geringe Mengen von
quellfähigem Ton (nicht durchstreichende Kluftbeläge, Dicke:
kleiner als 1 bis 2 mm)" & vbCrLf & "Phi = 8° - 16°"
274             prvAddParameter frmParameters, " 4,0", StartValue,
"F", "sandige Partikel, tonfreies verwittertes Gestein" & vbCrLf
& "Phi = 25° - 30°"

```

```

275     prvAddParameter frmParameters, " 6,0", StartValue,
"    G", "stark überverdichtet, verhärtet, nicht erweichbare
Tonmineralfüllungen (durchgehend, Dicke: < 5 mm)" & vbCrLf &
"Phi = 16° - 24°"
276     prvAddParameter frmParameters, " 8,0", StartValue,
"    H", "mittel oder gering verdichtet, erweichbar, tonige
Mineralfüllung (durchgehend, Dicke: < 5 mm)" & vbCrLf & "Phi =
12° - 16°"
277     prvAddParameter frmParameters, "10,0", StartValue,
"    J", "quellfähige tonige Füllungen z.B.: Montmorillonit
(durchgehend, Dicke: < 5 mm), (die Werte von Ja hängen vom
Anteil quellfähiger Partikel ab und sind anfällig auf
Wasserzutritt)" & vbCrLf & "Phi = 6° - 12°"
278     prvAddParameter frmParameters, " 6,0", StartValue,
"    K", "Zonen oder Bänder von verwittertem" & vbCrLf & "Phi = 6° -
24°"
279     prvAddParameter frmParameters, " 8,0", StartValue,
"    L", "oder zerbrochenem Gestein und" & vbCrLf & "Phi = 6° - 24°"
280     prvAddParameter frmParameters, "10,0", StartValue,
"    M", "Ton (siehe G, H, J für die Beschreibung des Tones)" &
vbCrLf & "Phi = 6° - 24°"
281     prvAddParameter frmParameters, " 5,0", StartValue,
"    N", "Zonen oder Bänder von siltigem oder sandigem Ton, geringe
Tonfraktion (nicht erweichbar)"
282     prvAddParameter frmParameters, "10,0", StartValue,
"    O", "Dicke, durchgehende Bänder und" & vbCrLf & "Phi = 6° -
24°"
283     prvAddParameter frmParameters, "13,0", StartValue,
"    P", "Zonen von Ton (siehe G, H, J, für die Beschreibung des
Tones)" & vbCrLf & "Phi = 6° - 24°"
284
285     .Show 1, OwnerForm
286     If IsEmpty(.ExitValue) Then
287         clsClassification ShowParameterWindow = StartValue
288     Else
289         clsClassification_ShowParameterWindow = .ExitValue
290     End If
291 End With
292 End If
293
294 Case Else
295     If TestPresence Then
296         clsClassification_ShowParameterWindow = False
297     Else
298         clsClassification ShowParameterWindow = Empty
299     End If
300 End Select

```

```

301 End Function
302 Private Sub prvAddParameter(frm As frmParameters, Value As
String, StartValue As Variant, Col1 As String, Col2 As String)
303     With frm.AddParameterItem(Value)
304         .AddCol Col1, True, 500, vbCenter
305         .AddCol Col2, False, 4000, vbLeftJustify
306         .AddCol Value, False, 1000, vbRightJustify
307
308         If CMe(StartValue) = CMe(Value) Then
309             .Selected = True
310         End If
311     End With
312 End Sub
313

```

Datei: clsRockMassQualityQTBM.cls (VB Class Module)

```

01 VERSION 1.0 CLASS
02 BEGIN
03     MultiUse = -1 'True
04     Persistable = 0 'NotPersistable
05     DataBindingBehavior = 0 'vbNone
06     DataSourceBehavior = 0 'vbNone
07     MTSTransactionMode = 0 'NotAnMTSObject
08 END
09 Attribute VB_Name = "clsRockMassQualityQTBM"
10 Attribute VB_GlobalNameSpace = False
11 Attribute VB_Creatable = True
12 Attribute VB_PredeclaredId = False
13 Attribute VB_Exposed = True
14 Option Explicit
15 Implements clsClassification
16
17 Private Const rcxRMQTBMpieces As String = rcxPieces
18 Private Const rcxRMQTBMCLength As String = rcxCLength
19 Private Const rcxRMQTBMJn As String = "Jn"
20 Private Const rcxRMQTBMJr As String = "Jr"
21 Private Const rcxRMQTBMJa As String = "Ja"
22 Private Const rcxRMQTBMJw As String = "Jw"
23 Private Const rcxRMQTBMMSRF As String = "SRF"
24
25 Private Const rcxRMQTBMsigma As String = "Sigma"
26 Private Const rcxRMQTBMF As String = "F"
27 Private Const rcxRMQTBMCLI As String = "CLI"
28 Private Const rcxRMQTBMq As String = "q"
29 Private Const rcxRMQTBMms As String = "s"
30

```

```

31 Private clsRockMassQualityQ As clsRockMassQualityQ
32
33 Private Sub Class_Initialize()
34     Set clsClassification = New clsClassification
35     Set clsRockMassQualityQ = New clsRockMassQualityQ
36 End Sub
37
38 Private Sub Class_Terminate()
39     Set clsRockMassQualityQ = Nothing
40 End Sub
41
42 Private Function clsClassification Calculate(Parameter As
Scripting.Dictionary) As Double
43     Dim RockMassQualityQ As Double, Sigma As Double, F As Double,
CLI As Double, Q As Double, s As Double
44
45     RockMassQualityQ =
clsRockMassQualityQ.ClassificationInterface.Calculate(Parameter)
46     Sigma = GetItem(Parameter, rcxRMQTBSigma, vbNullString, True)
47     F = GetItem(Parameter, rcxRMQTBMF, vbNullString, True)
48     CLI = GetItem(Parameter, rcxRMQTBMLI, vbNullString, True)
49     Q = GetItem(Parameter, rcxRMQTBmq, vbNullString, True)
50     s = GetItem(Parameter, rcxRMQTBms, vbNullString, True)
51
52     clsClassification_Calculate =
prvRockMassQualityQTBM(RockMassQualityQ, Sigma, F, CLI, Q, s)
53 End Function
54
55 Private Property Get clsClassification ClassCount() As Long
56 End Property
57
58 Private Function clsClassification ClassDesignation(ByVal Value
As Double) As String
59     clsClassification ClassDesignation =
clsClassification ClassDesignationByIndex(clsClassification Clas
sIndexByValue(Value))
60 End Function
61
62 Private Property Get clsClassification ClassificationAutorYear()
As String
63     clsClassification ClassificationAutorYear = "Barton (2000)"
64 End Property
65
66 Private Property Get clsClassification ClassificationName() As
String
67     clsClassification ClassificationName = "Rock Mass Quality
QTBM"
68 End Property

```

```

69
70 Private Function clsClassification_ClassIndexByValue(ByVal Value
As Double) As Long
71 End Function
72 Private Function clsClassification ClassDesignationByIndex(ByVal
Index As Long) As String
73 End Function
74
75 Private Property Get clsClassification FormatString() As String
76     clsClassification FormatString = "0.000"
77 End Property
78
79 Private Property Get clsClassification_Unit() As String
80     clsClassification_Unit = ""
81 End Property
82
83 Private Property Get
clsClassification ValueRangeMax(Berechnungslänge As Double) As
Double
84     clsClassification_ValueRangeMax = 1000
85 End Property
86
87 Private Property Get clsClassification ValueRangeMin() As Double
88     clsClassification_ValueRangeMin = 0.001
89 End Property
90
91 Private Function prvRockMassQualityQTBM(RockMassQualityQ As
Double, Sigma As Double, F As Double, CLI As Double, Q As
Double, s As Double) As Double
92     prvRockMassQualityQTBM = RockMassQualityQ * Sigma / (F ^ 10 /
20 ^ 9) * 20 / CLI * Q / 20 * s / 5
93 End Function
94
95 Public Function RockMassQualityQTBM(Pieces As String, CoreLength
As Double, Jn As Double, Jr As Double, Ja As Double, Jw As
Double, SRF As Double, Sigma As Double, F As Double, CLI As
Double, Q As Double, s As Double)
96     Dim Dict As Scripting.Dictionary
97
98     Set Dict = New Scripting.Dictionary
99     Dict.Add rcxRMQTBMPieces, Pieces
100    Dict.Add rcxRMQTBMLength, CoreLength
101    Dict.Add rcxRMQTBMJn, Jn
102    Dict.Add rcxRMQTBMJr, Jr
103    Dict.Add rcxRMQTBMJa, Ja
104    Dict.Add rcxRMQTBMJw, Jw
105    Dict.Add rcxRMQTBMSRF, SRF
106

```

```

107 Dict.Add rcxRMQTBMSigma, Sigma
108 Dict.Add rcxRMQTBMF, F
109 Dict.Add rcxRMQTBMLI, CLI
110 Dict.Add rcxRMQTBmq, Q
111 Dict.Add rcxRMQTBMs, s
112
113 RockMassQualityQTBM = clsClassification_Calculate(Dict)
114 End Function
115 Public Property Get ClassificationInterface() As
clsClassification
116 Set ClassificationInterface = Me
117 End Property
118
119 Private Property Get clsClassification_ParameterList() As
Scripting.IDictionary
120 Dim Q As clsRockMassQualityQ
121 Set Q = New clsRockMassQualityQ
122 Set clsClassification_ParameterList =
Q.ClassificationInterface.ParameterList
123 With clsClassification_ParameterList
124 .Add rcxRMQTBMSigma, "geschätzte Gesteinsfestigkeit [MPa]"
125 .Add rcxRMQTBMF, "durchschnittliche Schneidemenge"
126 .Add rcxRMQTBMLI, "Cutter Life Index"
127 .Add rcxRMQTBmq, "Quarzgehalt [%]"
128 .Add rcxRMQTBMs, "induzierte biaxiale Spannung"
129 End With
130 End Property
131
132 Private Function clsClassification_ShowParameterWindow(Parameter
As String, Optional StartValue As Variant, Optional ParentWindow
As Object, Optional TestPresence As Boolean = False) As Variant
133 If TestPresence Then
134 clsClassification_ShowParameterWindow = False
135 Else
136 clsClassification_ShowParameterWindow = StartValue
137 End If
138 End Function
139

```

Datei: clsRQDIndex.cls (VB Class Module)

```

01 VERSION 1.0 CLASS
02 BEGIN
03 MultiUse = -1 'True
04 Persistable = 0 'NotPersistable
05 DataBindingBehavior = 0 'vbNone
06 DataSourceBehavior = 0 'vbNone

```

```

07 MTSTransactionMode = 0 'NotAnMSTObject
08 END
09 Attribute VB Name = "clsRQDIndex"
10 Attribute VB GlobalNameSpace = False
11 Attribute VB Creatable = True
12 Attribute VB_PredeclaredId = False
13 Attribute VB_Exposed = True
14 Option Explicit
15 Implements clsClassification
16
17 Private Const rcxRQDPieces As String = rcxPieces
18 Private Const rcxRQDCLength As String = rcxCLength
19
20 Private Sub Class_Initialize()
21 Set clsClassification = New clsClassification
22 End Sub
23
24 Private Function clsClassification_Calculate(Parameter As
Scripting.Dictionary) As Double
25 Dim Pieces As String, kml As Double
26 Pieces = GetItem(Parameter, rcxRQDPieces, vbNullString, True)
27 kml = Val(GetItem(Parameter, rcxRQDCLength, vbNullString,
True))
28
29 clsClassification_Calculate =
prvRQDIndex(PhrasePiecesExt(Pieces, kml,
clsClassification_ClassificationName), kml)
30 End Function
31
32 Private Property Get clsClassification_ClassCount() As Long
33 clsClassification_ClassCount = 5
34 End Property
35
36 Private Function clsClassification_ClassDesignation(ByVal Value
As Double) As String
37 clsClassification_ClassDesignation =
clsClassification_ClassDesignationByIndex(clsClassification_Clas
sIndexByValue(Value))
38 End Function
39
40 Private Property Get clsClassification_ClassificationAutorYear()
As String
41 clsClassification_ClassificationAutorYear = "Deere (196?)"
42 End Property
43
44 Private Property Get clsClassification_ClassificationName() As
String
45 clsClassification_ClassificationName = "RQD-Index"

```

```

46 End Property
47
48 Private Function clsClassification ClassIndexByValue(ByVal Value
As Double) As Long
49     Dim ValueBack As Long
50
51     Select Case Value
52         Case Is <= 25: ValueBack = 1
53         Case Is <= 50: ValueBack = 2
54         Case Is <= 75: ValueBack = 3
55         Case Is <= 90: ValueBack = 4
56         Case Else: ValueBack = 5
57     End Select
58     clsClassification_ClassIndexByValue = ValueBack
59 End Function
60 Private Function clsClassification ClassDesignationByIndex(ByVal
Index As Long) As String
61     Dim Value As String
62
63     Select Case Index
64         Case 1: Value = "sehr gering (very poor)"
65         Case 2: Value = "gering (poor)"
66         Case 3: Value = "mittel (fair)"
67         Case 4: Value = "gut (good)"
68         Case 5: Value = "sehr gut (excellent)"
69     End Select
70     clsClassification ClassDesignationByIndex = Value
71 End Function
72
73 Private Property Get clsClassification_FormatString() As String
74     clsClassification_FormatString = "0"
75 End Property
76
77 Private Property Get clsClassification Unit() As String
78     clsClassification Unit = "%"
79 End Property
80
81 Private Property Get
clsClassification ValueRangeMax(Berechnungslänge As Double) As
Double
82     clsClassification ValueRangeMax = 100
83 End Property
84
85 Private Property Get clsClassification ValueRangeMin() As Double
86     clsClassification ValueRangeMin = 0
87 End Property
88

```

```

89 Private Function prvRQDIndex(Pieces As Pieces, kml As Double) As
Double
90     Dim N As Long, v As Double           'Hilfsvariablen
91
92     v = 0                                'Null setzten (ist
eigentlich nicht nötig, sicher ist sicher)
93     For N = 1 To UBound(Pieces.Piece) 'kein Stück vergessen...
94         With Pieces.Piece(N)           'dann gehts ein wenig
schneller
95             If .Length > 10 Then       'min. Bruchstücklänge
96                 Select Case .PreservationDegree
97                     Case pdNone, pdA, pdB 'nur Erhaltungsgrad A, B &
NONE
98                         v = v + .Length
99                     Case Else           'falscher Erhaltungsgrad
100                         'nix
101                     End Select
102             End If
103         End With
104     Next
105     prvRQDIndex = v / kml * 100         'eigentliche Berechnung &
zurückgeben
106 End Function
107
108 Public Function RQDIndex(Pieces As String, CoreLength As Double)
109     Dim Dict As Scripting.Dictionary
110
111     Set Dict = New Scripting.Dictionary
112     Dict.Add rcxRQDPieces, Pieces
113     Dict.Add rcxRQDCLength, CoreLength
114
115     RQDIndex = clsClassification Calculate(Dict)
116 End Function
117 Public Property Get ClassificationInterface() As
clsClassification
118     Set ClassificationInterface = Me
119 End Property
120 Private Property Get clsClassification ParameterList() As
Scripting.IDictionary
121     Set clsClassification ParameterList = New Scripting.Dictionary
122     With clsClassification ParameterList
123         .CompareMode = TextCompare
124         .Add rcxRQDPieces, "Bruchstücke der Berechnungseinheit"
125         .Add rcxRQDCLength, "Länge der Berechnungseinheit"
126     End With
127 End Property
128

```

```

129 Private Function clsClassification_ShowParameterWindow(Parameter
As String, Optional StartValue As Variant, Optional ParentWindow
As Object, Optional TestPresence As Boolean = False) As Variant
130     If TestPresence Then
131         clsClassification_ShowParameterWindow = False
132     Else
133         clsClassification_ShowParameterWindow = StartValue
134     End If
135 End Function
136
137

```

Datei: clsSMR.cls (VB Class Module)

```

01 VERSION 1.0 CLASS
02 BEGIN
03     MultiUse = -1 'True
04     Persistable = 0 'NotPersistable
05     DataBindingBehavior = 0 'vbNone
06     DataSourceBehavior = 0 'vbNone
07     MTSTransactionMode = 0 'NotAnMTSObject
08 END
09 Attribute VB Name = "clsSMR"
10 Attribute VB GlobalNameSpace = False
11 Attribute VB Creatable = True
12 Attribute VB_PredeclaredId = False
13 Attribute VB_Exposed = True
14 Option Explicit
15 Implements clsClassification
16
17 Private Const rcxSMRPieces As String = rcxPieces
18 Private Const rcxSMRCLength As String = rcxCLength
19 Private Const rcxSMRQu As String = "Qu"
20 Private Const rcxSMRRQu As String = "RQu"
21 Private Const rcxSMRRa As String = "Ra"
22 Private Const rcxSMRRb As String = "Rb"
23 Private Const rcxSMRRc As String = "Rc"
24 Private Const rcxSMRRd As String = "Rd"
25 Private Const rcxSMRF1 As String = "F1"
26 Private Const rcxSMRF1Alt As String = "Alpha"
27 Private Const rcxSMRF2 As String = "F2"
28 Private Const rcxSMRF2Alt As String = "Beta"
29 Private Const rcxSMRF3 As String = "F3"
30 Private Const rcxSMRF4 As String = "F4"
31
32 Private clsRMR As clsRMR
33

```

```

34 Private Sub Class_Initialize()
35     Set clsClassification = New clsClassification
36     Set clsRMR = New clsRMR
37 End Sub
38
39 Private Sub Class_Terminate()
40     Set clsRMR = Nothing
41 End Sub
42
43 Private Function clsClassification_Calculate(Parameter As
Scripting.Dictionary) As Double
44     Dim RMR As Double, F1 As Double, F2 As Double, F3 As Double,
F4 As Double
45     Dim v As Variant
46
47     RMR = clsRMR.ClassificationInterface.Calculate(Parameter)
48
49     v = GetItem(Parameter, rcxSMRF1Alt, "leer", False)
50     If v = "leer" Then
51         F1 = Val(GetItem(Parameter, rcxSMRF1, vbNullString, True))
52     Else
53         F1 = CalculateF1(CMe(v / 180 * Atn(1) * 4))
54     End If
55
56     v = GetItem(Parameter, rcxSMRF2Alt, "leer", False)
57     If v = "leer" Then
58         F2 = Val(GetItem(Parameter, rcxSMRF2, vbNullString, True))
59     Else
60         F2 = CalculateF2(CMe(v / 180 * Atn(1) * 4))
61     End If
62
63     F3 = Val(GetItem(Parameter, rcxSMRF3, vbNullString, True))
64     F4 = Val(GetItem(Parameter, rcxSMRF4, vbNullString, True))
65
66     clsClassification.Calculate = prvSMR(RMR, F1, F2, F3, F4)
67 End Function
68
69 Private Property Get clsClassification_ClassCount() As Long
70     clsClassification.ClassCount = 5
71 End Property
72
73 Private Function clsClassification_ClassDesignation(ByVal Value
As Double) As String
74     clsClassification.ClassDesignation =
clsClassification.ClassDesignationByIndex(clsClassification.Clas
sIndexByValue(Value))
75 End Function
76

```

```

77 Private Property Get clsClassification_ClassificationAutorYear()
As String
78     clsClassification ClassificationAutorYear = "Romana (1985)"
79 End Property
80
81 Private Property Get clsClassification_ClassificationName() As
String
82     clsClassification_ClassificationName = "SMR-System"
83 End Property
84
85 Private Function clsClassification_ClassIndexByValue(ByVal Value
As Double) As Long
86     Dim ValueBack As Long
87
88     Select Case Value
89         Case Is > 80: ValueBack = 1
90         Case Is > 60: ValueBack = 2
91         Case Is > 40: ValueBack = 3
92         Case Is > 20: ValueBack = 4
93         Case Else: ValueBack = 5
94     End Select
95     clsClassification ClassIndexByValue = ValueBack
96 End Function
97 Private Function clsClassification_ClassDesignationByIndex(ByVal
Index As Long) As String
98     Dim Value As String
99
100    Select Case Index
101        Case 1: Value = "sehr gut"
102        Case 2: Value = "gut"
103        Case 3: Value = "normal"
104        Case 4: Value = "schlecht"
105        Case 5: Value = "sehr schlecht"
106    End Select
107    clsClassification ClassDesignationByIndex = Value
108 End Function
109
110 Private Property Get clsClassification_FormatString() As String
111     clsClassification FormatString = "0"
112 End Property
113
114 Private Property Get clsClassification_Unit() As String
115     clsClassification_Unit = ""
116 End Property
117
118 Private Property Get
clsClassification ValueRangeMax(Berechnungslänge As Double) As
Double

```

```

119     clsClassification_ValueRangeMax = 100
120 End Property
121
122 Private Property Get clsClassification ValueRangeMin() As Double
123     clsClassification ValueRangeMin = 0
124 End Property
125
126 Private Function prvSMR(RMR As Double, F1 As Double, F2 As
Double, F3 As Double, F4 As Double) As Double
127     prvSMR = RMR + (F1 * F2 * F3) + F4
128 End Function
129
130 Public Function SMR(Pieces As String, CoreLength As Double, Qu
As Double, Ra As Double, Rb As Double, Rc As Double, Rd As
Double, F1 As Double, F2 As Double, F3 As Double, F4 As Double)
131     Dim Dict As Scripting.Dictionary
132
133     Set Dict = New Scripting.Dictionary
134     Dict.Add rcxSMRPieces, Pieces
135     Dict.Add rcxSMRCLength, CoreLength
136
137     Dict.Add rcxSMRQu, Qu
138     Dict.Add rcxSMRRa, Ra
139     Dict.Add rcxSMRRb, Rb
140     Dict.Add rcxSMRRc, Rc
141     Dict.Add rcxSMRRd, Rd
142
143     Dict.Add rcxSMRF1, F1
144     Dict.Add rcxSMRF2, F2
145     Dict.Add rcxSMRF3, F3
146     Dict.Add rcxSMRF4, F4
147
148     SMR = clsClassification Calculate(Dict)
149 End Function
150 Public Property Get ClassificationInterface() As
clsClassification
151     Set ClassificationInterface = Me
152 End Property
153
154 Public Function CalculateF1(AlphaRAD As Double) As Double
155     CalculateF1 = (1 - Sin(AlphaRAD)) ^ 2
156 End Function
157 Public Function CalculateF2(BetaRAD As Double) As Double
158     Select Case BetaRAD / (Atn(1) * 4) * 180
159         Case Is <= 21: CalculateF2 = 0.15
160         Case Is >= 45: CalculateF2 = 1
161         Case Else: CalculateF2 = Tan(BetaRAD) ^ 2
162     End Select

```

