

Anhang A: Verzeichnis der Listings

Listing 1: Ausschnitt zur Veranschaulichung der Syntax	23
Listing 3.1: Implementierung der einfachen main-Methode	35
Listing 3.2: Einfache Deklarationen von Variablen	41 f.
Listing 3.3: Deklarationslisten mit Variablen in einem Ausdruck	43
Listing 3.4: Deklaration und Initialisierung von Variablen	43 f.
Listing 3.5: Deklarationslisten mit optional initialisierten Variablen	44
Listing 3.6: Deklaration und Initialisierung von Konstanten	45
Listing 3.7: Deklarationsliste mit Konstanten in einem Ausdruck	46
Listing 3.8: Verwendung von binären und arithmetischen Operatoren	48 f.
Listing 3.9: Verwendung des unären Minusoperators	50
Listing 3.10: Verwendung von Präfix-Inkrement und Postfix-Dekrement	52
Listing 3.11: Binärer und arithmetischer Zuweisungsoperator	54
Listing 3.12: Klammerung arithmetischer Ausdrücke	55
Listing 3.13: Sequenz von Anweisungen in einem Ausdruck	56
Listing 3.14: Deklaration eines Arrays fester Größe	57
Listing 3.15: Zugriffe auf Elemente eines Arrays über Indizierungsoperator	58
Listing 3.16: Deklaration von Arrays mit Initialisierungslisten	60
Listing 3.17: Verwendung eines zweidimensionalen Arrays fester Größe	61
Listing 3.18: Typdefinitionen mit dem Schlüsselwort typedef	63
Listing 3.19: Verwendung der Schlüsselwörter register und volatile	64
Listing 4.1: Leere Präprozessor-Direktive	67
Listing 4.2: Inkludieren von Header-Dateien	69
Listing 4.3: Einfache Definition eines Präprozessor-Symbols	70
Listing 4.4: Definition eines Präprozessor-Symbols mit zugehörigem Wert	71
Listing 4.5: Verwendung eines Präprozessor-Symbols in einer Deklaration	71
Listing 4.6: Definition mehrerer Präprozessor-Symbole	72

Anhang A: Verzeichnis der Listings (4.7 – 5.19)

Listing 4.7: Entfernen eines Präprozessor-Symbols	73
Listing 4.8: Bedingter Programmcode durch Präprozessor und Symbol	74
Listing 4.9: Bedingter Programmcode durch verschachtelte Direktiven	76
Listing 4.10: Verzweigter Programmcode mit Bedingung im Präprozessor	77
Listing 4.11: Bedingter Programmcode mit der defined -Direktive	78
Listing 4.12: Verzweigter Programmcode mit Bedingung und defined -Direktive	79
Listing 4.13: Bedingung mit vordefiniertem Präprozessor-Symbol	81
Listing 4.14: Benutzerdefinierte Fehlermeldung an den Compiler	81
Listing 4.15: Benutzerdefinierte Veränderung vordefinierter Symbole	82
Listing 4.16: Header-Guard für Inhalt einer beliebigen Header-Datei	84
Listing 5.1: Deklaration von Klassennamen mit class und struct	86
Listing 5.2: Definition einer Klasse ohne Eigenschaften	87
Listing 5.3: Definition von Klassen mit deklarierten Eigenschaften	88
Listing 5.4: Klassendefinition mit Sichtbarkeitsbereichen und Eigenschaften	90 f.
Listing 5.5: Bildung von Instanzen einer Klasse	93
Listing 5.6: Bildung von Instanzen in einer Deklarationsliste	93
Listing 5.7: Bildung von Instanzen bei einer Klassendefinition	94
Listing 5.8: Fehlerhafte Bildung der Instanz einer undefinierten Klasse	94
Listing 5.9: Zugriffe auf Eigenschaften von Instanzen über Punktoperator	95
Listing 5.10: Komplexer Datentyp als Eigenschaft in einer Klassendefinition	96
Listing 5.11: Zugriffe auf Eigenschaften eines eingebetteten Objektes	97
Listing 5.12: Scoping im globalen Bereich einer Übersetzungseinheit	101 f.
Listing 5.13: Deklaration zweier globaler Methoden	105
Listing 5.14: Scoping in globalen Methoden	105 f.
Listing 5.15: Scoping in lokalen Bereichen einer Methode	108
Listing 5.16: Scoping in Klassenbereichen	110
Listing 5.17: Eingebettete Klassenbereiche	111
Listing 5.18: Instanzen eingebetteter Klassen	112
Listing 5.19: Zugriff auf eine globale Variable mit dem Scope-Operator	113