```

163 End Function
164
165 Private Property Get clsClassification ParameterList() As
Scripting.IDictionary
166     Dim RMR As clsRMR
167     Set RMR = New clsRMR
168     Set clsClassification_ParameterList =
RMR.ClassificationInterface.ParameterList
169     With clsClassification_ParameterList
170         .Add rcxSMRF1, "Wertung für Winkel zwischen Hang und Klufft"
171         .Add rcxSMRF1Alt, "alternative Bestimmung für F1 über Winkel
Alpha [Grad] (Winkel zwischen Hang und Klufft)"
172         .Add rcxSMRF2, "Wertung für Einfallwinkel für den ebenen
Versagensfall"
173         .Add rcxSMRF2Alt, "alternative Bestimmung für F2 über Winkel
Beta [Grad] (Kluffteinfallwinkel)"
174         .Add rcxSMRF3, "Wertung für Relation zwischen Hangoberfläche
und Kluffteinfallen"
175         .Add rcxSMRF4, "Wertung für Vorrichtungsmethode"
176     End With
177 End Property
178
179 Private Function clsClassification ShowParameterWindow(Parameter
As String, Optional StartValue As Variant, Optional OwnerForm As
Object, Optional TestPresence As Boolean = False) As Variant
180     Select Case LCase(Parameter)
181     Case LCase(rcxSMRF3)
182         If TestPresence Then
183             clsClassification ShowParameterWindow = True
184         Else
185             With frmParameters
186                 .Caption = "Parameter bestimmen: " &
clsClassification ClassificationName
187                 .Labell = "F3 - Relation zwischen Hangoberfläche und
Kluffteinfallen"
188                 prvAddParameter frmParameters, " 0", StartValue,
"sehr günstig"
189                 prvAddParameter frmParameters, "-6", StartValue,
"gunstig"
190                 prvAddParameter frmParameters, "-25", StartValue,
"gut"
191                 prvAddParameter frmParameters, "-50", StartValue,
"ungunstig"
192                 prvAddParameter frmParameters, "-60", StartValue,
"sehr ungunstig"
193
194                 .Show 1, OwnerForm
195                 If IsEmpty(.ExitValue) Then

```

```

196             clsClassification_ShowParameterWindow = StartValue
197         Else
198             clsClassification ShowParameterWindow = .ExitValue
199         End If
200     End With
201 End If
202
203 Case LCase(rcxSMRF4)
204     If TestPresence Then
205         clsClassification ShowParameterWindow = True
206     Else
207         With frmParameters
208             .Caption = "Parameter bestimmen: " &
clsClassification ClassificationName
209             .Labell = "F4 - Vorrichtungsmethode"
210             prvAddParameter frmParameters, "15", StartValue,
"natürlicher Hang"
211             prvAddParameter frmParameters, "10", StartValue,
"Vorbereiten"
212             prvAddParameter frmParameters, " 8", StartValue,
"schonendes Sprengen"
213             prvAddParameter frmParameters, " 0", StartValue,
"Sprengen oder mechanische Vorrichtung"
214             prvAddParameter frmParameters, "-8", StartValue,
"mangelhaftes Sprengen"
215
216             .Show 1, OwnerForm
217             If IsEmpty(.ExitValue) Then
218                 clsClassification ShowParameterWindow = StartValue
219             Else
220                 clsClassification_ShowParameterWindow = .ExitValue
221             End If
222         End With
223     End If
224
225 Case Else
226     If TestPresence Then
227         clsClassification ShowParameterWindow = False
228     Else
229         clsClassification ShowParameterWindow = Empty
230     End If
231 End Select
232 End Function
233
234 Private Sub prvAddParameter(frm As frmParameters, Value As
String, StartValue As Variant, Coll As String)
235     With frm.AddParameterItem(Value)
236         .AddCol Coll, False, 4500, vbLeftJustify

```

```

237 .AddCol Value, False, 1000, vbRightJustify
238
239 If CMe(StartValue) = CMe(Value) Then
240     .Selected = True
241 End If
242 End With
243 End Sub
244

```

Datei: clsZerrüttungsgradZG.cls (VB Class Module)

```

01 VERSION 1.0 CLASS
02 BEGIN
03     MultiUse = -1 'True
04     Persistable = 0 'NotPersistable
05     DataBindingBehavior = 0 'vbNone
06     DataSourceBehavior = 0 'vbNone
07     MTSTransactionMode = 0 'NotAnMTSObject
08 END
09 Attribute VB_Name = "clsZerrüttungsgradZG"
10 Attribute VB_GlobalNameSpace = False
11 Attribute VB_Creatable = True
12 Attribute VB_PredeclaredId = False
13 Attribute VB_Exposed = True
14 Option Explicit
15 Implements clsClassification
16
17 Private Const rcxZGPieces As String = rcxPieces
18 Private Const rcxZGCLength As String = rcxCLength
19
20 Private Sub Class_Initialize()
21     Set clsClassification = New clsClassification
22 End Sub
23
24 Private Function clsClassification Calculate(Parameter As
Scripting.Dictionary) As Double
25     Dim Pieces As String, kml As Double
26     Pieces = GetItem(Parameter, rcxZGPieces, vbNullString, True)
27     kml = Val(GetItem(Parameter, rcxZGCLength, vbNullString,
True))
28
29     clsClassification Calculate =
CalculateZG(PhrasePieces(Pieces), kml)
30 End Function
31
32 Private Property Get clsClassification ClassCount() As Long
33     clsClassification_ClassCount = 4

```

```

34 End Property
35
36 Private Function clsClassification ClassDesignation(ByVal Value
As Double) As String
37     clsClassification ClassDesignation =
clsClassification_ClassDesignationByIndex(clsClassification_Clas
sIndexByValue(Value))
38 End Function
39
40 Private Function clsClassification ClassDesignationByIndex(ByVal
Index As Long) As String
41     Dim Value As String
42
43     Select Case Index
44         Case 1: Value = "schwach"
45         Case 2: Value = "mäßig"
46         Case 3: Value = "stark"
47         Case 4: Value = "sehr stark"
48     End Select
49     clsClassification_ClassDesignationByIndex = Value
50 End Function
51
52 Private Property Get clsClassification ClassificationAutorYear()
As String
53     clsClassification_ClassificationAutorYear = "H. Molek (1983,
1984)"
54 End Property
55
56 Private Property Get clsClassification ClassificationName() As
String
57     clsClassification_ClassificationName = "Zerrüttungsgrad ZG"
58 End Property
59
60 Private Function clsClassification ClassIndexByValue(ByVal Value
As Double) As Long
61     Dim ValueBack As Long
62
63     Select Case Value
64         Case Is >= 0.08: ValueBack = 1
65         Case Is >= 0.05: ValueBack = 2
66         Case Is >= 0.02: ValueBack = 3
67         Case Else: ValueBack = 4
68     End Select
69     clsClassification ClassIndexByValue = ValueBack
70 End Function
71
72 Private Property Get clsClassification FormatString() As String
73     clsClassification_FormatString = "0.00"

```

```

74 End Property
75
76 Private Property Get clsClassification Unit() As String
77     clsClassification Unit = ""
78 End Property
79
80 Private Property Get
81     clsClassification_ValueRangeMax(Berechnungslänge As Double) As
82     Double
83     clsClassification ValueRangeMax = 1
84 End Property
85
86 Private Property Get clsClassification_ValueRangeMin() As Double
87     clsClassification_ValueRangeMin = 0
88 End Property
89
90 Private Function CalculateZG(Pieces As Pieces, kml As Double) As
91     Double
92     Dim N As Long, SummeL As Double, F As Double, m As Long
93
94     SummeL = 0: m = 0
95     For N = 1 To UBound(Pieces.Piece)
96         With Pieces.Piece(N)
97             If .Length >= 1 Then
98                 Select Case .PreservationDegree
99                     Case pdNone, pdA, pdB
100                        SummeL = SummeL + .Length
101                        m = m + 1
102                     Case Else
103                        'nix
104                 End Select
105             End If
106         End With
107     Next
108
109     If SummeL <= kml Then
110         If m = 0 Then
111             CalculateZG = 0
112         Else
113             F = SummeL / kml
114             CalculateZG = F * SummeL / (m * kml)
115         End If
116     Else
117         Err.Raise vbObjectError + 3,
118         clsClassification ClassificationName, "Die Summe der
119         eingegebenen Daten ist größer als die Kernmarschlänge.": Exit
120     End If
121 End Function

```

```

116 End Function
117
118 Public Function ZG(Pieces As String, CoreLength As Double)
119     Dim Dict As Scripting.Dictionary
120
121     Set Dict = New Scripting.Dictionary
122     Dict.Add rcxZGPieces, Pieces
123     Dict.Add rcxZGCLength, CoreLength
124
125     ZG = clsClassification Calculate(Dict)
126 End Function
127
128 Public Property Get ClassificationInterface() As
129     clsClassification
130     Set ClassificationInterface = Me
131 End Property
132
133 Private Property Get clsClassification_ParameterList() As
134     Scripting.IDictionary
135     Dim RQD As clsRQDIndex
136     Set RQD = New clsRQDIndex
137     Set clsClassification_ParameterList =
138     RQD.ClassificationInterface.ParameterList
139 End Property
140
141 Private Function clsClassification_ShowParameterWindow(Parameter
142     As String, Optional StartValue As Variant, Optional ParentWindow
143     As Object, Optional TestPresence As Boolean = False) As Variant
144     If TestPresence Then
145         clsClassification ShowParameterWindow = False
146     Else
147         clsClassification_ShowParameterWindow = StartValue
148     End If
149 End Function

```

Datei: frmParameters.frm (VB Form)

```

01 VERSION 5.00
02 Object = "{6FBA474E-43AC-11CE-9A0E-00AA0062BB4C}#1.0#0";
03 "SYSINFO.OCX"
04 Begin VB.Form frmParameters
05     BorderStyle     = 4 'Festes Werkzeugfenster
06     Caption         = "Parameter bestimmen"
07     ClientHeight    = 5775
08     ClientLeft     = 45
09     ClientTop      = 285

```

```
09 ClientWidth = 6630
10 ControlBox = 0 'False
11 Icon = "frmParameters.frx":0000
12 KeyPreview = -1 'True
13 LinkTopic = "Form1"
14 MaxButton = 0 'False
15 MinButton = 0 'False
16 ScaleHeight = 5775
17 ScaleWidth = 6630
18 ShowInTaskbar = 0 'False
19 StartUpPosition = 1 'Fenstermitte
20 Begin VB.PictureBox Picture1
21 BackColor = &H80000005&
22 Height = 4815
23 Left = 120
24 ScaleHeight = 4755
25 ScaleWidth = 6315
26 TabIndex = 1
27 Top = 360
28 Width = 6375
29 Begin SysInfoLib.SysInfo SysInfol
30 Left = 3480
31 Top = 3360
32 _ExtentX = 1005
33 _ExtentY = 1005
34 Version = 393216
35 End
36 Begin VB.VScrollBar VScroll11
37 Height = 4695
38 Left = 6000
39 TabIndex = 4
40 Top = 0
41 Width = 255
42 End
43 Begin RockClassX.ctlParameterItem ctlParameterItem1
44 Height = 735
45 Index = 0
46 Left = 120
47 Top = 120
48 Visible = 0 'False
49 Width = 5775
50 _ExtentX = 6588
51 _ExtentY = 0
52 End
53 End
54 Begin VB.CommandButton Command1
55 Cancel = -1 'True
56 Caption = "&Abbruch"
```

```
57 Height = 375
58 Index = 1
59 Left = 5040
60 TabIndex = 3
61 Top = 5280
62 Width = 1455
63 End
64 Begin VB.CommandButton Command1
65 Caption = "&Ok"
66 Default = -1 'True
67 Height = 375
68 Index = 0
69 Left = 3480
70 TabIndex = 2
71 Top = 5280
72 Width = 1455
73 End
74 Begin VB.Label Label1
75 Caption = "Label1"
76 BeginProperty Font
77 Name = "MS Sans Serif"
78 Size = 8.25
79 Charset = 0
80 Weight = 700
81 Underline = 0 'False
82 Italic = 0 'False
83 Strikethrough = 0 'False
84 EndProperty
85 Height = 255
86 Left = 120
87 TabIndex = 0
88 Top = 120
89 Width = 6375
90 End
91 End
92 Attribute VB_Name = "frmParameters"
93 Attribute VB_GlobalNameSpace = False
94 Attribute VB_Creatable = False
95 Attribute VB_PredeclaredId = True
96 Attribute VB_Exposed = False
97 Option Explicit
98
99 Public ExitValue As String
100
101 Private Sub Command1 Click(Index As Integer)
102 If Index = 0 Then
103 Dim N As Long
104
```

```
105 For N = 1 To ctlParameterItem1.UBound
106     If ctlParameterItem1(N).Selected Then
107         ExitValue = Trim(ctlParameterItem1(N).Value)
108         Exit For
109     End If
110 Next
111 Else
112     ExitValue = vbNullString
113 End If
114
115 Unload Me
116 End Sub
117
118 Private Sub ctlParameterItem1_Click(Index As Integer)
119     Dim N As Long
120     For N = ctlParameterItem1.LBound To ctlParameterItem1.UBound
121         ctlParameterItem1(N).Selected = (N = Index)
122     Next
123     Call ShowFirstSelected
124 End Sub
125
126 Public Function AddParameterItem(Value As Variant) As
ctlParameterItem
127     Dim I As ctlParameterItem, N As Long
128
129     N = ctlParameterItem1.UBound + 1
130     Load ctlParameterItem1(N)
131     Set I = ctlParameterItem1(N)
132     ctlParameterItem1(N).Visible = True
133
134     I.Value = Value
135
136     Set AddParameterItem = I
137 End Function
138
139 Private Sub Form_Activate()
140     Call VScroll1_Change
141     Call ShowFirstSelected
142 End Sub
143
144 Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
145     Dim N As Long, I As Long
146
147     For N = 1 To ctlParameterItem1.UBound
148         If ctlParameterItem1(N).Selected Then
149             I = N
150         End If
151     Next
```

```
152
153     Select Case KeyCode
154         Case vbKeyUp: I = I - 1: KeyCode = 0
155         Case vbKeyDown: I = I + 1: KeyCode = 0
156     End Select
157
158     If I < 1 Then I = 1
159     If I > ctlParameterItem1.UBound Then I =
ctlParameterItem1.UBound
160
161     For N = 1 To ctlParameterItem1.UBound
162         ctlParameterItem1(N).Selected = N = I
163     Next
164
165     Call ShowFirstSelected
166 End Sub
167
168 Private Sub VScroll1_Change()
169     Dim N As Long, T As Single
170     Dim H As Single, W As Single
171     Dim R As Single
172
173     VScroll1.Move Picture1.ScaleWidth - SysInfo1.ScrollBarSize, 0,
SysInfo1.ScrollBarSize, Picture1.ScaleHeight
174     W = Picture1.ScaleWidth - VScroll1.Width
175     T = VScroll1.Enabled * VScroll1.Value
176     R = Screen.TwipsPerPixelY * 5
177     For N = 1 To ctlParameterItem1.UBound
178         With ctlParameterItem1(N)
179             .Move 0, T, W
180             T = T + .Height + R
181             H = H + .Height + R
182         End With
183     Next
184     W = H - Picture1.ScaleHeight
185     VScroll1.Enabled = W > 0
186     If W < 0 Then W = 0
187
188     VScroll1.Min = 0
189     VScroll1.Max = W
190     VScroll1.SmallChange = Picture1.ScaleHeight / 10
191     VScroll1.LargeChange = Picture1.ScaleHeight / 3
192     Picture1.Refresh
193 End Sub
194
195 Private Sub VScroll1_Scroll()
196     VScroll1_Change
197 End Sub
```

```

198
199 Private Sub ShowFirstSelected()
200     Dim T As Single, H As Single
201     Dim N As Long
202
203     On Error Resume Next
204
205     For N = 1 To ctlParameterItem1.UBound
206         If ctlParameterItem1(N).Selected Then
207             T = ctlParameterItem1(N).Top
208             H = ctlParameterItem1(N).Height
209
210             If T < 0 Then
211                 VScroll1.Value = T - ctlParameterItem1(1).Top
212             Else
213                 If T + H > Picture1.ScaleHeight Then
214                     VScroll1.Value = (T - ctlParameterItem1(1).Top) + H -
Picture1.ScaleHeight
215                 End If
216             End If
217
218         End If
219     Next
220 End Sub
221

```

Datei: ctlParameterItem.ctl (VB User Control)

```

01 VERSION 5.00
02 Begin VB.UserControl ctlParameterItem
03     BackColor = &H80000005&
04     CanGetFocus = 0 'False
05     ClientHeight = 4230
06     ClientLeft = 0
07     ClientTop = 0
08     ClientWidth = 7785
09     ScaleHeight = 4230
10     ScaleWidth = 7785
11     Begin VB.Label Label1
12         AutoSize = -1 'True
13         BackStyle = 0 'Transparent
14         Caption = "Label1"
15         Height = 195
16         Index = 0
17         Left = 1080
18         TabIndex = 0
19         Top = 1440

```

```

20     UseMnemonic = 0 'False
21     Visible = 0 'False
22     Width = 480
23     WordWrap = -1 'True
24     End
25 End
26 Attribute VB_Name = "ctlParameterItem"
27 Attribute VB_GlobalNameSpace = False
28 Attribute VB_Creatable = True
29 Attribute VB_PredeclaredId = False
30 Attribute VB_Exposed = False
31 Option Explicit
32
33 Private Type ColType
34     Text As String
35     Bold As Boolean
36     Width As Single
37     Alignment As AlignmentConstants
38     Label As Label
39 End Type
40 Private Cols() As ColType
41 Private mySelected As Boolean
42 Private myValue As Variant
43
44 Public Event Click()
45
46 Private Sub Label1 Click(Index As Integer)
47     UserControl Click
48 End Sub
49
50 Private Sub UserControl_Click()
51     RaiseEvent Click
52 End Sub
53
54 Private Sub UserControl_Initialize()
55     Clear
56 End Sub
57
58 Public Sub Clear()
59     ReDim Cols(0 To 0)
60 End Sub
61
62 Public Sub AddCol(Text As String, Bold As Boolean, Width As
Single, Alignment As AlignmentConstants)
63     Dim N As Long
64
65     If UBound(Cols) = 0 Then
66         ReDim Cols(1 To 1)

```

```
67 Else
68     ReDim Preserve Cols(1 To UBound(Cols) + 1)
69 End If
70 With Cols(UBound(Cols))
71     .Text = Text
72     .Bold = Bold
73     .Width = Width
74     .Alignment = Alignment
75
76     N = Label1.UBound + 1
77     Load Label1(N)
78     Set .Label = Label1(N)
79     .Label.Visible = True
80 End With
81 Call UserControl Resize
82 End Sub
83
84
85 Public Property Get Selected() As Boolean
86     Selected = mySelected
87 End Property
88 Public Property Let Selected(Value As Boolean)
89     mySelected = Value
90     Call UserControl_Resize
91 End Property
92
93 Private Sub UserControl Resize()
94     Dim N As Long, L As Single, H As Single
95     L = 0: H = 0
96     For N = 1 To UBound(Cols)
97         With Cols(N)
98             .Label.Caption = .Text
99             .Label.AutoSize = True
100
101             If H < .Label.Height Then
102                 H = .Label.Height
103             End If
104
105             .Label.Move L, 0, .Width
106             .Label.Font.Bold = .Bold
107             .Label.Alignment = .Alignment
108
109             If mySelected Then
110                 .Label.ForeColor = vbHighlightText
111             Else
112                 .Label.ForeColor = vbButtonText
113             End If
114
```

```
115         L = L + .Width + Screen.TwipsPerPixelX * 2
116     End With
117 Next
118 If mySelected Then
119     BackColor = vbHighlight
120 Else
121     BackColor = vbWindowBackground
122 End If
123
124     UserControl.Height = H
125 End Sub
126
127 Public Property Get Value() As Variant
128     Value = myValue
129 End Property
130 Public Property Let Value(ByVal vNewValue As Variant)
131     myValue = vNewValue
132 End Property
133
```

Anlage VII: Manuelles Einbinden der Komponente RockClassX

Um die Funktionen der Komponente RockClassX manuell einzubinden oder andere Auswerteverfahren für Excel-Tabellen nutzbar zu machen, müssen die im folgenden dargestellten Schritte ausgeführt werden. Dazu ist die Nutzung der bei MS Excel mitgelieferten Entwicklungsumgebung für Visual Basic vonnöten.

Der Aufruf der Entwicklungsumgebung erfolgt - wie in Abbildung 1 gezeigt - aus dem „Extras“-Menü von MS Excel.

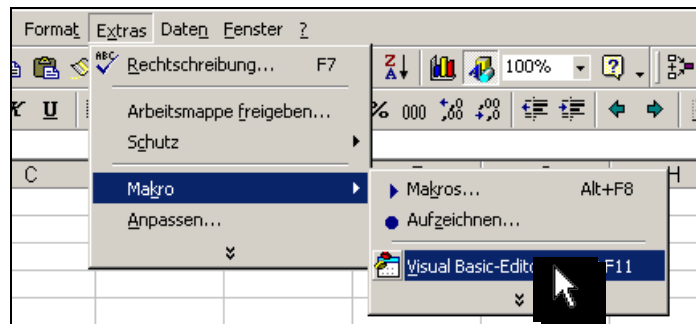


Abbildung 1: Aufruf des Visual Basic-Editors

Im Menü „Extras“ der Microsoft Visual Basic - Entwicklungsumgebung kann über den Befehl „Verweise...“ nun eine Verbindung zwischen dem VBAProject und der ActiveX-Komponente hergestellt werden. Dazu wird das Kontrollkästchen vor dem Eintrag „Gebirgsklassifikation ActiveX-DLL (Studienarbeit Jörg Meier, 2001)“²⁰ aktiviert und das Dialogfeld mit „Ok“ geschlossen (vgl. Abbildung 2).

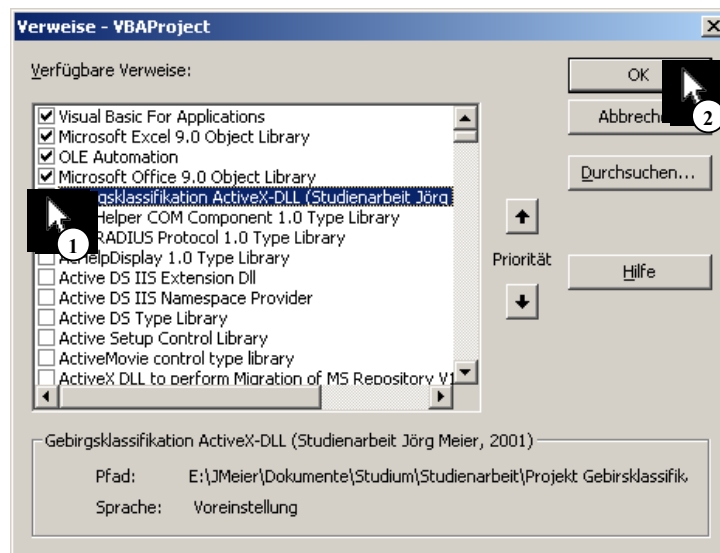


Abbildung 2: Einstellen des Verweises auf die ActiveX-Komponente

²⁰ Dieser Eintrag ist nur verfügbar, falls die ActiveX-Komponente „RockClassX“ korrekt installiert wurde. (vgl. Abschnitt 4.3.2 auf Seite 67)

Um eine klare Trennung zwischen dem nötigen Quellcode, den Arbeitsblättern („Worksheets“) und ggf. vorhandenem anderen Quellcode zu schaffen wird empfohlen, ein zusätzliches Standardmodul²¹ in das Projekt einzufügen (Menü „Einfügen“, Befehl „Modul“).

Für jede Klassifikation muss nun eine Funktion geschrieben werden, welcher die erforderlichen Parameter übergeben werden und welche ihrerseits wiederum das Ergebnis in dem gewünschten Format zurückgibt. Als genereller Aufbau der Einzelschritte dieser „Übermittlerfunktion“ kommen zwei Möglichkeiten in Betracht. Innerhalb der Funktion müssen folgende Schritte ausgeführt werden:

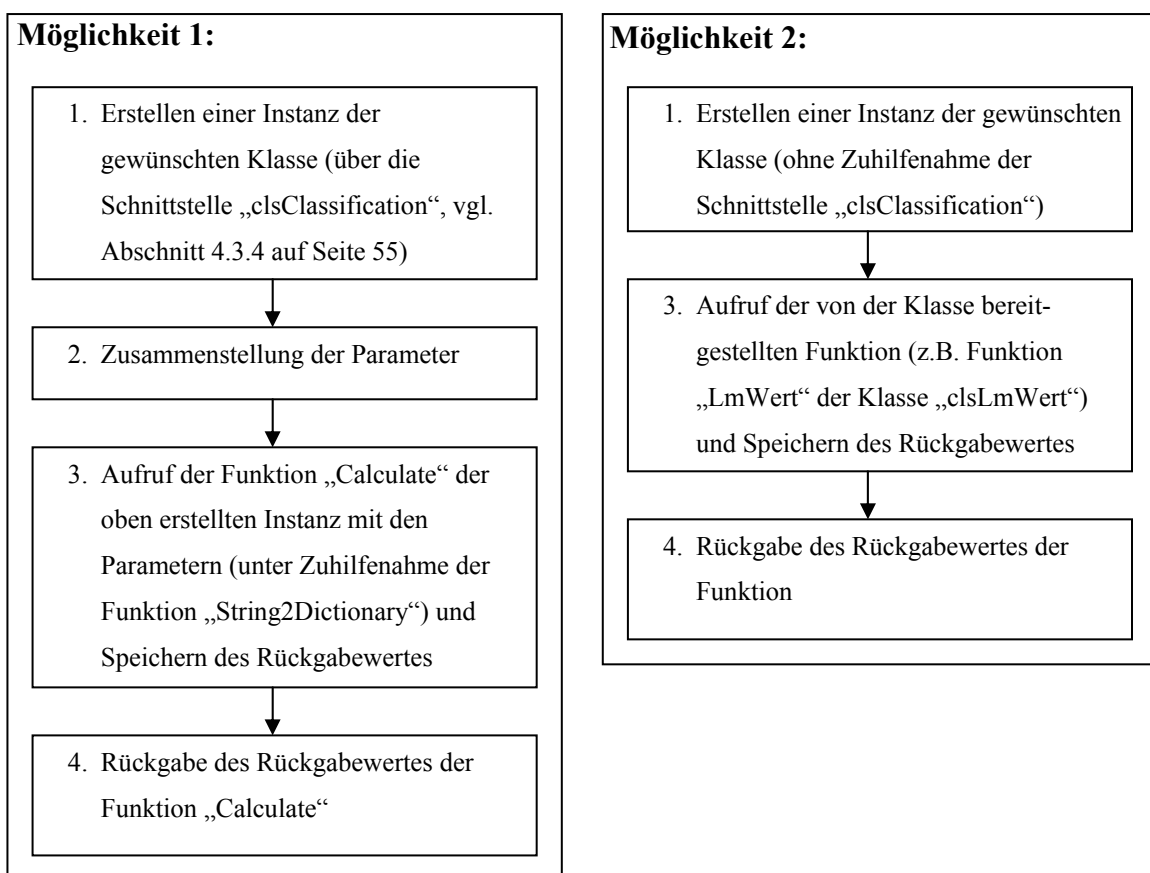


Abbildung 3: Genereller Aufbau der Übermittlerfunktion für das Einbinden der RockClassX-Komponente

Abbildung 4 und Abbildung 5 zeigen die programmtechnische Umsetzung einer solchen Übermittlerfunktion. Die so geschriebene Funktion kann wie in Kapitel 4.3.3 beschrieben in der Excel-Tabelle genutzt werden.

²¹ Ein Modul, das nur Prozeduren, Typen, Datendeklarationen und Datendefinitionen enthält.

```

Public Function GetLmWert(Pieces As String, Kernmarsch As Double, _
                        Optional Klassegröße As Double = 5) As Variant

    Dim Value As Variant 'Variable für den Rückgabewert

    'eine neue Instanz der RockClassX deklarieren und erstellen (diese
    'Instanz wird nach verlassen der Funktion automatisch von VBA zerstört)
    Dim RockClass As RockClassX.clsClassification
    Set RockClass = New RockClassX.clsLmWert

    'Fehlerbehandlung aktivieren
    On Error Resume Next 'beim Auftreten eines Fehlers diesen ignorieren
    Err.Clear           'eventuellen, letzten Fehlercode löschen

    'Parameter zusammenstellen
    Dim ParameterString As String
    ParameterString = "pieces=" & Pieces & " " & _
                    "kernmarsch=" & Kernmarsch & " " & _
                    "klasse=" & Klassegröße

    'Wert berechnen
    Value = RockClass.Calculate(RockClassX.String2Dictionary(ParameterString))

    'falls ein Fehler auftritt, diesen ausgeben
    If Err <> 0 Then
        Value = Err.Description & " (Quelle: " & Err.Source & _
                ", Fehlercode=&H" & Hex(Err.Number) & ")"
    End If

    GetLmWert = Value 'Wert zurückgeben
End Function

```

Zeichenerklärung:

Quelle.....	normaler Text
Variant.....	Schlüsselwörter
GetLmWert.....	Bezeichner
'eine neue In.....	Kommentare, alle Eingaben nach einem Hochkomma werden ignoriert

Abbildung 4: Quellcode für das Einbinden der RockClassX-Komponente

```

Public Function GetLmWert(Pieces As String, Kernmarsch As Double, _
                        Optional Klassegröße As Double = 5) As Variant

    Dim Value As Variant 'Variable für den Rückgabewert

    'eine neue Instanz der RockClassX deklarieren und erstellen (diese
    'Instanz wird nach verlassen der Funktion automatisch von VBA zerstört)
    Dim RockClass As RockClassX.clsLmWert
    Set RockClass = New RockClassX.clsLmWert

    'Fehlerbehandlung aktivieren
    On Error Resume Next 'beim Auftreten eines Fehlers diesen ignorieren
    Err.Clear           'eventuellen, letzten Fehlercode löschen

    'Wert berechnen
    Value = RockClass.LmWert(Pieces, Kernmarsch, Klassegröße)

    'falls ein Fehler auftritt, diesen ausgeben
    If Err <> 0 Then
        Value = Err.Description & " (Quelle: " & Err.Source & _
                ", Fehlercode=&H" & Hex(Err.Number) & ")"
    End If

    GetLmWert = Value 'Wert zurückgeben
End Function

```

Abbildung 5: alternativer Quellcode für das Einbinden der RockClassX-Komponente

Anlage VIII: Code für Generierung aller Bruchstückkombinationen eines Kernmarsches

Diese Anlage enthält den Quellcode für die Generierung aller möglichen Kombinationen von Kernbruchstücken in einem Kernmarsch.

Dieses Programm erzeugt für eine gegebene Kernmarschlänge („Kernmarschlänge“) und ein gegebenes Raster („Schrittweite“) alle möglichen Kombinationen von Kernbruchstücken und weist diese einer Datenbank zu. Bei der Generierung werden die Kernbruchstücken der Größe nach sortiert. Dies geschieht zum einen, um doppelte Fälle auszuschließen und zum andern wird die Sortierung der Kernbruchstücke innerhalb eines Kernmarsches (Berechnungslänge) von keinem der betrachteten Verfahren beachtet. Abbildung zeigt das laufende Programm:

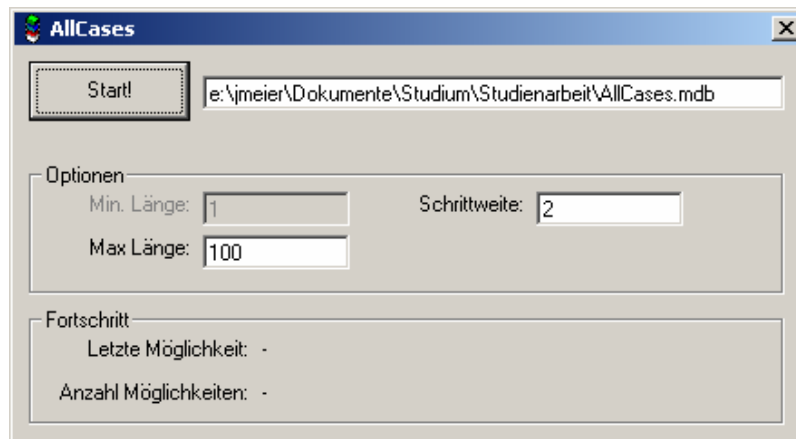


Abbildung: Programm „AllCases“

Bei der Erzeugung sollte darauf geachtet werden, dass der Wert für die Schrittweite nicht zu klein gegenüber dem Wert der Kernmarschlänge gewählt wird, da sonst zum einen die Kapazität der Datenbank leicht überschritten wird (maximale Datenbankgröße: 2 GB) und zum anderen die Erzeugung leicht mehrere Tage in Anspruch nehmen kann. Die Erzeugung der Kombinationen für einen 1,0 m langen Kernmarsch bei einer Schrittweite von 2 cm benötigte auf einem PIII 600 MHz ca. 2,5 h reine Rechenzeit und erzeugte 1 295 971 Kombinationen; die Datenbank für eine Schrittweite von 1 cm konnte nicht vollständig erzeugt werden, da die Maximalgröße der Datenbank (2 GB) überschritten worden wäre (vgl. Anlage IX: „Anzahl möglicher Kernbruchstückkombinationen“).

Es wird von dem Programm vorausgesetzt, dass an der angegebenen Stelle eine Datenbank mit einer Tabelle „Cases“ und den Feldern „Piesces“ (String), „LmWert“ (Zahl vom Typ Double), „RQDIndex“ (Zahl vom Typ Double) und „ZG“ (Zahl vom Typ Double) besteht.

Aus Platzgründen wurde nur der Quellcode abgedruckt, der zur eigentlichen Erzeugung notwendig ist. Der Aufruf der Funktion „GeneratePieces“ führt die Generierung aus:

Quellcode: Generierung aller Bruchstückkombinationen eines Kernmarsches

```
01 Attribute VB Name = "basAllCases"
02 Option Explicit
03
04 Private CasesCount As Long
05 Public Recordset As DAO.Recordset
06
07 Public Function GeneratePieces(ByVal L As Long, ByVal LastN As Long, ByVal pre As
String, ByVal KMLMin As Long, ByVal KMLStp As Long) As Long
08     AddCase pre 'Fall hinzufügen
09     If L <> 0 Then
10         Dim N As Long, A As Long
11
12         If L > LastN Then
13             A = LastN
14         Else
15             A = L
16         End If
17         For N = A To KMLMin Step -KMLStp
18             GeneratePieces L - N, N, pre & "/" & N, KMLMin, KMLStp
19         Next
20     End If
21
22     GeneratePieces = CasesCount
23 End Function
24
25 Private Sub AddCase(ByVal strCase As String)
26     Static clsLmWert As RockClassX.clsLmWert
27     Static clsRQDIndex As RockClassX.clsRQDIndex
28     Static clsZerrüttungsgradZG As RockClassX.clsZerrüttungsgradZG
29
30     If strCase <> vbNullString Then
31
32         If Left(strCase, 1) = "/" Then
33             strCase = Mid(strCase, 2)
34         End If
35
36         If clsLmWert Is Nothing Then Set clsLmWert = New RockClassX.clsLmWert
37         If clsRQDIndex Is Nothing Then Set clsRQDIndex = New RockClassX.clsRQDIndex
38         If clsZerrüttungsgradZG Is Nothing Then Set clsZerrüttungsgradZG = New
RockClassX.clsZerrüttungsgradZG
39
40         With Recordset
41             .AddNew
42             .Fields("Pieces") = strCase
43
44             .Fields("LmWert") = clsLmWert.LmWert(strCase, 100)
45             .Fields("LmWert theoretisch") = clsLmWert.LmWert(strCase, 100, 0)
46             .Fields("RQDIndex") = clsRQDIndex.RQDIndex(strCase, 100)
47             .Fields("ZG") = clsZerrüttungsgradZG.ZG(strCase, 100)
48             .Update
49         End With
50         CasesCount = CasesCount + 1
51
52         Static tmr As Single
53         If Abs(tmr - Timer) > 0.5 Then
54             Form1.Label2.Caption = strCase
55             Form1.Label4.Caption = Format(CasesCount, "#,##0")
56             Form1.Refresh
57             tmr = Timer
58         End If
59     End If
60 End Sub
61
```

Anlage IX: Anzahl möglicher Kernbruchstückkombinationen

Für die Untersuchung des Verhaltens der Kernauswertverfahren kann es von Interesse sein, aus der theoretisch unendlichen Vielfalt der möglichen Kernbruchstückkombinationen mit und ohne Kernverlusten eine repräsentative Untermenge zu untersuchen (vgl. Anlage VIII: „Code für Generierung aller Bruchstückkombinationen eines Kernmarsches“).

Dabei wird davon ausgegangen, dass der Bohrkern nur an ausgezeichneten Stellen gleichverteilt über den Berechnungsbereich brechen kann (minimaler Bruchabstand). Für die Berechnung der Anzahl der aus diesen Gegebenheiten resultierenden Kernbruchstückkombinationen einer Berechnungslänge konnte keine arithmetische Gleichung gefunden werden. Um die Anzahl dennoch bestimmen zu können, kann der Quellcode genutzt werden.

Abbildung zeigt den Verlauf der Anzahl der Anzahl der Möglichen über dem minimalen Bruchabstand für eine Berechnungslänge vom 1,0 m. Der durch die Trendlinie gekennzeichnete (theoretische) Pfad der Anzahl der Möglichkeiten entspricht nicht dem exakten Verlauf, da dieser nicht stetig erfolgt, nähert ihn aber an.

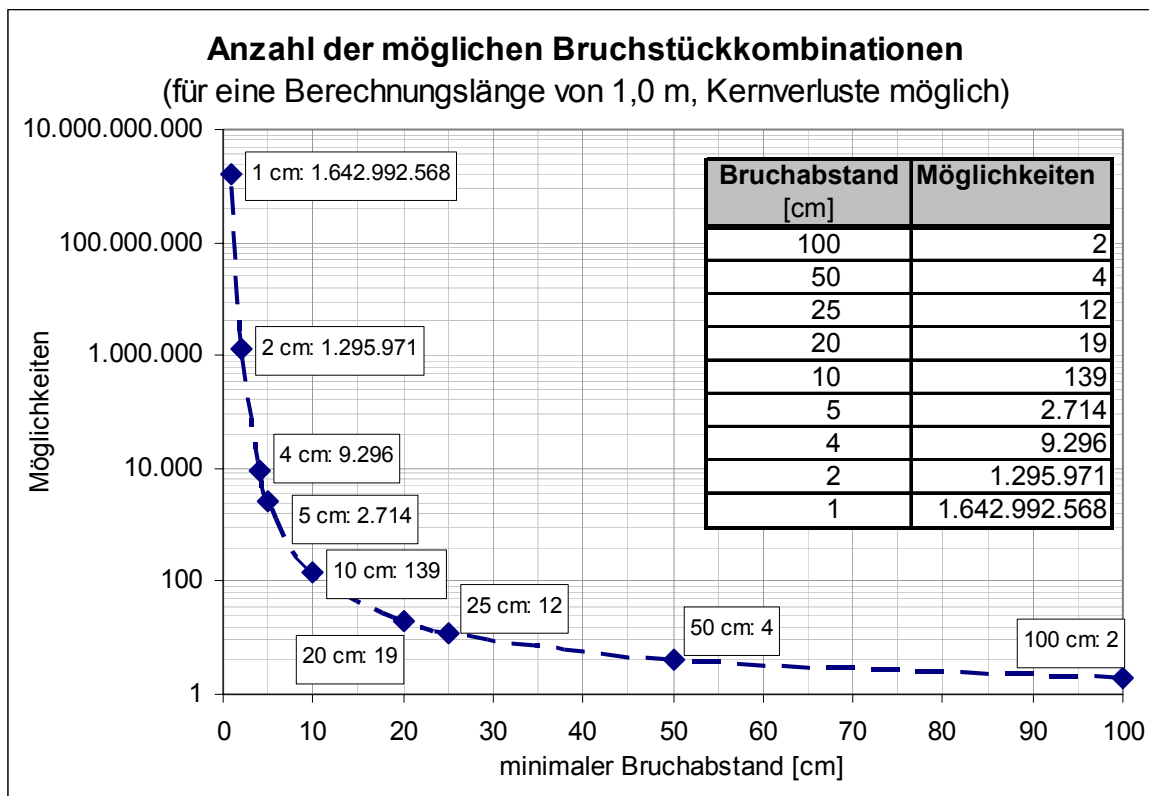


Abbildung: Anzahl möglicher Kernbruchstückkombinationen

Der in Quellcode aufgezeigte Algorithmus benötigt vor allem für sehr kleine Bruchabstände relativ hohe Rechenleistung. Der Quellcode gilt ausschließlich für ganzzahlige Teiler der Berechnungslänge und arbeitet nur mit natürlichen Zahlen (für die Berechnungslänge als auch für den Bruchabstand). Der Algorithmus wird über den Aufruf der Funktion „GetCaseCount“ gestartet und erwartet als Parameter die Berechnungslänge und den minimalen Bruchstückabstand.

Quellcode: Anzahl möglicher Kernbruchstückkombinationen

```
01 Private CasesCount As Long
02
03 Public Function GetCaseCount(Berechnungslänge As Long, BruchAlle As Long)
04     CasesCount = 0
05     GetCaseCount = GeneratePieces(Berechnungslänge, Berechnungslänge, 1, BruchAlle)
06 End Function
07 Private Function GeneratePieces(ByVal L As Long, ByVal LastN As Long, ByVal KMLMin
As Long, ByVal KMLStp As Long) As Long
08     Dim N As Long, A As Long
09
10     CasesCount = CasesCount + 1 'Fall hinzufügen
11
12     If L <> 0 Then
13
14         If L > LastN Then
15             A = LastN
16         Else
17             A = L
18         End If
19         For N = A To KMLMin Step -KMLStp
20             GeneratePieces L - N, N, KMLMin, KMLStp
21         Next
22     End If
23
24     GeneratePieces = CasesCount
25 End Function
```

Anlage X: Korrelation L_m -Wert, RQD-Index und Zerrüttungsgrad Z_G

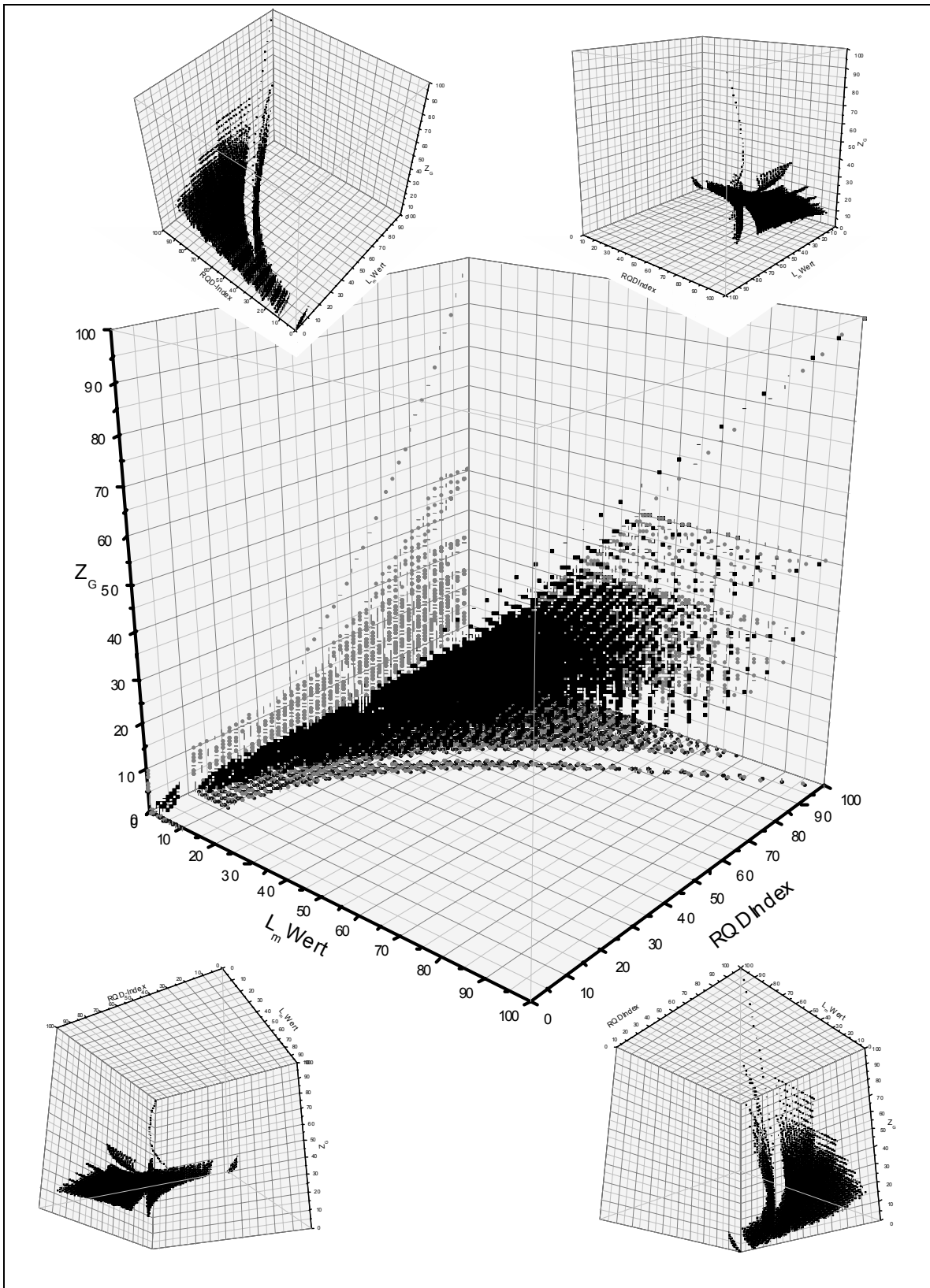
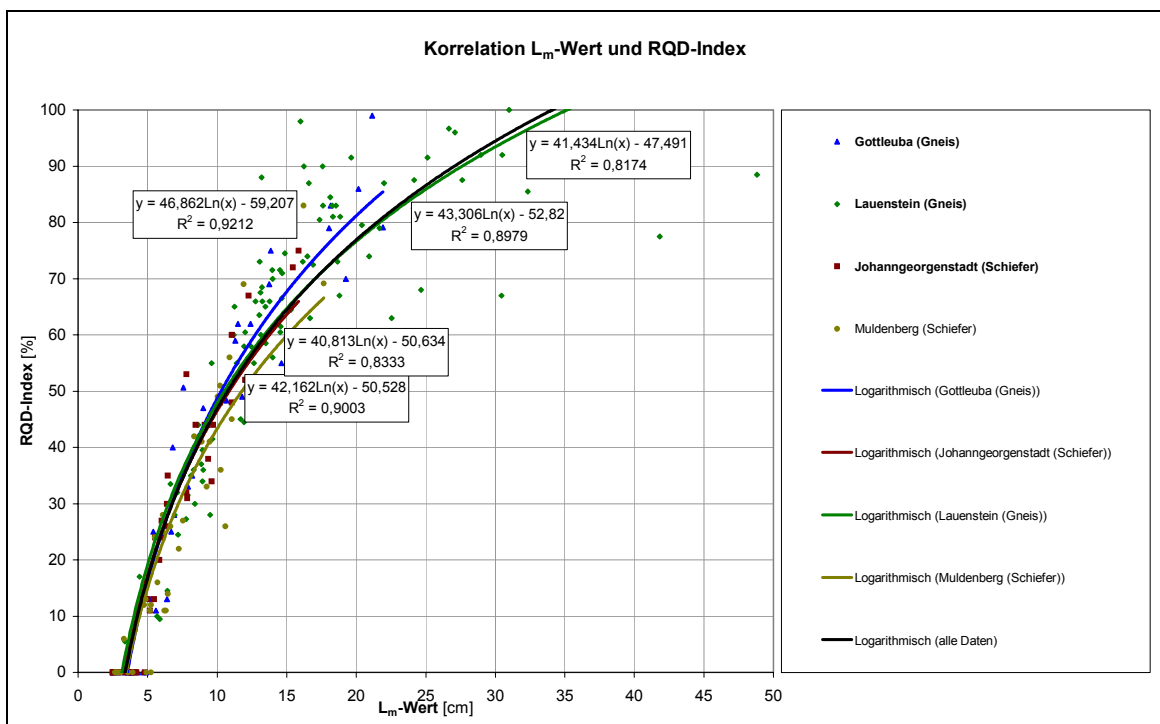
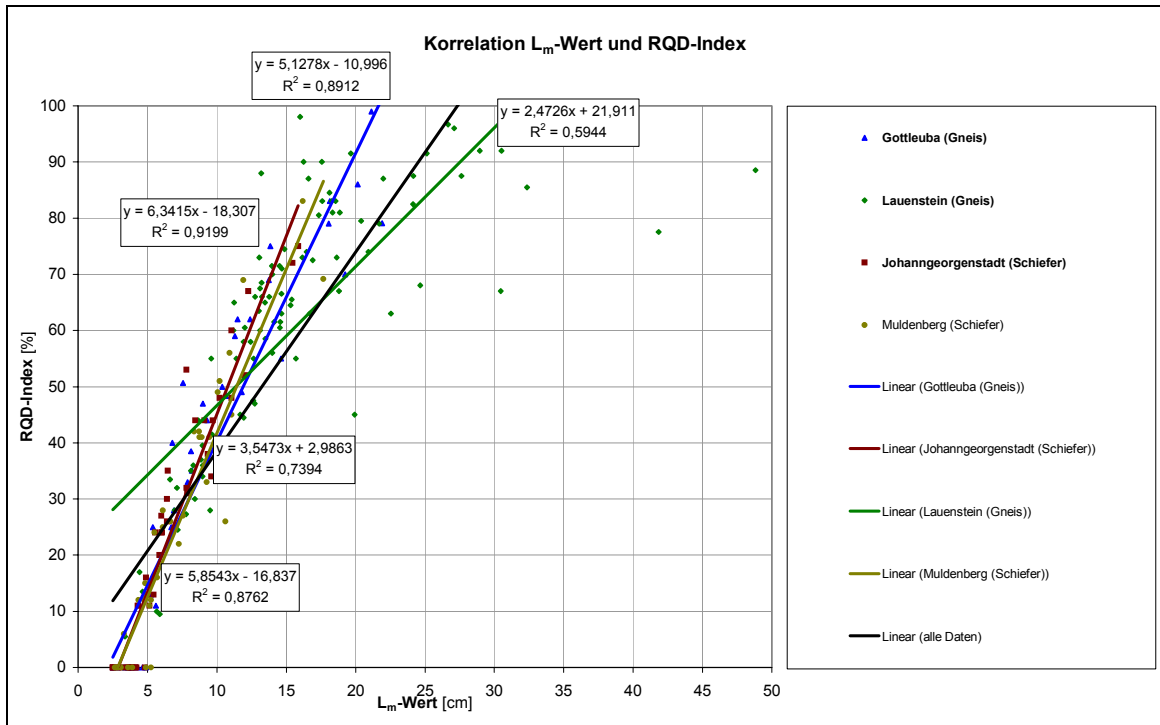
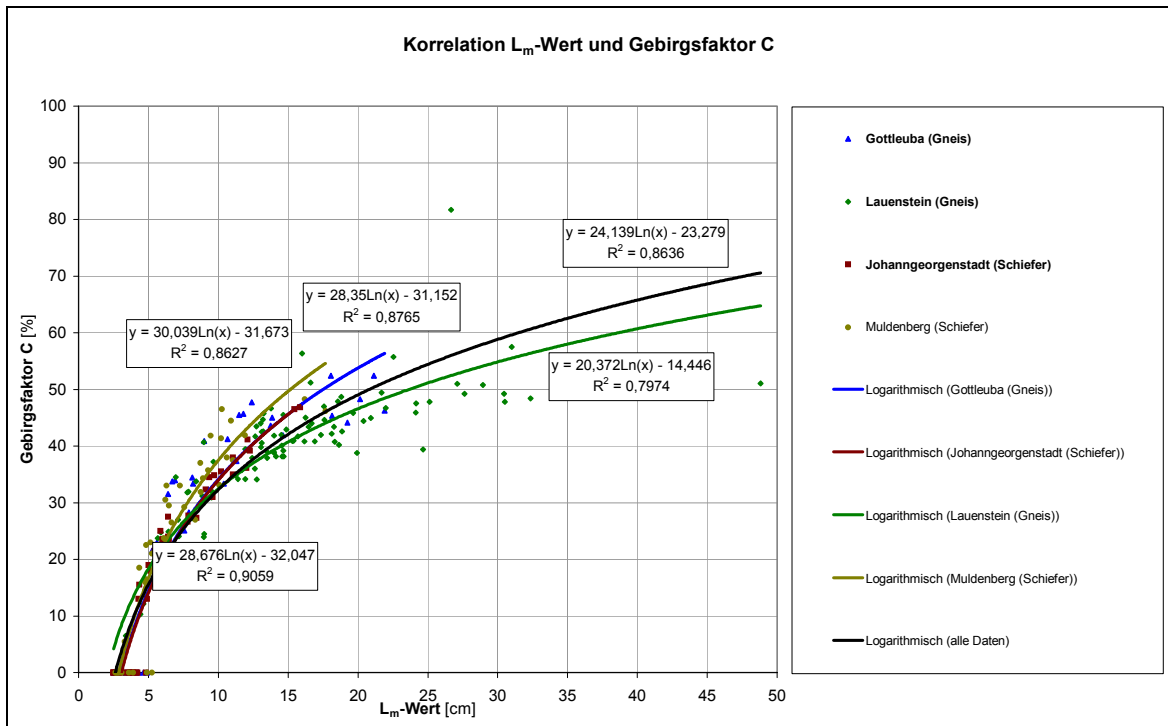
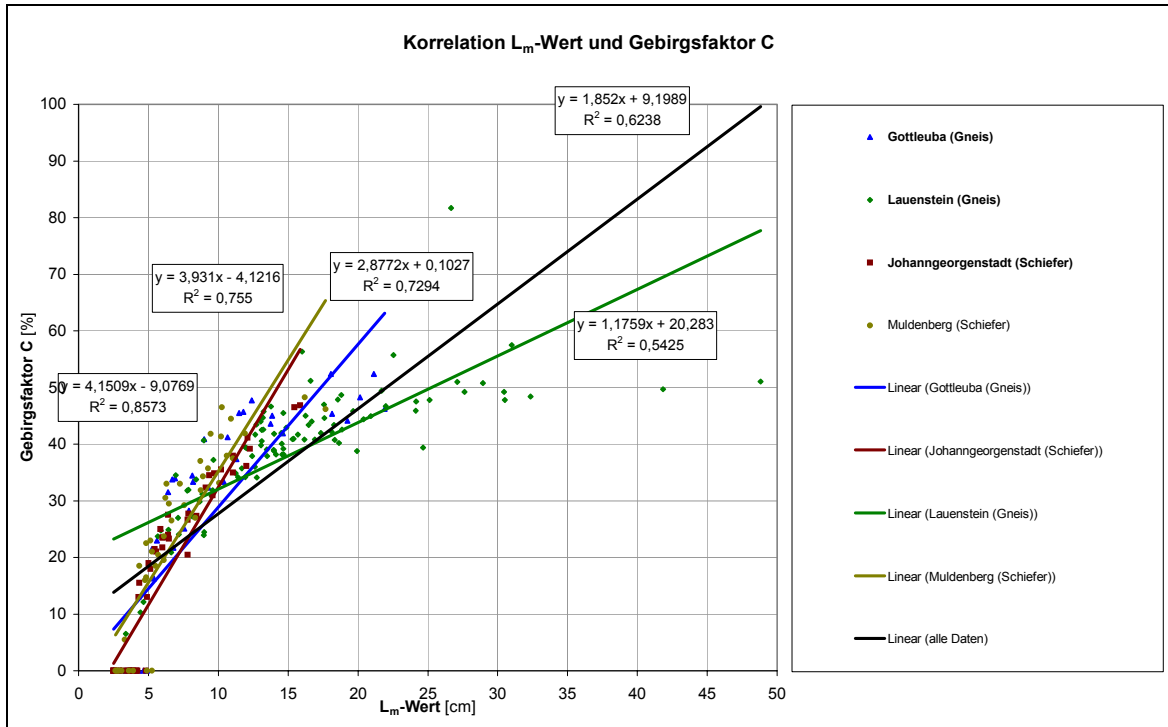
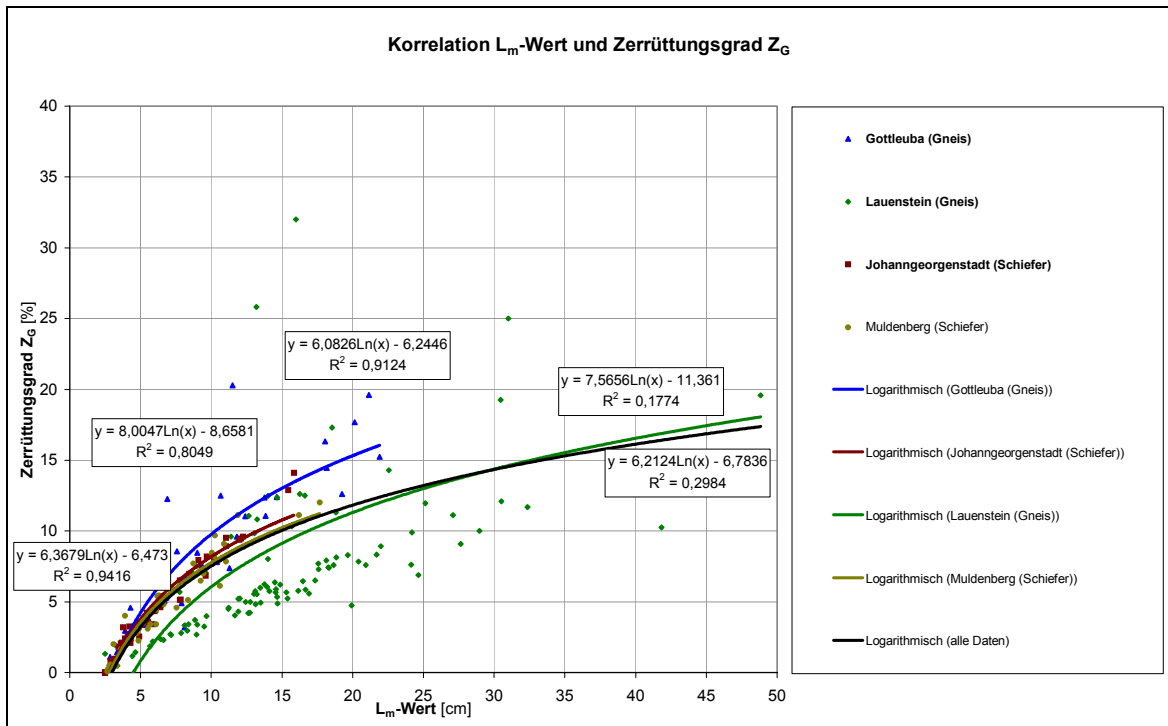
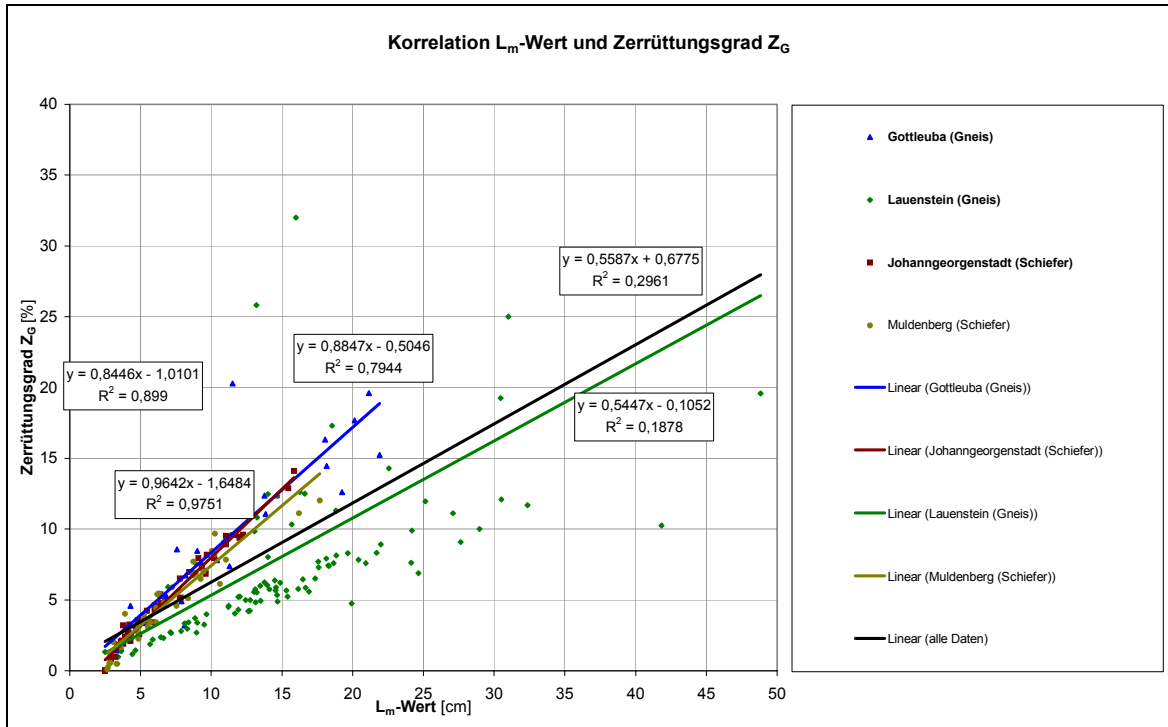


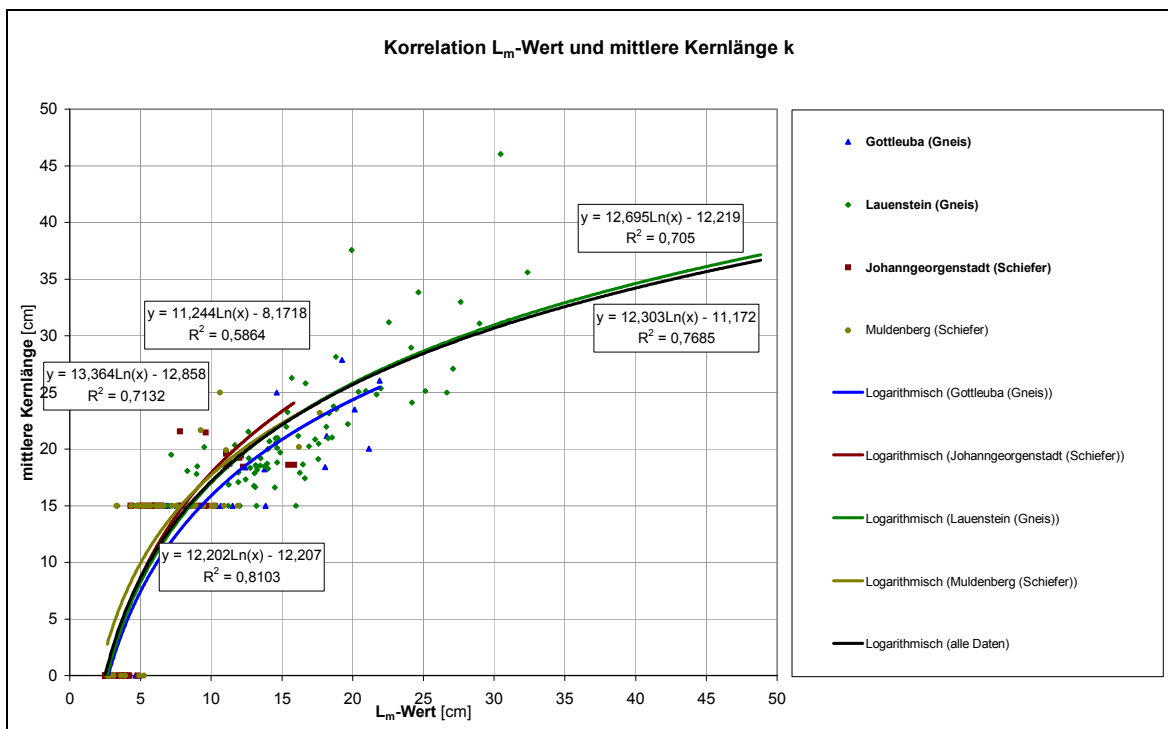
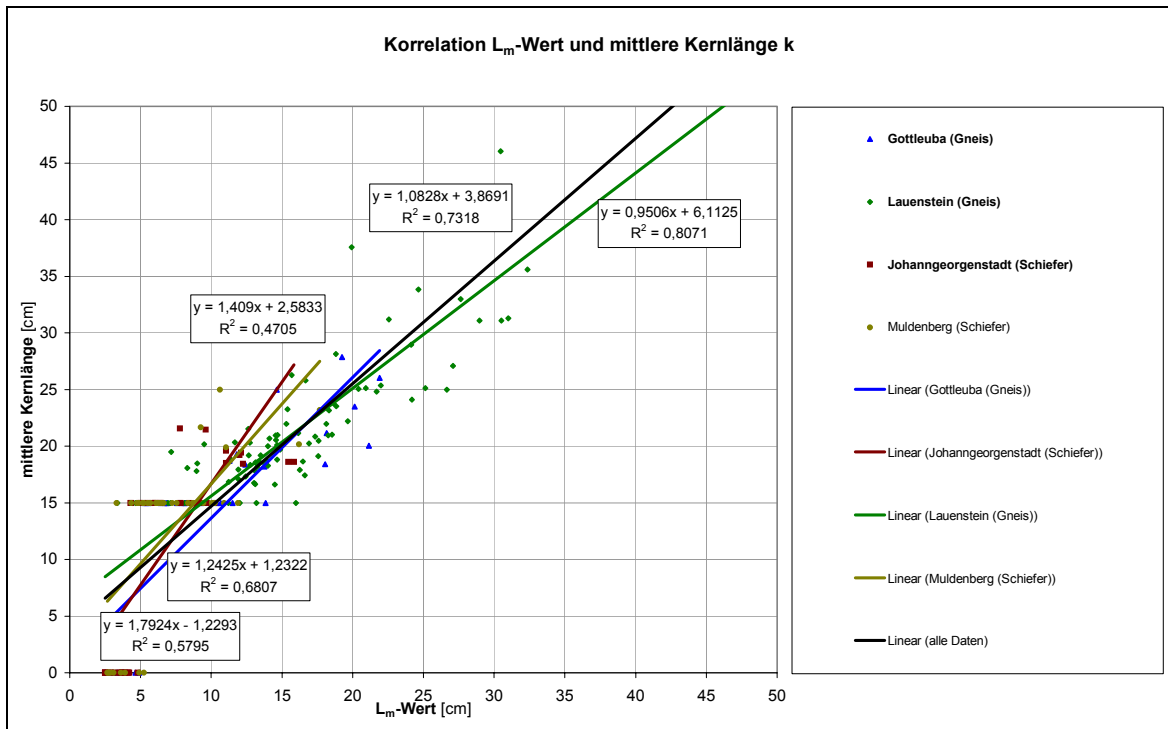
Abbildung: Verteilung der Datentripel L_m -Wert, RQD-Index und Zerrüttungsgrad Z_G

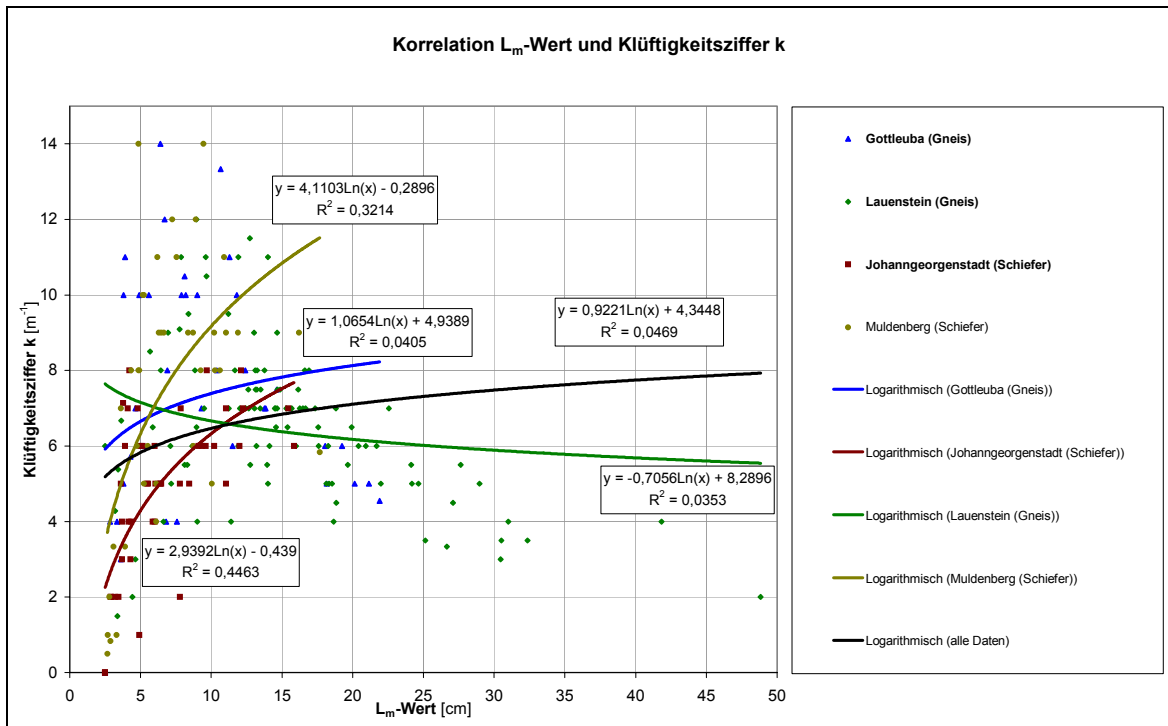
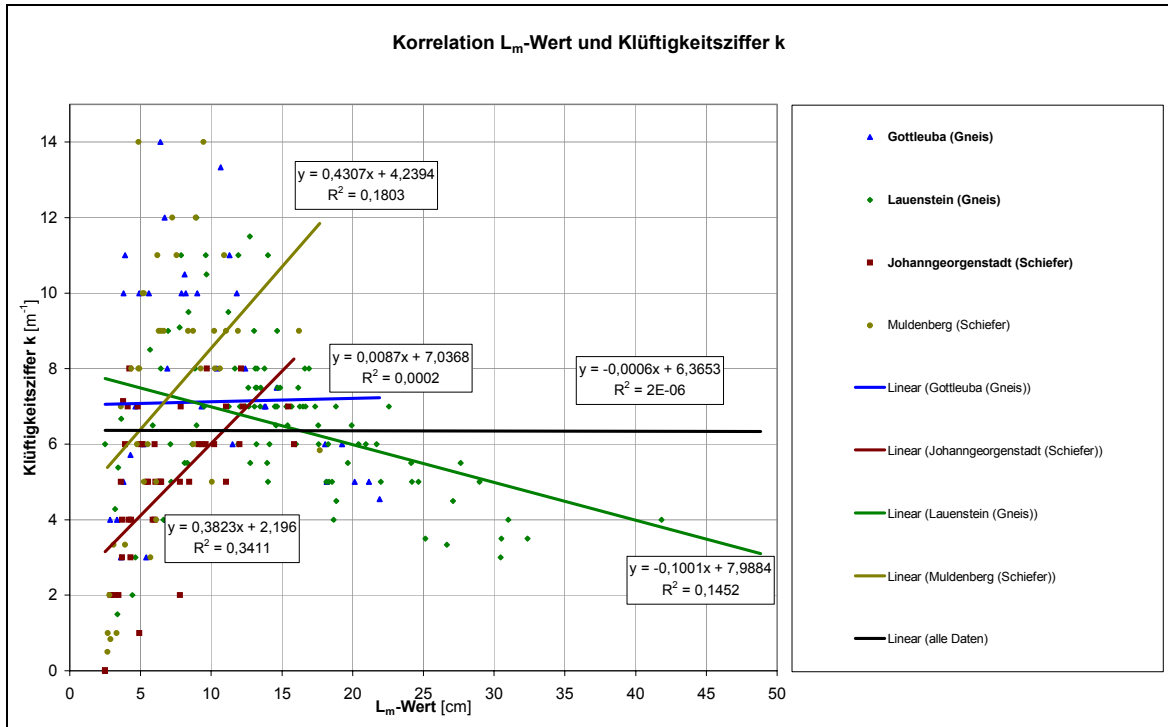
Anlage XI: Vergleich der Auswerteverfahren (Diagramme)

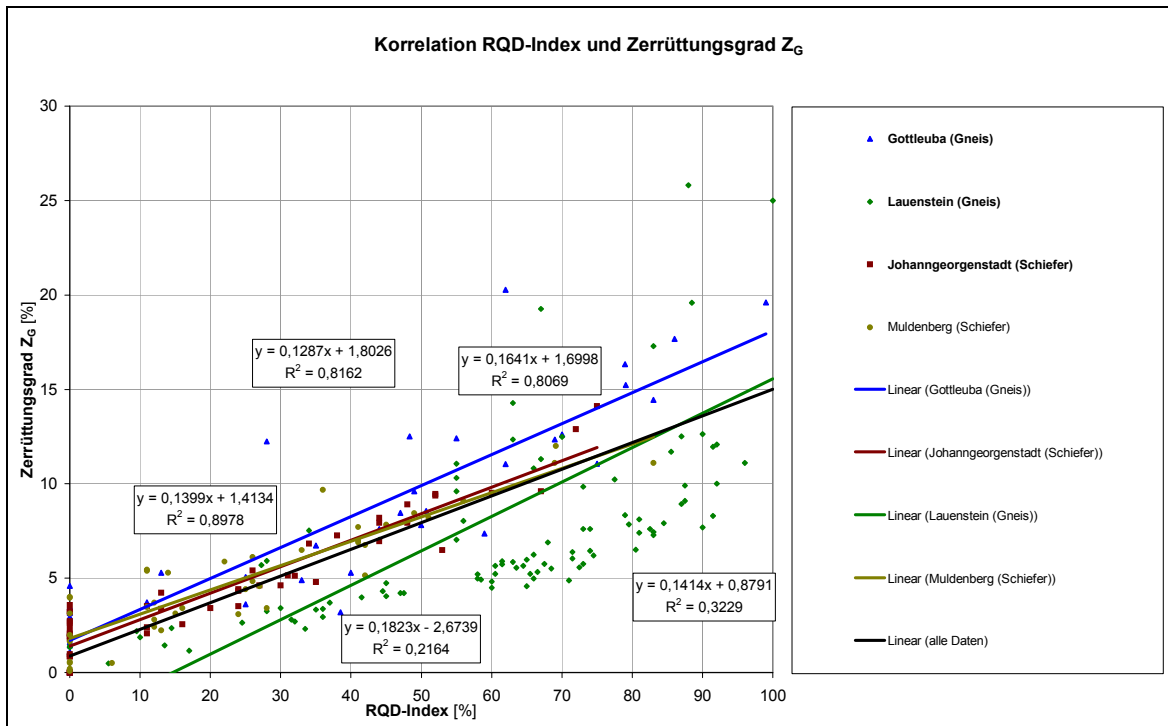
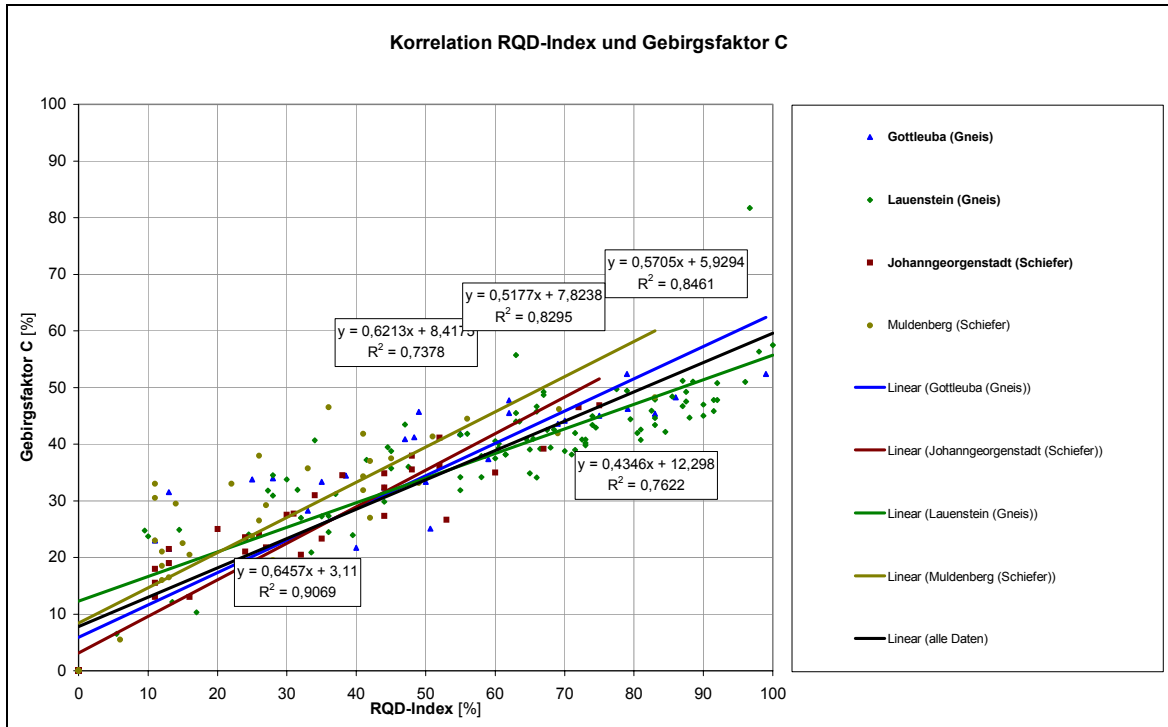


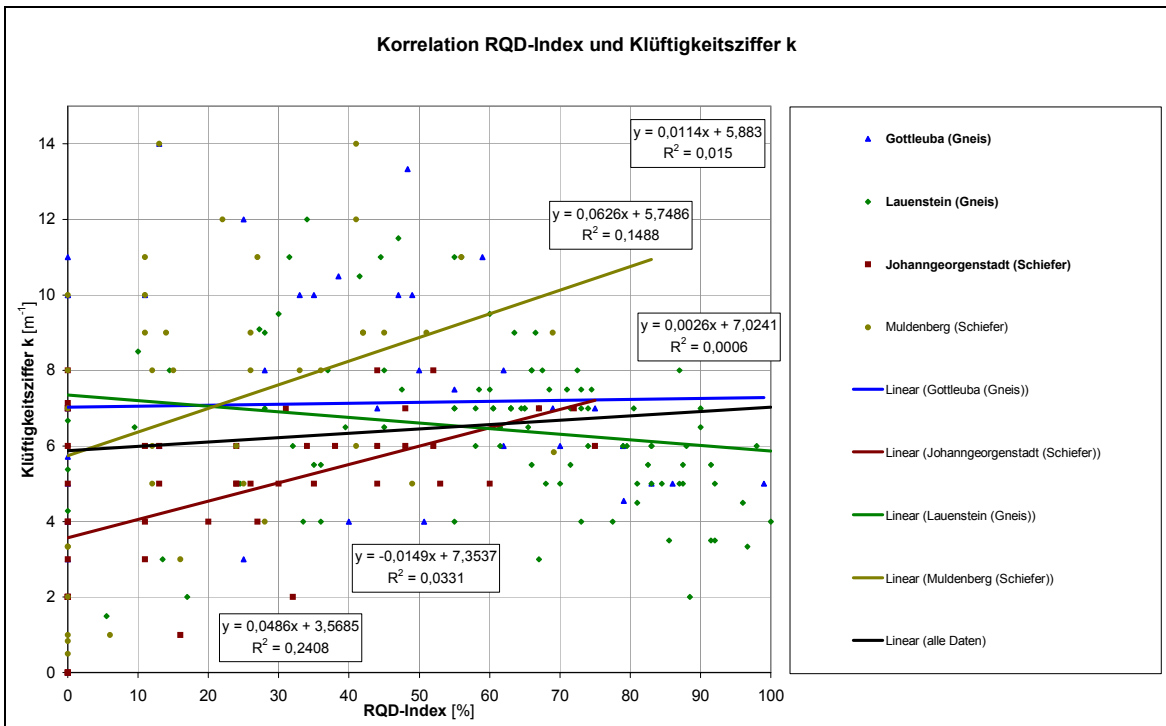
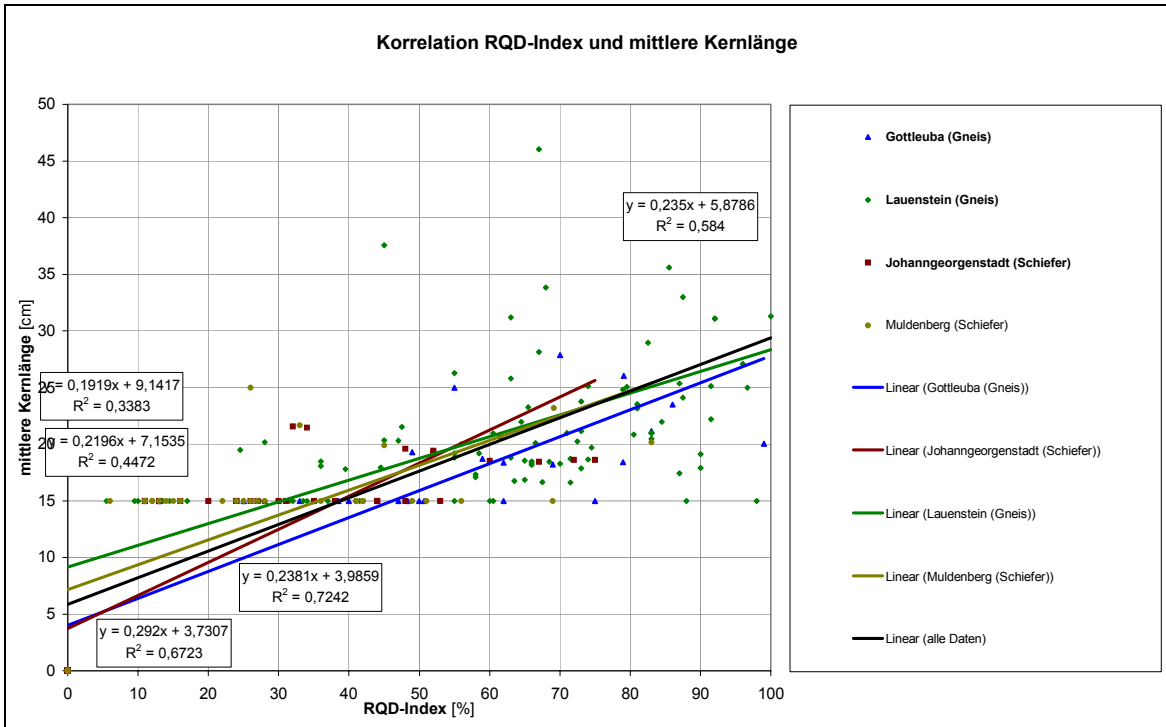


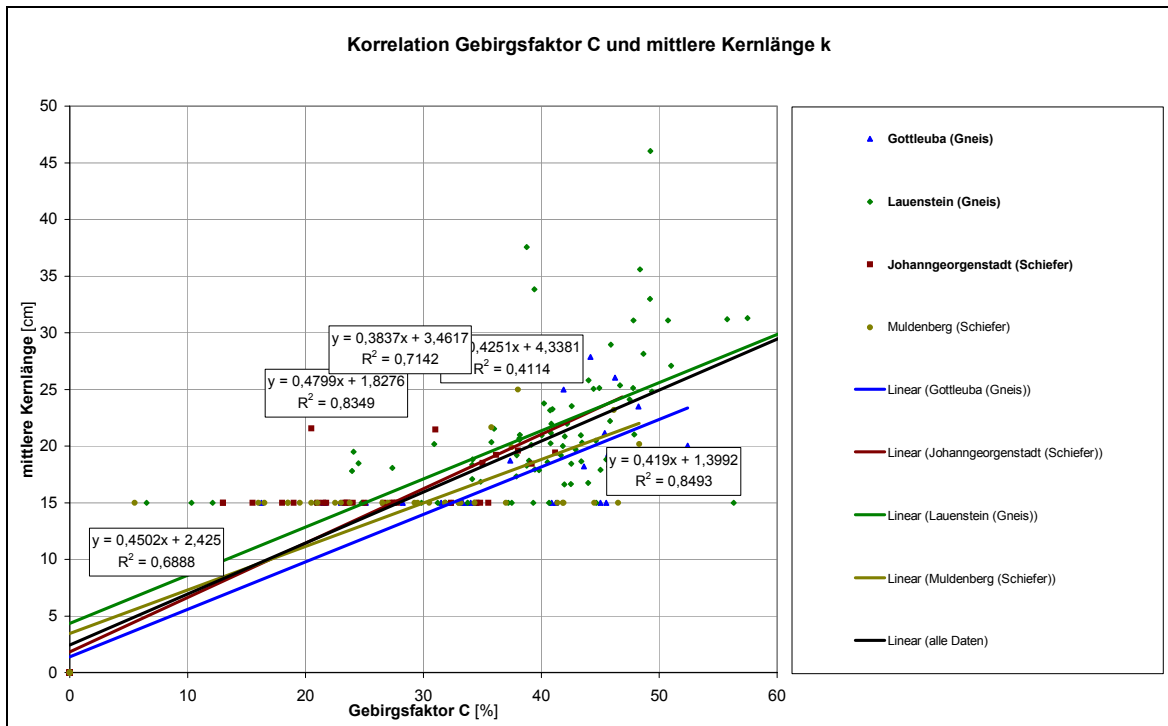
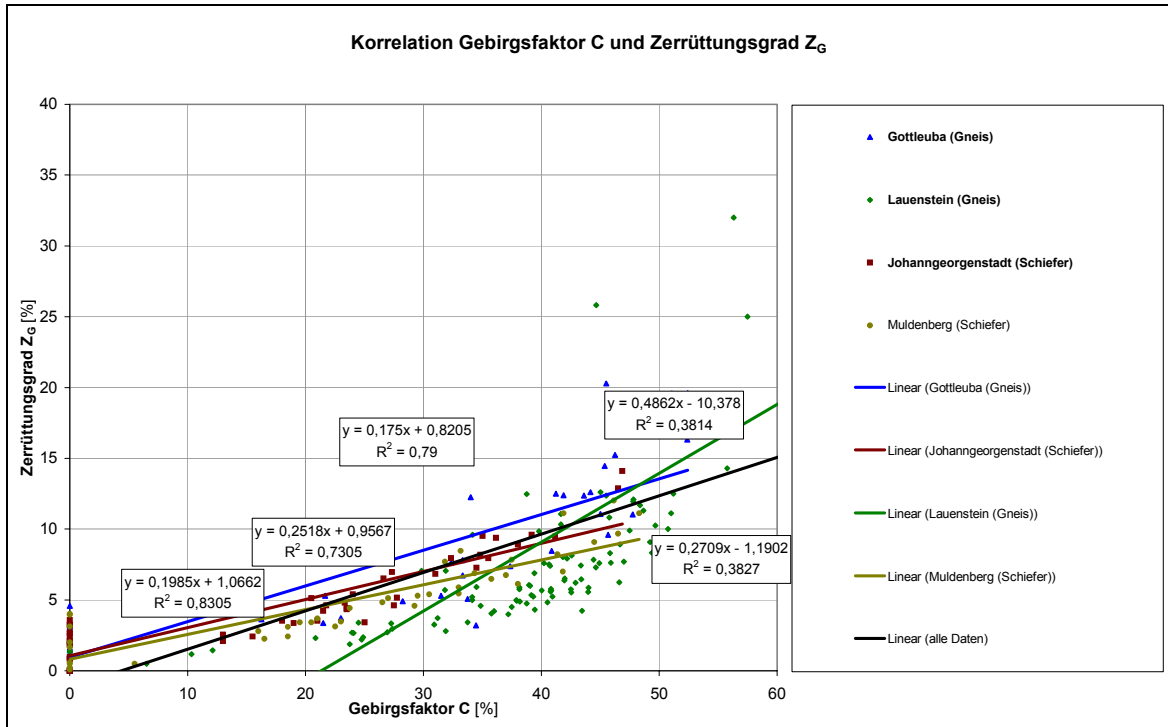


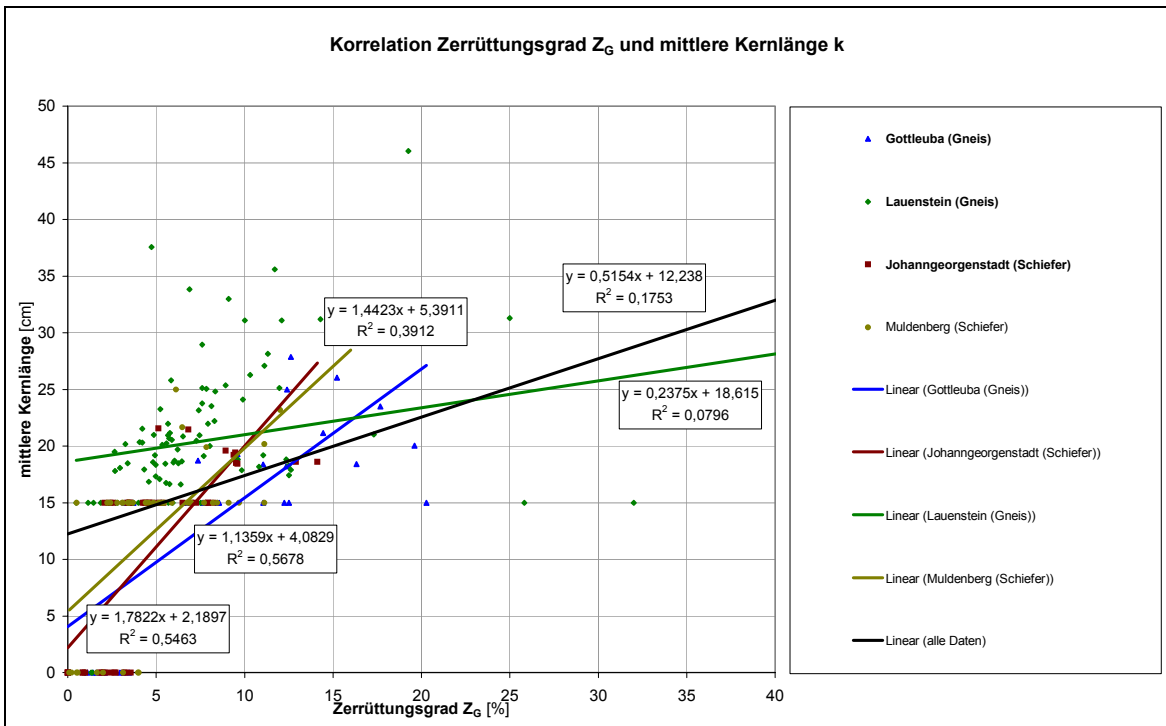
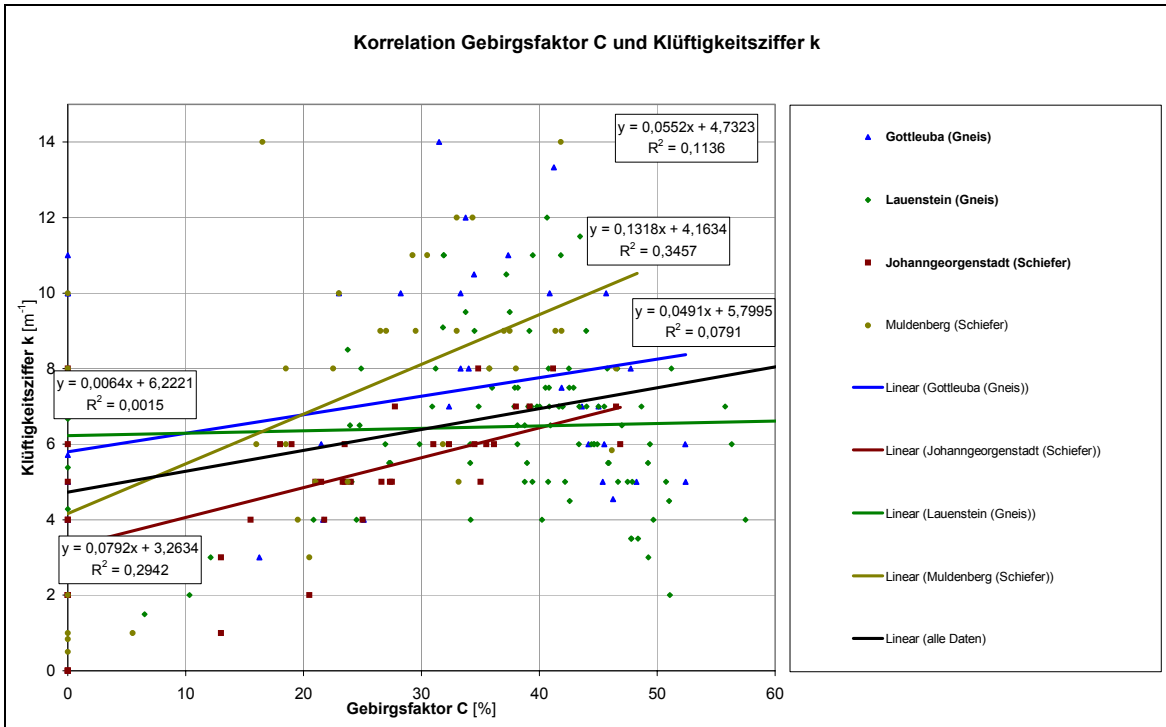


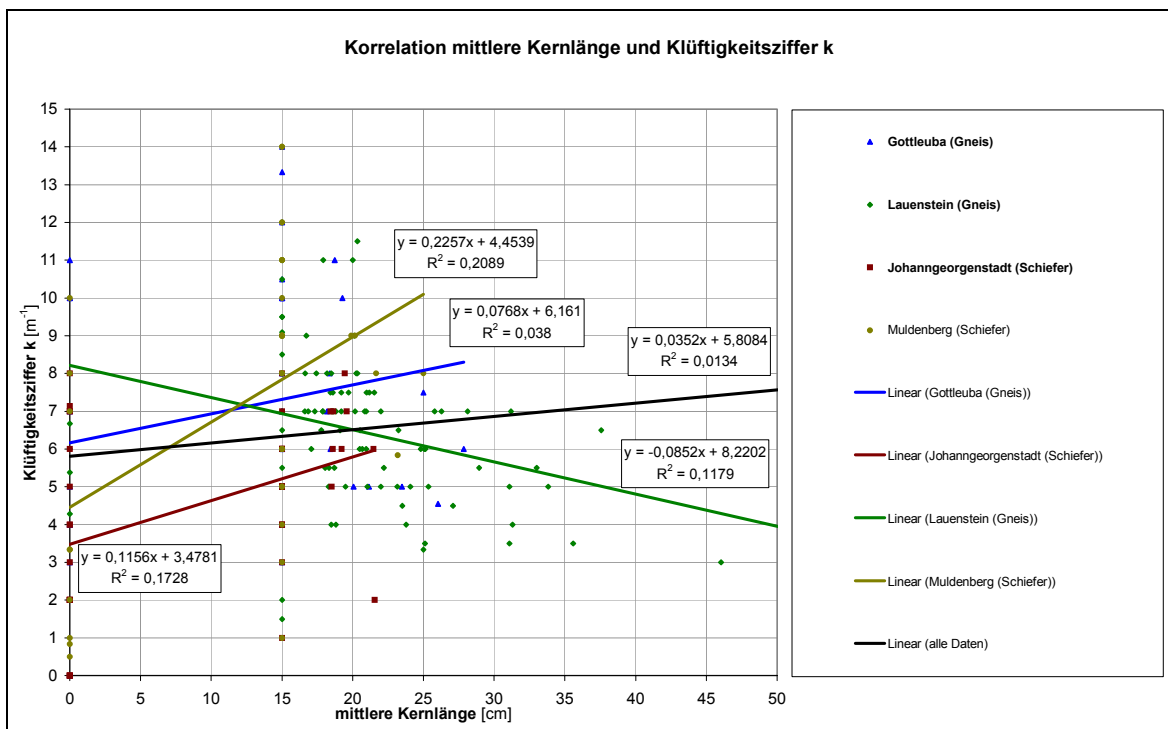
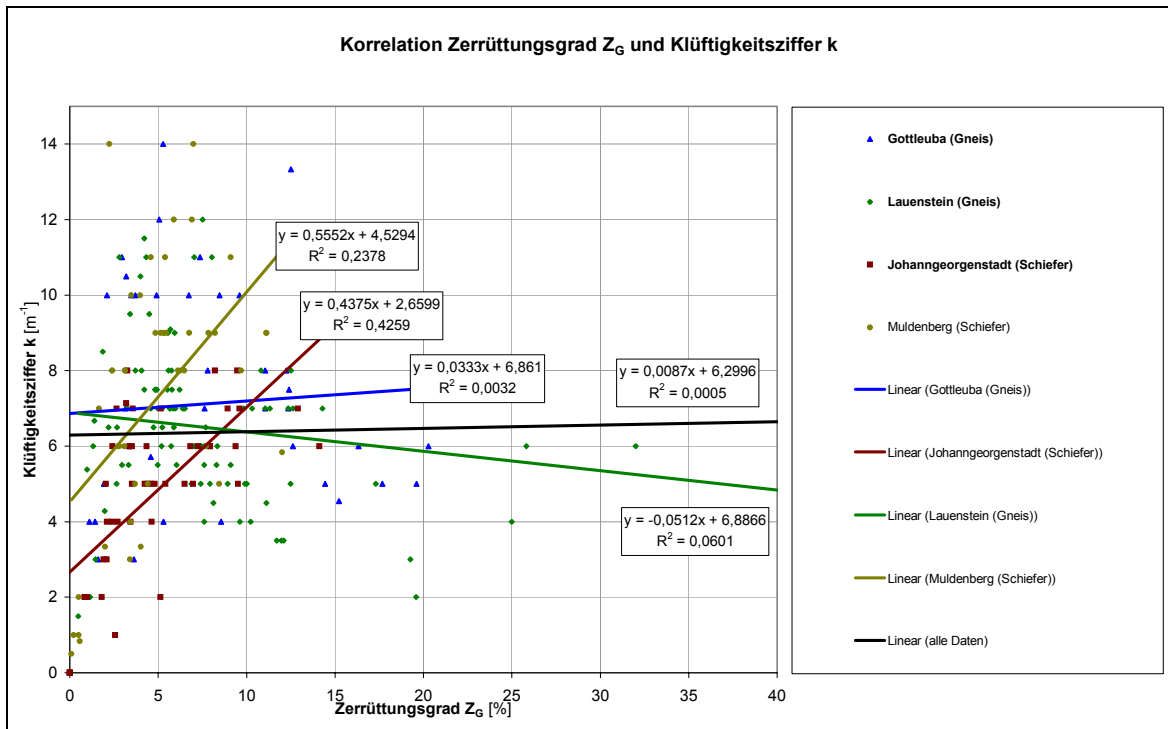


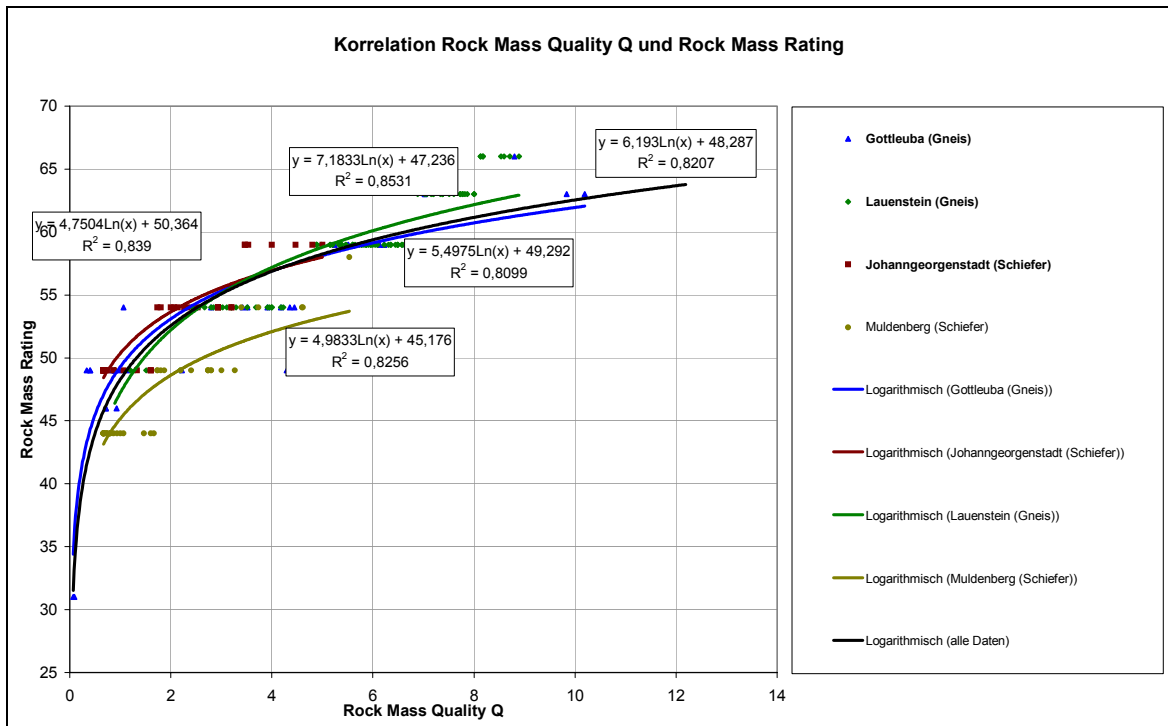
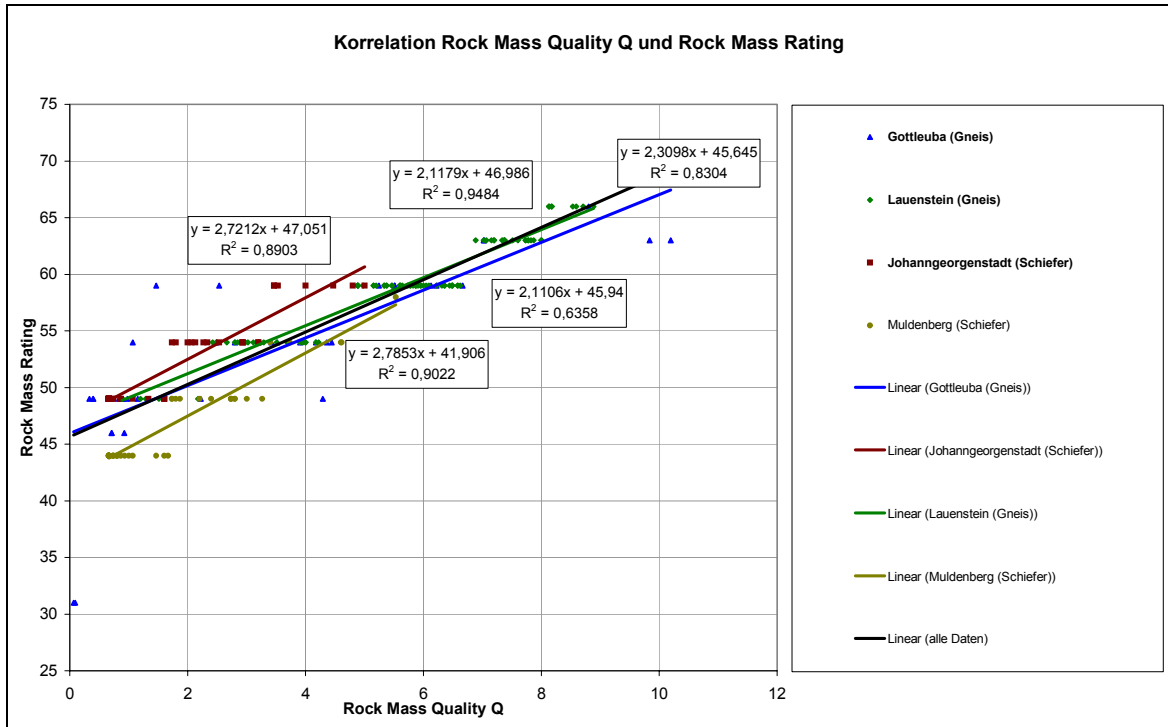










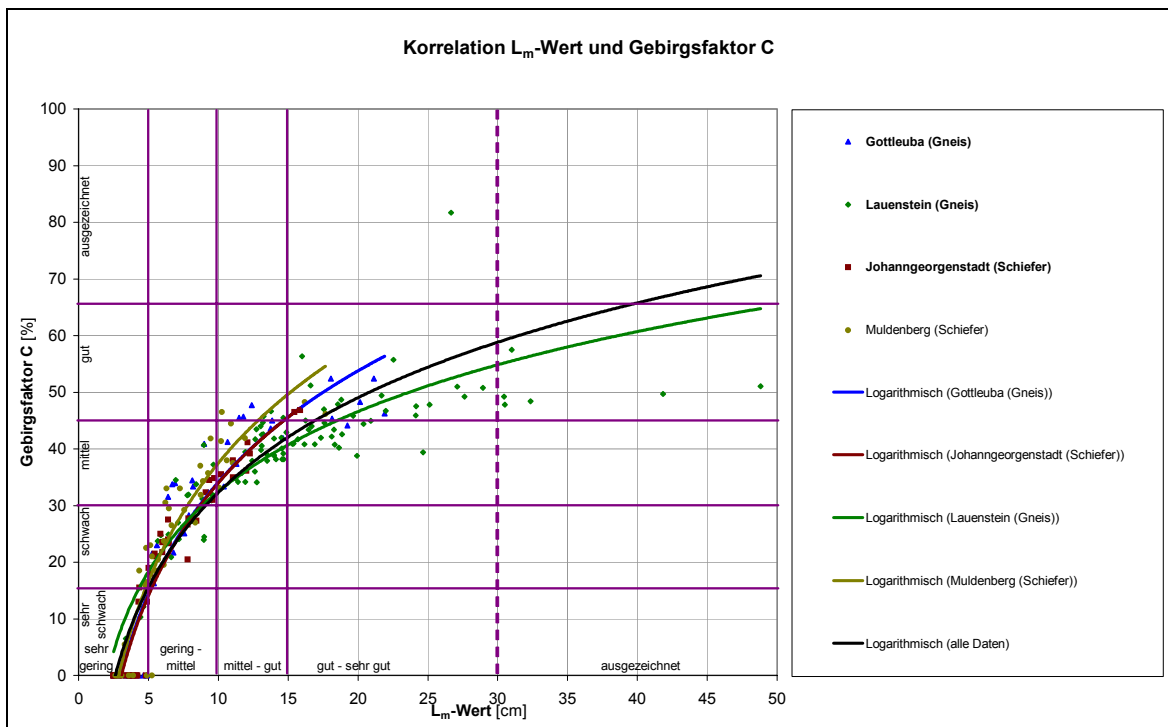
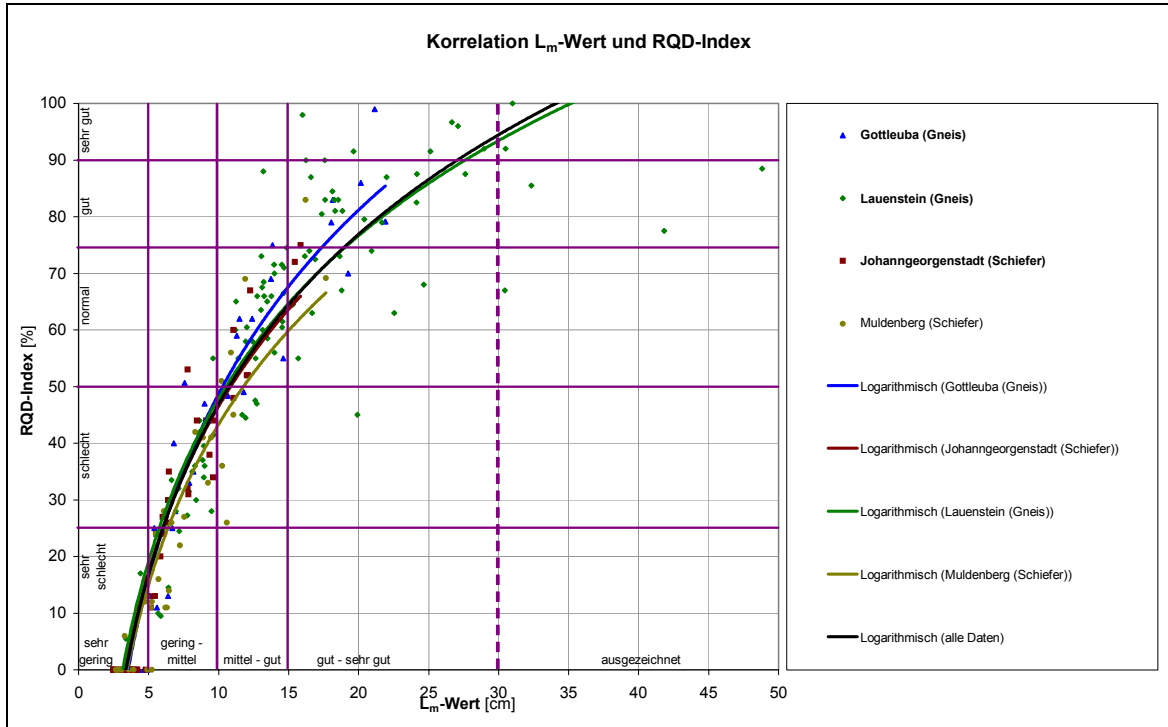


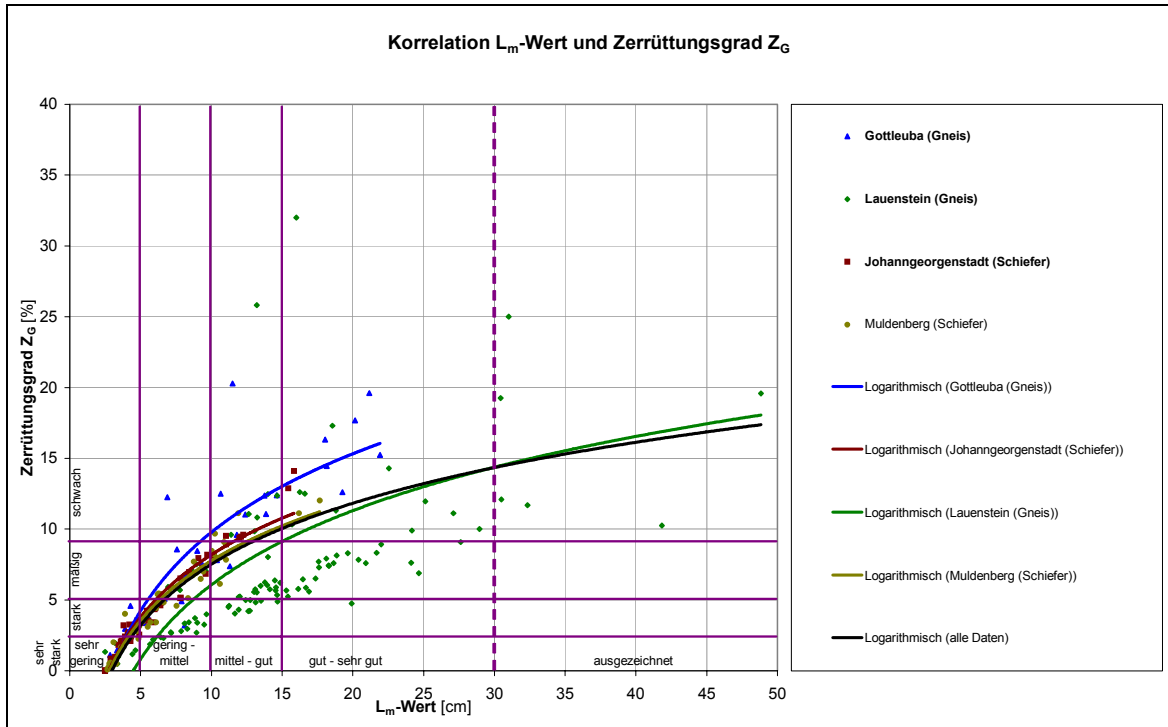
Zusammenstellung der Bestimmtheitsmaße einzelnen Verfahren (für lineare Korrelation)

Für verschiedene Verfahren - wie z. B. den L_m -Wert werden für nichtlineare Korrelationen (z.B. logarithmisch) bessere Korrelationen erzielt. Um die Vergleichbarkeit zu wahren wurden in der untenstehenden Tabelle ausschließlich die Bestimmtheitsmaße (R^2) für eine lineare Korrelation aufgeführt, die spezielleren (nichtlinearen) Korrelationen können den Diagrammen dieser Anlage entnommen werden.

R^2	L_m					RQD					C					Z_G					mk					k					
	Gottleuba	Lauenstein	Johanngeorgenstadt	Muldenberg	gesamt	Gottleuba	Lauenstein	Johanngeorgenstadt	Muldenberg	gesamt	Gottleuba	Lauenstein	Johanngeorgenstadt	Muldenberg	gesamt	Gottleuba	Lauenstein	Johanngeorgenstadt	Muldenberg	gesamt	Gottleuba	Lauenstein	Johanngeorgenstadt	Muldenberg	gesamt	Gottleuba	Lauenstein	Johanngeorgenstadt	Muldenberg	gesamt	
L _F	Gottleuba	1,000				0,891				0,729					0,794					0,681					0,000						
	Lauenstein		1,000			0,594				0,542					0,188					0,807					0,145						
	Johanngeorgenstadt			1,000			0,920			0,857				0,975					0,579					0,341							
	Muldenberg				1,000			0,876		0,755				0,899					0,471					0,180							
	gesamt					1,000				0,739					0,624					0,296				0,732							0,000
RQD	Gottleuba	0,891				1,000				0,846					0,807					0,724				0,001							
	Lauenstein		0,594				1,000			0,762					0,216				0,338				0,033								
	Johanngeorgenstadt			0,920				1,000		0,907				0,898					0,672				0,241								
	Muldenberg				0,876				1,000		0,738				0,816					0,447				0,149							
	gesamt					0,739				1,000					0,830					0,584				0,015							0,114
C	Gottleuba	0,729				0,846				1,000					0,731				0,849				0,079								
	Lauenstein		0,542				0,762				1,000				0,381				0,411				0,002								
	Johanngeorgenstadt			0,857				0,907				1,000			0,831				0,835				0,294								
	Muldenberg				0,755				0,738				1,000			0,790				0,714				0,346							
	gesamt					0,624				0,830				1,000					0,689				0,114								0,114
Z _G	Gottleuba	0,794				0,807				0,731					1,000				0,568				0,003								
	Lauenstein		0,188				0,216				0,381				1,000				0,080				0,060								
	Johanngeorgenstadt			0,975				0,898				0,831				1,000			0,546				0,426								
	Muldenberg				0,899				0,816				0,790				1,000			0,391				0,238							
	gesamt					0,296				0,323					0,383				1,000				0,001								0,001
mk	Gottleuba	0,681				0,724				0,849					0,568				1,000				0,038								
	Lauenstein		0,807				0,338				0,411				0,080				1,000				0,118								
	Johanngeorgenstadt			0,579				0,672				0,835			0,546				1,000				0,173								
	Muldenberg				0,471				0,447				0,714			0,391				1,000				0,209							
	gesamt					0,732				0,584					0,689				0,175				0,013								0,013
k	Gottleuba	0,000				0,001				0,079					0,003				0,038				1,000								
	Lauenstein		0,145				0,033				0,002				0,060				0,118				1,000								
	Johanngeorgenstadt			0,341				0,241				0,294			0,426				0,173				1,000								
	Muldenberg				0,180				0,149				0,346			0,238				0,209			1,000								
	gesamt					0,000				0,015				0,114					0,001				0,013								1,000
Mittelwert					0,478				0,498					0,528				0,236				0,439									0,029
Mittelwert (ohne k)					0,598				0,619					0,631				0,294				0,545									-
Platzierung					III				II				I		V				IV			VI									

Anlage XII: Vergleich der Klassen der Auswerteverfahren mit den Klassen des L_m -Wertes





Anlage XIII: Bibliographie

Diese ursprünglich von BIENIAWSKI [6] erstellte und erweiterte Bibliographie listet in chronologischer Reihenfolge Publikationen auf, die im Zusammenhang mit Gebirgsklassifikation stehen. Aufgrund des Umfangs der Liste und der Verfügbarkeit der Literatur konnten nicht alle Publikationen in die vorliegende Arbeit einbezogen werden.

TERZAGHI, K. (1946) "Rock Defects and Loads on Tunnel Support." Rock Tunneling with Steel Supports, ed. R. V. Proctor and T. White, Commercial Shearing Co., Youngstown, OH, S. 15-99.
STINI, I. (1950) "Tunnelbaugeologie" Springer-Verlag, Vienna, 336 S.
LAUFFER, H. (1958) "Gebirgsklassifizierung für den Stollenbau." Geol. Bauwesen 74, S. 46-51.
DEERE, D. U. (1963) "Technical Description of Rock Cores for Engineering Purposes" Rock Mech. Eng. Geol. 1, S. 16-22.
COATES, D. F. (1964) "Classification of Rock for Rock Mechanics." Int. J. Rock Mech. Min. Sei. 1, S. 421-429.
DEERE, D. U. & MILLER, R. P. (1966) "Engineering Classification and Index Properties of Intact Rock" Air Force Laboratory Technical Report no. AFNL-TR-65-116, Albuquerque, NM.
DEERE, D. U.; HENDRON A. J.; PATTON, F. D. & CORDING, E. J. (1967) "Design of Surface and Near Surface Constniction in Rock." Proc. 8th U.S. Symp. Rock Mech., AIME, New York, S. 237-302.
ROCHA, M. (1967) "A Method of Integral Sampling of Rock Masses." RockMech.3, S. 1-12.
BREKKE, T. L. (1968) "Blocky and Seamy Rock in Tunneling." Bull. Assoc. Eng. Geol. 5(1), S. 1-12.
DEERE, D. U. (1968) "Geological Considerations." Rock Mechanics in Engineering Practice, ed. R. G. Stagg and D. C. Zienkiewicz, Wiley, New York, S. 1-20.
CECIL, O. S. (1970) "Correlation of Rockbolts - Shotorete Support and Rock Quality Parameters in Scandinavian Tunnels," Ph.D. thesis, University of Illinois, Urbana, 414 S.
COON, R. F. & A. H. MERRITT (1970) "Predicting In Situ Modulus of Deformation Using Rock Quality Indexes," Determination of the In Situ Modulus of Deformation of Rock, ASTM Special Publication 477, Philadelphia, S. 154-173.
DEERE, D. U., R. B. PECK, H. PARKER, J. E. MONSEES & B. SCHMIDT (1970)

"Design of Tunnel Support Systems." High. Res. Rec., no. 339, S. 26-33,
FRANKLIN, F. A. (1970) "Observations and Tests for Engineering Description and Mapping of Rocks." Proc. 2nd Int. Cong. Rock Mech., ISRM, Belgrade, vol. 1, paper 1-3.
OBERT, L., C. RIEH. (1971) "Classification of Rock for Engineering Purposes." Proc. 1st Aust.-N.Z. Conf. Geomech., Australian Geomechanics Society, Melbourne, S. 435-441.
CORDING, E. J. & D. U. DEERE (1972) "Rock Tunnel Supports and Field Measurements." Proc. Rapid Excav. Tunneling Conf., AIME, New York, S. 601-622.
MERRITT, A. H. (1972) "Geologie Prediction for Underground Excavations." Proc. Rapid Excav. Tunneling Conf., AIME, New York, S. 115-132.
RABCEWICZ, L. & T. GOLSER. (1972) "Application of the NATM to the Underground Works at Tarbela." Water Power, S. 88-93.
SOKAL, R. R. (1972) "Classification: Purposes, Principles, Progress and Prospects." Science 185(4157), S. 1115-1123.
WICKHAM, G. E., H. R. TIEDEMANN & E. H. SKINNER (1972) "Support Determination Based on Geologie Predictions." Proc. Rapid Excav. Tunneling Conf., AIME, New York, S. 43-64.
BIENIAWSKI, Z. T. (1973) "Engineering Classification of Jointed Rock Masses" In Transaction of the South African Institution of Civil Engineers, 15, S. 335-344.
BARTON, N. (1974) "A review of the Shear Strength of Filled Discontinuities" Norwegian Geotechnical Institute, NR 105, S. 1-30
BARTON, N.; LIEN, R.; LUNDE, J.: Engineering Classification of Rock Masses for the Design of Tunnel Support - Rock Mechanics 6, Springer-Verlag, 1974
BIENIAWSKI, Z. T. (1974) "Estimating the Strength of Rock Materials." J. S. Afr. Inst. Min. Metall. 74(8), S. 312-320.
DEARMAN, W. R. & P. G. FOOKES (1974) "Engineering Geological Mapping for Civil Engineering Practice." Q. J. Eng. Geol. t, S. 223-256.
FÄCHER, F., L. RABCEWICZ & J. GOLSER (1974) "Zum derzeitigen Stand der Gebirgsklassifizierung in Stollen- und Tunnelbau." Proc. XXII Geomech. Colloq., Salzburg, S. 51-58.
FRANKLIN, J. A., C. LOUIS & P. MASURE (1974) "Rock Material Classification." Proc. 2nd int. Congr. Eng. Geol., IAEG, Sao Paulo, S. 325-341.
HANSÁGI, I.: A Method of Determining the Degree of Fissuration of Rock - Int. J. Rock Mech. Min. Sci. & Geomech. Abstr. Vol. 11, Pergamon Press 1974

LOUIS, C. (1974) "Reconnaissance des Massifs Rocheux par Sondages et Classifications Geotechniques des Roches." Ann. Inst. Tech. Paris, no. 108, S. 97-122.
MEIER, G.: Ingenieurgeol. Geländeuntersuchungen über repräsentative Ermittlungs- und Darstellungsmethoden strukturgeol. Parameter zur Bewertung der hydr. Gebirgsverhältnisse und der Injizierbarkeit von klüftigem Gebirge - BA Freiberg Diss. 1974
MÖBUS, G.: Zur Bearbeitung der Klüftigkeit des Kernmaterials von Bohrungen - Zeitschrift für angewandte Geologie, Bd. 20 (1974) Heft 8
PROTODYAKONOV, M. M. (1974) "Klassifikacija Gomych Porod." Tunnels Ouvrages Souterrains I, S. 31-34.
WICKHAM, G. E., H. R. TIEDEMANN & E. H. SKINNER (1974) "Ground Support Prediction Model, RSR Concept." Proc. Rapid Excav. Tunneling Conf., AIME, New York, S. 691-707.
BIENIAWSKI, Z. T. & R. K. MASCHEK (1975) "Monitoring the Behavior of Rock Tunnels during Construction." Civ. Eng. S. Afr. 17, S. 255-264.
CECIL, O. S. (1975) "Correlation of Rock Bolt-shotcrete Support and Rock Quality Parameters in Scandinavian Tunnels", Proceedings No 27, Swedish Geotechnical Institute, Stockholm 1-119.
FRANKLIN, J. A. (1975) "Safety and Economy in Tunneling" Proc. 10th Can. Rock Mech. Symp., Queens University, Kingston, S. 27-53.
HELFRICH, H. K.: Gebirgstechische Kennziffern mit petrographisch-strukturellem Begriffsinhalt - Rock Mechanics, Suppl. 4, 1975
KULHAWY, F. H. (1975) "Stress-Deformation Properties of Rock and Discontinuities." Eng. Geol. 9, S. 327-350.
WEAVER, J. M. (1975) "Geological Factors Significant in the Assessment of Rippability." Civ. Eng. S. Afr. 17, S. 313-316.
BARTON, N.: Recent experiences with the Q-system of tunnel support design - Proceedings of the Symposium on Exploration for Rock Engineering, Johannesburg, November 1976
BIENIAWSKI, Z. T & C. M. ORR (1976) "Rapid Site Appraisal for Dam Foundations by the Geomechanics Classification." Proc. 12th Cong. Large Dams, ICOLD, Mexico City, S. 483-501.
BIENIAWSKI, Z. T. (1976) "Elandsberg Pumped Storage Scheme - Rock Engineering

Investigations." Exploration for Rock Engineering, ed. Z. T. Bieniawski, A. A. Balkema, Johannesburg, S. 273-289.
BIENIAWSKI, Z. T.: Rock mass classification in rock engineering - Proceedings of the Symposium on Exploration for Rock Engineering, Johannesburg, November 1976
DAVIES, P. H. (1976) "Instrumentation in Tunnels to Assist in Economic Lining." Exploration for Rock Engineering, ed. Z. T. Bieniawski, A. A. Balkema, Johannesburg, S. 243-252.
FRANKLIN, J. A. (1976) "An Observational Approach to the Selection and Control of Rock Tunnel Linings." Proc. Conf. Shotcrete Ground Control, ASCE, Easton, MA, S. 556-596.
KENDORSKI, F. S. & J. A. BISCHOFF (1976) "Engineering Inspection and Appraisal of Rock Tunnels." Proc. Rapid Excav. Tunneling Conf., AIME, New York, S. 81-99.
LAUBSCHER, D. H.; TAYLOR, H. W.: The importance of geomechanics classification of jointed rock masses in mining operations - Proceedings of the Symposium on Exploration for Rock Engineering, Johannesburg, November 1976
MCDONOUGH, J. T. (1976) "Site Evaluation for Cavability and Underground Support Design at the Climax Mine." Proc. 17th U.S. Symp. Rock Mech., University of Utah, Snowbird, S. 3A2-3A15.
OLIVER, H. J.: Determination of RQD from petroscope observations - Proceedings of the Symposium on Exploration for Rock Engineering, Johannesburg, November 1976
PRIEST, S.D. & HUDSON, J.A. (1976) "Estimation of discontinuity spacing and trace length using scan line surveys" Int. J. Rock Mech. Min. Sci and Geomech., Vol. 18, S. 183-197.
FERGUSON, G. A. (1977) "The Design of Support Systems for Excavations in Chrysotile Asbestos Mines" M. Phil. thesis, University of Rhodesia, Salisbury, 261 S.
LAUBSCHER, D. H. (1977) "Geomechanics Classification of Jointed Rock Masses - Mining Applications." Trans. Instn. Min. Metall. 86, S. A-1-A-7.
SPAUN, G. (1977) "Contractual Evaluation of Rock Exploration in Tunnelling." Exploration for Rock Engineering, ed. Z. T. Bieniawski, A. A. Balkema, Johannesburg, vol. 2, S. 49-52.
BIENIAWSKI, Z. T. (1978) "Determining Rock Mass Deformability - Experience from Case Histories." Int. J. Rock Mech. Min. Sei. 15, S. 237-247.
DITTRICH, E.: Die Ermittlung ingenieurgeologisch bedeutsamer Gefügeparameter an

Hand von Bohrkernen - Z. geol. Wiss. Berlin 1978 Band 6
DOWDING, C. D., ED. (1978) "Site Characterization and Exploration" ASCE, New York, 321 S.
FISHER, P. & D. C. BANKS (1978) "Influence of the Regional Geologic Setting on Site Geological Features." Site Characterization and Exploration, ed. C. E. Dowding, ASCE, New York, S. 302-321.
HAIMSON, B. C. (1978) "The Hydrofracturing Stress Measuring Method and Field Results" Int. J. Rock Mech. Min. S. 15, S. 167-178.
HWONG, T. (1978) "Classification of the Rock Mass Structures and Determination of Rock Mass Quality." Bull. Int. Assoc. Eng. Geol., no. 18, S. 139-142.
MEIER, G.: Zum Problem der hydrodynamisch wirksamen Trennflächen - Freiburger Forschungshefte., R A597, Leipzig 1978
MÜLLER, L. (1978) "Removing Misconceptions on the New Austrian Tunnelling Method." Tunnels Tunnelling 10, S. 667-671.
RUTLEDGE, J. C. & R. L. PRESTON (1978) "Experience with Engineering Classifications of Rock." Proc. Int. Tunneling Symp., Tokyo, S. A3.1-A3.7.
RUTLEDGE, J. C. (1978) "Engineering classification of rock for the determination of tunnel support" in Proceedings of the International Tunneling Symposium Tokyo.
BIENIAWSKI, Z. T. (1979) "The Geomechanics classification in rock engineering applications" Proc. In Proc. 4th Int. Cong. Rock Mech. ISRM, Montreal, 2, pp. 41-48.
BIENIAWSKI, Z. T. (1979) "Tunnel Design by Rock Mass Classifications" U.S. Army Corps of Engineers Technical Report GL-799-19, Waterways Experiment Station, Vicksburg, MS, S. 50-62.
BLACKKEY, E. A. (1979) "Park River Auxiliary Tunnel." J. Constr. Div. ASCE 105 (C04), S. 341-349.
BRADLEY, W.B. (1979) "Failure of Inclined Boreholes" Trans. of ASME, Vol. 101, Dec. 1979, S. 232-239
EINSTEIN, H. H., W. STEINER & G. B. BAECHEER (1979) "Assessment of Empirical Design Methods for Tunnels in Rock." Proc. Rapid Excav. Tunneling Conf., AIME, New York, S. 683-706.
GOLSER, J. (1979) "Another View of the NATM." Tunnels Tunnelling II, S. 41-42.
JAEGER, J. C. & N. G. W. COOK (1979) "Fundamentals of Rock Mechanics" Chapman & Hall, London, 3rd ed., 593 S.

KIDYBINSKI, A. (1979) "Experience with Rock Penetrometers for Mine Rock Stability Predictions." Proc. 4th Int. Congr. Rock Mech., ISRM, Montreux, S. 293-301.
OLIVIER, H. J. (1979) "A New Engineering - Geological Rock Durability Classification." Eng. Geol. 14, S. 255-279.
OLIVIER, H. J. (1979) "Applicability of the Geomechanics Classification to the Oranee-Fish Tunnel Rock Masses." Civ. Eng. S. Afr. 21, S. 179-185.
BACZYNSKI, N. (1980) "Rock Mass Characterization and its Application to Assessment of Unsupported Underground Openings" Ph.D. thesis, University of Melbourne, 233 S.
BARTON, N., F. LOSET, R. LIEN & J. LUNDE (1980) "Application of Q-System in Design Decisions" Subsurface Space, ed. M. Bergman, Pergamon, New York, S. 553-561.
GOODMAN, R. E. (1980) "Introduction to Rock Mechanics" Wiley, New York, 478 S.
HOEK, E. & BROWN, E. T. (1980) "Empirical Strength Criterion for Rock Masses." J. Geotech. Eng. ASCE 106(GT9), S. 1013-1035.
HOEK, E. & BROWN, E. T. (1980) "Underground Excavations in Rock" S. 527. Institution of Mining and Metallurgy, London.
JOHN, M. (1980) "Investigation and Design for the Arlberg Expressway Tunnel." Tunnels Tunnelling 12, S. 46-51.
WILLIAMSON, D. A. (1980) "Uniform Rock Classification for Geotechnical Engineering Purposes." Trans. Res. Rec., no. 783, S. 9-14.
BIENIAWSKI, Z. T. (1981) "Rock Classifications: State of the Art and Prospects for Standardization." Transp. Res. Rec., no. 783, S. 2-8.
BROWN, E. T. (1981) "Putting the NATM in Perspective." Tunnels Tunnelling 13, S. 13-17.
CAMERON-CLARK, L. S. & S. BUDAVARI (1981) "Correlation of Rock Mass Classification Parameters obtained from Borehole and In Situ Observations." Eng. Geol. 17, S. 19-53.
DAUGHERTY, C. W. (1981) "Logging of Geologic Discontinuities in Boreholes and Rock Cores." Proc. Short Course Subsurf. Explor., George Washington University, Washington, DC.
ENGELS, J. G., J. T. CAHILL & E. A. BLACKKEY (1981) "Geotechnical Performance of a Large Machined-Bored Precast Concrete Lined Tunnel." Proc. RapidExcav. Tunneling Conf., AIME, New York, S. 1510-1533.
GHOSE, A. K. & N. M. RAJU (1981) "Characterization of Rock Mass vis-à-vis

Application of Rock Bolting in Indian Coal Measures." Proc. 22nd U.S. Symp. Rock Mech., MIT, Cambridge, MA, S. 422-427.
International Association of Engineering Geology. (1981). "Guidelines for Site Investigations." Bull. Int. Assoc. Eng. Geol., no. 24, S. 185-226.
International Association of Engineering Geology. (1981). "Rock and Soil Description for Engineering Geological Mapping," Bull. Int. Assoc. Eng. Geol., no. 24, S. 235-274.
International Society for Rock Mechanics. (1981). "Basic Geotechnical Description of Rock Masses." Int. J. Rock Mech. Min. Sei. 18, S. 85-110.
International Society for Rock Mechanics. (1981). Rock Characterization, Testing and Monitoring - ISRM Suggested Methods, Pergamon, London, 211 S.
LAUBSCHER, D. H. (1981) "Selection of Mass Underground Mining Methods." Design and Operation of Caving and Sub-Level Storing Mines, ed. D. R. Stewart, AIME, New York, S. 23-38.
CUMMINGS, R. A., F. S. KENDORSKI & Z. T. BIENIAWSKI (1982) "Caving Rock Mass Classification and Support Estimation" U. S. Bureau of Mines Contract Report #J0100103, Engineers International, Inc., Chicago, 195 S.
FOWELL, R. J. & S. T. JOHNSON (1982) "Rock Classifications for Rapid Excavation Systems." Proc. Symp. Strata Mech., Eisevier, Amsterdam, S. 241-244.
HOCK, E. (1982) "Geotechnical Considerations in Tunnel Design and Contract Preparation." Trans. Inst. Min. Metall. 91, S. AIOI-A109.
JETHWA, J. L., A. K. DUBE.B. SINGH & R. S. MITHAI (1982) "Evaluation of Methods for Tunnel Support Design in Squeezing Rock Conditions." Proc. 4th Int. Congr. Int. Assoc. Eng. Geol., Delhi, vol. 5, S. 125-134.
KIDYBINSKI, A. (1982) "Classification of Rock for Longwall Cavability." State-of-the-Art of Ground Control in Longwall Mining, AIME, New York, S. 31-38.
KIRSTEN, H. A. D. (1982) "A Classification System for Excavation in Natural Materials." Civ. Eng. S. Afr. 24, S. 293-308.
MÖBUS, G.: Methodik der tektonischen Bearbeitung des Kernmaterials von Bohrungen - Freiburger Forschungshefte C 372 Leipzig 1982
MORENO TALLON, E. (1982) "Comparison and Application of the Geomechanics Classification Schemes in Tunnel Construction." Proc. Tunneling '82, Institute of Mining and Metallurgy, London, S. 241-246.
PALMSTRÖM, A. (1982) "The Volumetrie Joint Count - a Useful and Simple Measure of

the Degree of Rock Jointing." Proc. 4th int. Congr. int. Assoc. Eng. Geol., Delhi, vol. 5, S. 221-228.
ABAD, J., B. CELADA.E. CHACON, V. GUTIERREZ & E. HIDALGO (1983) "Application of Geomechanical Classification to Predict the Convergence of Coal Mine Galleries and to Design Their Supports." Proc. 5th Int. Congr. Rock Mech., ISRM, Melbourne, vol. 2, S. E15
ABDULLATIF, O. M. & D. M. GRUDEN (1983) "The Relationship between Rock Mass Quality and Ease of Excavation." Bull. Int. Assoc. Eng. Geol., no. 28, S. 184-87.
BARTON, N. (1983) "Application of Q-System and Index Tests to Estimate Shear Strength and Deformability of Rock Masses." Proc. Int. Symp. Eng. Geol. Underground Constr., A. A. Balkema, Rotterdam, S. 51-70.
BIENIAWSKI, Z. T. (1983) "The Geomechanics Classification (RMR System) in Design Applications to Underground Excavations." Proc. Int. Symp. Eng. Geol. Underground Constr., A. A. Balkema, Rotterdam, S. 1.33-1.47.
EINSTEIN, H. H., A. S. AZZOUZ, A. F. MCKNOWN & D.E. THOMPSON (1983) "Evaluation of Design and Performance - Porter Square Transit Station Chamber Lining." Proc. Rapid Excav. Tunneling Conf., AIME, New York, S. 597-620.
GONZALEZ DE VALIEJO, L. L. (1983) "A New Rock Classification System for Underground Assessment Using Surface Data." Proc. Int. Symp. Eng. Geol. Underground Constr., A. A. Balkema, Rotterdam, S. 85-94.
KENDORSKI, F., R. CUMMINGS, Z. T. BIENIAWSKI & E. SKINNER (1983) "Rock Mass Classification for Block Caving Mine Drift Support." Proc. 5th int. Congr. Rock Mech., ISRM, Melbourne, S. B51-B63.
KIRSTEN, H. A. D. (1983) "The Combined Q/NATM System - The Design and Specification of Primary Tunnel Support." S. Afr. Tunnelling 6, S. 18-23.
LOKIN, P., R. NIJAJILOVIC & M. VASIC (1983) "An Approach to Rock Mass Classification for Underground Works." Proc. 5th Int. Congr. Rock Mech., ISRM, Melbourne, vol. 1, S. B87-B92.
MELLIS, L. M. J. & A. G. DELL (1983) "Primary Support Assessment with the Q/NATM System and Rock-Lining Interaction Considerations for Permanent Support Design." Proc. Symp. Rock Mech. Design Tunnels, SANGORM, Johannesburg, S. 15-32.
MOLEK, H.: Aussagemöglichkeiten strukturgeologischer Bohrkernuntersuchungsmethoden in Subrosionsgebieten - Zeitschrift für angewandte Geologie, Bd. 29 (1983) Heft 10

NAKAO, K., S. LIHOSHI & S. KOYAMA (1983) "Statistical Reconsiderations on the Parameters for Geomechanics Classification." Proc. 5th Int. Congr. Rock Mech., ISRM, Melbourne, vol. 1, S. B13-B16.
OLIVEIRA, R., C. COSTA & J. DAVIS (1983) "Engineering Geological Studies and Design of Castelo Do Bode Tunnel." Proc. Int. Symp. Eng. Geol. Underground Constr., A. A. Balkema, Rotterdam, vol. 1, S. 11.69-11.84.
PRIEST, S. D. & E. T. BROWN (1983) "Probabilistic Stability Analysis of Variable Rock Slopes." Trans. Inst. Min. Metall. London 92, S. AI-A12.
SERAFIRN, J. L. & J. P. PEREIRA (1983) "Considerations of the Geomechanics Classification of Bieniawski." Proc. Int. Symp. Eng. Geol. Underground Constr., A. A. Balkema, Rotterdam, vol. 1, S. 11.33-11.42.
UNAL, E. (1983) "Design Guidelines and Roof Control Standards for Coal Mine Roofs," Ph.D. thesis, Pennsylvania State University, University Park, 355 S.
UNRUG, K. & T. B. SZWILSKI (1983) "Strata Cavability in Longwall Mining." Proc. 2nd Int. Conf. Stability Underground Mine, AIME, New York, S. 131-147.
WELTMAN, A. J. & J. M. HEAD (1983) "Site Investigation Manual" Construction Industry Research and Information Association, London, Special Publication no. 25, 144 S.
WHITNEY, H. T. & G. L. BUTLER (1983) "The New Austrian Tunneling Method - a Rock Mechanics Philosophy." Proc. 24th U.S. Symp. Rock Mech., Texas A&M University, College Station, TX, S. 219-226.
YUDHBIR (1983) "An Empirical Failure Criterion for Rock Masses." Proc. 5th Int. Congr. Rock Mechanics, ISRM, Melbourne, S. BI-B8.
BIENIAWSKI, Z. T. (1984) "Rock Mechanics Design in Mining and Tunneling" A. A. Balkema, Rotterdam, S. 97-133.
BONIFACE, A. A. (1984) "Commentary on Three Methods of Estimating Support Requirements for Underground Excavations." Design and Construction of Large Underground Openings, ed. E. L. Giles and N. Gay, SANCOT, Johannesburg, S. 33-39.
LAUBSCHER, D. H. (1984) "Design Aspects and Effectiveness of Support Systems in Different Mining Conditions." Trans. Inst. Min. Metall. 93, S. A70-A81.
MOLEK, H.: Ingenieurgeol. Untersuchungen in Subrosionsgebieten unter besonderer Berücksichtigung der Ermittlung des Zerrüttungsgrades im postsalinaren Deckgebirge - BA Freiberg Diss. 1984
PECK, R. B. (1984) "Judgment in Geotechnical Engineering" Wiley, New York, 332 S.

U.S. National Committee on Tunneling Technology. (1984). Geotechnical Site Investigations for Underground Projects. National Academy Press, Washington, DC, 182 S.
WILLIAMSON, D. A. (1984) "Unified Rock Classification System." Bull. Assoc. Eng. Geol. 21(3), S. 345-354.
BONIFACE, A. (1985) "Support Requirements for Machine Driven Tunnels." S. Afr. Tunnelling 8, p. 7.
BROOK, N. & P. G. R. DHARMARATNE (1985) "Simplified Rock Mass Rating System for Mine Tunnel Support." Trans. Inst. Min. Metall. 94, S. A148-A154.
FAIRHURST, C. & D. LIN (1985) "Fuzzy Methodology in Tunnel Support Design." Proc. 26th U.S. Symp. Rock Mech., A. A. Balkema, Rotterdam, vol. 1, S. 269-278.
NEWMAN, D. A. (1985) "The Design of Coal Mine Roof Support for Longwall Mines in the Appalachian Coalfield," Ph.D. thesis, Pennsylvania State University, University Park, 401 S.
NGUYEN, V. U. & E. ASHWORTH (1985) "Rock Mass Classification by Fuzzy Sets." Proc. 26th U.S. Symp. Rock Mech., A. A. Balkema, Rotterdam, vol. 2, S. 937-946.
ROMANA, M. (1985) "New Adjustment Ratings for Application of Bieniawski Classification to Slopes." Proc. Int. Symp. Rock Mech. Excav. Min. Civ. Works, ISRM, Mexico City, S. 59-68.
SANDBAK, L. A. (1985) "Roadheader Drift Excavation and Geomechanics Rock Classification." Proc. Rapid Excav. Tunneling Conf., AIME, New York, vol. 2, S. 902-916.
SHEOREY, P. R. (1985) "Support Pressure Estimation in Failed Rock Conditions." Eng. Geol. 22, S. 127-140.
BROSCH, F. J. (1986) "Geology and Classification of Rock Masses - Examples from Austrian Tunnels." Bull. Int. Assoc. Eng. Geol., no. 33, S. 31-37.
FARMER, L. W. (1986) "Energy Based Rock Characterization." Application of Rock Characterization Techniques in Mine Design, ed. M. Karmis, AIME, New York, S. 17-23.
FRANKLIN, J. A. (1986) "Size-Strength System for Rock Characterization." Application of Rock Characterization Techniques in Mine Design, ed. M. Karmis, AIME, New York, S. 11-16.
GRAINGER, G. S. (1986) "Rock Mass Characteristics of the Rocky Mountain Pumped Storage Project Hydroelectric Tunnel and Shaft." Proc. 27th U.S. Symp. Rock

KAISER, P. K.; MACKAY, C.; GALE, A. D.: Evaluation of Rock Classification at B. C. Rail Tumbler Ridge Tunnels - Rock Mechanics and Rock Engineering, Springer-Verlag, 1986
NEWMAN, D. A. & Z. T. BIENIAWSKI. (1986) "Modified Version of the Geomechanics Classification for Entry Design in Underground Coal Mines." Trans. Soc. Min. Eng. AIME 280, S. 2134-2138.
NICHOLSON, G. A. & Z. T. BIENIAWSKI (1986) "An Empirical Constitutive Relationship for Rock Mass." Proc. 27th U.S. Symp. Rock Mech., AIME, New York, S. 760-766.
SINGH, R. N., A. M. ELMHERIG & M. Z. SUNU (1986) "Application of Rock Mass Characterization to the Stability Assessment and Blast Design in Hard Rock Surface Mining Excavations." Proc. 27th U.S. Symp. Rock Mech., AIME, New York, S. 471-478.
SINGH, R. N., B. DENBY, L. EGRETLI & A. G. PATHON (1986) "Assessment of Ground Rippability in Opencast Mining Operations." Min. Dept. Mag. Univ. Nottingham 38, S. 21-34.
SMITH, H. J. (1986) "Estimating Rippability by Rock Mass Classification." Proc. 27th U.S. Symp. Rock Mech., AIME, New York, S. 443-448.
STACEY, T. R.; PAGE, C. H.: Practical Handbook for Underground Rock Mechanics - Trans Tech Publications P.O. Box 266 D-3392 Clausthal-Zellerfeld Federal Republic of Germany
UNAL, E. (1986) "Empirical Approach to Calculate Rock Loads in Coal Mine Roadways." Proc. 5th Conf. Ground Control Coal Mines, West Virginia University, Morgantown, S. 234-241.
VENKATESWARLU, V. (1986) "Geomechanics Classification of Coal Measure Rocks vis-à-vis Roof Supports," Ph.D. thesis, Indian School of Mines, Dhanbad, 251 S.
BIENIAWSKI, Z. T. (1987) "Strata Control in Mineral Engineering" Wiley, New York, 212 S.
KLAASSEN, M. J., C. H. MACKAY, T. J. MOMS & D. G. WASYLUK (1987) "Engineering Geological Mapping and Computer Assisted Data Processing for Tunnels at the Rogers Pass Project, B.C." Proc. Rapid Excav. Tunneling Conf., AIME, New York, S. 1309-1322.
LEBEL, G. & C. O. BRAWNER (1987) "An Investigation on Rock Quality Index." Min. Sei. Tech. 5, S. 71-82.
SCHMIDT, B (1987) "Learning from Nuclear Repository Design: The Ground Control

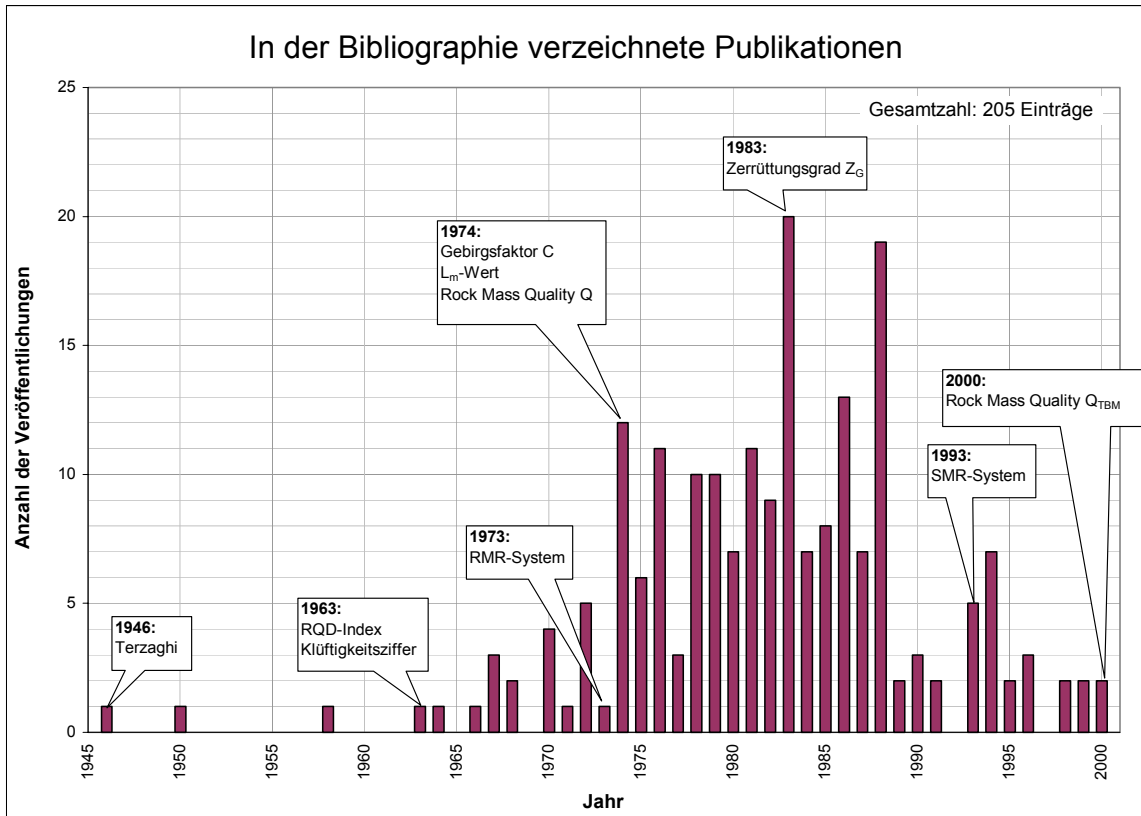
Plan." Proc. 6th Aust. Tunneling Conf., Australian Geomechanics Society, Melbourne, S. 11-19.
SINGH, R. N., B. DENBY & L. EGRETLI (1987) "Development of a New Rippability Index for Coal Measures." Proc. 28th U.S. Symp. Rock Mech., A. A. Balkema, Boston, S. 935-943.
SMITH, H. J. (1987) "Estimating the Mechanical Dredgeability of Rock." Proc. 28th U.S. Symp. Rock Mech., A. A. Balkema, Boston, S. 935-943.
STEVENS, C. R., L. A. SANDBAK & J. J. HUNTER (1987) "LHD Production and Design Modifications at the San Manuel Mine." Proc. 28th U.S. Symp. Rock Mech., A. A. Balkema, Boston, S. 1175-1185.
BARTON, N. (1988) "Rock Mass Classification and Tunnel Reinforcement Selection using the Q-System." Proc. Symp. Rock Class. Eng. Purp., ASTM Special Technical Publication 984, Philadelphia, S. 59-84.
BIENIAWSKI, Z. T. (1988) "Rock Mass Classification as a Design Aid in Tunnelling." Tunnels Tunneling 20, S. 19-22.
BROWN, E. T. & E. HOCK (1988) "Discussion on Shear Failure Envelope in Rock Masses." J. Geotech. Eng. ASCE 114, S. 371-373.
DEERE, D. U. & DEERE, D. W. (1988) "The RQD Index in Practice." Proc. Symp. Rock Class. Eng. Purp., ASTM Special Technical Publication 984, Philadelphia, S. 91-101.
FARIA SANTOS, C. (1988) "Analysis of Coal Mine Floor Stability," Ph.D. thesis, Pennsylvania State University, University Park, 189 S.
GHOSE, A. H. & D. GUPTA (1988) "A Rock Mass Classification Model for Caving Roofs." Int. J. Min. Geol. Eng. 5, S. 257-271.
HANNA, K. & D. P. CONOVER (1988) "Design of Coal Mine Entry Intersection." AIME-SME Ann. Meet., Phoenix, AZ, preprint #88-39.
KIRKALDIE, L., D. A. WILLIAMSON & P. V. PATTERSON (1988) "Rock Material Field Classification Procedure." Rock Classification Systems for Rock Engineering, ASTM STP 984, ASTM, Philadelphia, S. 133-167.
KIRSTEN, H. A. D. (1988) "Case Histories of Groundmass Characterization for Excavatability." Proc. Symp. Rock Class. Eng. Purp., ASTM Special Technical Publication 984, Philadelphia, S. 102-120.
MEIER, G.: Geotechnische Aufgaben bei Bergsicherungsarbeiten zur Erkundung und Sanierung von Altbergbau im Fels an typischen Beispielen des Thüringer Raumes und des

Erzgebirges - BA Freiberg Diss. 1988
MILNE, D. (1988) "Suggestions for standardization of rock mass classification" MSc Dissertation, Imperial College, University of London, S. 123.
NICHOLSON, G. A. (1988) "A Case History Review from a Perspective of Design by Rock Mass Classification Systems." Proc. Symp. Rock Class. Eng. Purp., ASTM Special Technical Publication 984, Philadelphia, S. 121-129.
POTVIN, Y. (1988) "Empirical open stope design in Canada" Ph.D. thesis. The University of British Columbia S. 350.
ROBERTSON, A. M. (1988) "Estimating Weak Rock Strength." AIME-SME Ann. Meet., Phoenix, AZ, preprint #88-145.
RODRIGUES, J. D. (1988) "Proposed Geotechnical Classification of Carbonate Rocks Based on Portuguese and Algerian Examples." Eng. Geol. 25, S. 33-43.
ROMANA, M.: Practice of SMR classification for slope appraisal - Proceedings 5th International Symposium on Landsides, Lausanne, Balkema, Rotterdam (1988)
SANDBAK, L. A. (1988) "Rock Mass Classification in LHD Mining at San Manuel." AIME-SME Ann. Meet., Phoenix, AZ, preprint Nr.88-26.
SKINNER, E. H. (1988) "A Ground Support Prediction Concept—the RSR Model." Proc. Symp. Rock Class. Eng. Purp., ASTM Special Technical Publication 984, Philadelphia, S. 35-51.
ZHOU, Y., C. HAYCOCKS & W. WU. (1988) "Geomechanics Classification for Multiple Seam Mining." AIME-SME Ann. Meet., Phoenix, AZ, preprint #88-II.
BIENIAWSKI, Z. T.: Engineering Rock Mass Classifications - John Wiley & Sons, Inc. 1989
DEERE, D. U. (1989) "Rock Quality Designation (RQD) after Twenty Years" U.S. Army Corps of Engineers Contract Report GL-89-1, Waterways Experiment Station, Vicksburg, MS, 67 S.
BIENIAWSKI, Z.T. (1990) "Tunnel Design by Rock Mass Classification" US Army Corps of Engineers, Washington, DC.
LAUBSCHER, D. H. (1990) "A Geomechanics Classification System for the Rating of Rock Mass in Mine design". Journal of the South African Institute of Mining & Metallurgy, vol. 90, no 10, S. 257-273.
LANG, B.; PAKALNIS, R. & VONGPAISAL, S. (1991) "Span Design in wide cut and fill Stopes at Detour Lake Mine" 93rd Annual General Meeting: Canadian Institute of Mining,

Vancouver, paper # 142.
MEIER, G.: Grundsätze von Bergsicherungsarbeiten im Gangbau - WTI Jahrgang 32/1991 Reihe A Heft 1
CHOQUET, P. & HADJIGEORGIOU, J. (1993) "Design of support for Underground Excavations" Chapter 12, Vol. 4, J. Hudson Ed. Comprehensive Rock Engineering, S.313-348.
GRIMSTAD, E. & BARTON, N. (1993) "Updating of the Q-system for Norwegian Method of Tunnelling" in Proceedings of the International Symposium on Sprayed Concrete, S. 46-66. Norwegian Concrete Association, Oslo.
HACK, H. R. G. K.; PRICE D. G.: A rock mass classification system for the design and safety analysis of slopes - Eurock '93, Ribeiro e Sousa & Grossmann (eds) 1993 Balkema, Rotterdam 90 5410 339 6
HUDSON, J. A.: Comprehensive rock engineering: principles, practice and projects (Volume 3, Rock testing and site characterization) - Pergamon Press Ltd 1993
HUDSON, J. A.: Comprehensive rock engineering: principles, practice and projects (Volume 5, Surface and underground case histories) - Pergamon Press Ltd 1993
ANDRASKAY, E.: Überlegungen zur Definition der Ausbruchklassen in der SIA-Norm 198 Untertagebau, Ausgabe 1993 - Felsbau 12 (1994) Nr. 6, Verlag Glückauf GmbH
AYAYDIN, N.: Entwicklung und neuester Stand der Gebirgsklassifizierung in Österreich - Felsbau 12 (1994) Nr. 6, Verlag Glückauf GmbH
BARTON, N.; GRIMSTAD, E.: The Q-System following Twenty Years of Application in NMT Support Selection - Felsbau 12 (1994) Nr. 6, Verlag Glückauf GmbH
BAUDENDISTEL, M.: Zur Vortriebsklassifizierung in Deutschland - Felsbau 12 (1994) Nr. 6, Verlag Glückauf GmbH
BIENIAWSKI, Z. T.; ALBER, M.: Effektive Gebirgsklassifizierung durch systematisches Entwurfsverfahren - Felsbau 12 (1994) Nr. 6, Verlag Glückauf GmbH
DALLER, J.; RIEDMÜLLER, G.; SCHUBERT, W.: Zur Problematik der Gebirgsklassifizierung im Tunnelbau - Felsbau 12 (1994) Nr. 6, Verlag Glückauf GmbH
JOHN, M.: Zielsetzung der Gebirgsklassifizierung - Felsbau 12 (1994) Nr. 6, Verlag Glückauf GmbH
HOEK, E.; KAISER, P.K. & BAWDEN, W. (1995) "Support of Underground Excavations in Hard Rock" Balkema. S. 215.
MAIDL: Handbuch des Tunnel- und Stollenbaus - Verlag Glückauf GmbH Essen 1995,

Band II, 2. Auflage
BARTON, N. (1996) "Investigation, design and support of major road tunnels in jointed rock using NMT principles" in Proceedings IX Australian Tunnelling Conference, S. 145-159. Australian Underground Construction and Tunnelling Association, Sydney.
FECKER, E.; REIK, G.: Baugeologie - Enke Verlag Stuttgart 1996, 2. Auflage
WARREN, C.D. P.M. VARLEY R. PARKIN (1996) "UK Tunnels: Geotechnical Monitoring and Encountered Conditions" Ch. 14 of Engineering Geology of the Channel Tunnel, Eds. Harris et al., S. 219 243 Thomas Telford.
HOEK, E.: Practical Rock Engineering - University of Toronto
KOYMBAS, D.: Geotechnik - Tunnelbau und Tunnelmechanik - Springer-Verlag Berlin Heidelberg New York 1998
DEPARTMENT OF MINERALS AND ENERGY WESTERN AUSTRALIA: Technical Update No.1: The "Q-System" geotechnical design method was updated in 1994 - Mining Operation Division Mineral House, Perth
MEIER, G.: Ein repräsentatives Verfahren zur ingenieurgeologischen Bohrkernauswertung im Fels in Altbergbaugebieten - Berichte 12. Nat. Tagung f. Ing.-Geol. Halle 1999
BARTON, N; JHUN: TBM Performance estimation in rock using QTBM - SR005.pdf
EICHLER, K.: Fels- und Tunnelbau - expert-Verlag 2000

An dieser Stelle sei noch auf die angekündigte Ausgabe 4/01 der Zeitschrift FELSBAU (Erscheinungstermin 26.07.2001) mit dem Schwerpunktthema Gebirgsklassifizierung verwiesen (Quelle: FELSBAU 19 (2001) Nr. 1, Seite 6), die aufgrund des Erscheinungstermins nicht in die Auswertung für diese Studienarbeit einfließen konnte.



In der Bibliographie verzeichnete Publikationen in den Jahren