Listing 5.20: Deklaration eines Aufzählungstyps mit dem Schlüsselwort enum	114
Listing 5.21: Definition eines Aufzählungstyps mit dem Schlüsselwort enum	115
Listing 5.22: Definition eines Aufzählungstyps mit initialisierten Enumeratoren	116
Listing 5.23: Bildung und Verwendung von Instanzen zweier Aufzählungstypen	117
Listing 5.24: Aufzählungstyp mit initialisierten und generierten Werten	118
Listing 5.25: Fehlerhaft definierte Aufzählungstypen	119
Listing 5.26: Lebensdauer einer Variablen und Konstanten in Scopes	121
Listing 5.27: Aufrufe einer Methode mit lokaler Lebensdauer einer Variablen	122
Listing 5.28: Lebensdauer eines Objektes im Methodenbereich	123
Listing 5.29: Lebensdauer einer lokalen Variablen mit dem Schlüsselwort static	124
Listing 5.30: Aufrufe einer Methode mit statischer Lebensdauer einer Variablen	125
Listing 5.31: Eigenschaft einer Klasse mit dem Schlüsselwort static	127
Listing 5.32: Zugriffe auf eine statisch deklarierte Eigenschaft	128
Listing 5.33: Eingebettete und statische Eigenschaft einer Klasse	128
Listing 5.34: Konstante und statische Eigenschaft einer Klasse	129
Listing 5.35: Verwendung von Vergleichsoperatoren	132
Listing 5.36: Einfache Kontrollstruktur einer bedingten Anweisung	133
Listing 5.37: Bedingung mit Ausdrücken im lokalen Scope	134
Listing 5.38: Bedingte Anweisungen mit dem Schlüsselwort else	135
Listing 5.39: Verschachtelte Bedingungen	136
Listing 5.40: Einfache Kontrollstruktur einer Zählschleife	137
Listing 5.41: Zählschleife mit leeren Ausdrücken und lokalem Scope	138
Listing 5.42: Zählschleife mit Sequenzen und eingebetteter Bedingung	139
Listing 5.43: Kontrollstruktur einer kopfgesteuerten Schleife mit lokalem Scope	140
Listing 5.44: Kontrollstruktur einer fußgesteuerten Schleife mit lokalem Scope	141
Listing 5.45: Einfache Kontrollstruktur einer Fallunterscheidung	143
Listing 5.46: Einfache Fallunterscheidung durch bedingte Anweisung	144
Listing 5.47: Fallunterscheidung mit mehreren case -Fällen	145
Listing 5.48: Abbrüche von Fallunterscheidungen mit dem Schlüsselwort break	146
Listing 5.49: Fallunterscheidung mit default -Fall und Enumerator	147 f.

Anhang A: Verzeichnis der Listings (5.50 – 7.4)

Listing 5.50: Typdefinition mit einer Klasse	148
Listing 5.51: Verwendung des ternären Operators	150
Listing 5.52: Äquivalente if-else -Kontrollstruktur	150
Listing 5.53: Arrays fester Größen in Verbindung mit Objekten	151 f.
Listing 6.1: Definition von Namensräumen	155
Listing 6.2: Überschreiben des globalen Namensraumes	155
Listing 6.3: Unzulässige Deklaration eines Namensraumes	156
Listing 6.4: Definition verschachtelter Namensräume	156 f.
Listing 6.5: Fehlerhaft verschachtelte Namensräume	158
Listing 6.6: Namenlose und verschachtelte Namensräume	158 f.
Listing 6.7: Explizite Zugriffe auf Daten in Namensräumen	160
Listing 6.8: Globaler Zugriff über den Scope-Operator	161
Listing 6.9: Verwendung der using -Direktive	162
Listing 6.10: Verwendung der using -Direktive mit Scope-Operator	162 f.
Listing 6.11: Explizite Zugriffe auf Daten namenloser Namensräume	164
Listing 6.12: Verwendung der using -Direktive mit namenlosem Namensraum	164
Listing 6.13: Aliasing eines Namensraumes	165
Listing 6.14: Aliasing eines Namensraumes mit using -Direktive	166
Listing 6.15: Aliasing von verschachtelten Namensräumen	166 f.
Listing 6.16: Aliasing von verschachtelten und namenlosen Namensräumen	167
Listing 6.17: Aliasing eines Aliases auf einen Namensraum	168
Listing 6.18: Mehrdeutigkeit von Namensräumen	168 f.
Listing 6.19: Mehrdeutigkeit von Namensräumen über Alias	169 f.
Listing 6.20: Mehrdeutigkeit von Namensräumen durch globale Daten	170
Listing 7.1: Deklaration vorzeichenloser Ganzzahlen	177
Listing 7.2: Deklaration und Initialisierung einer vorzeichenbehafteten Ganzzahl	180
Listing 7.3: Verwendung eines Gleitkommawertes mit NaN und Inf	184
Listing 7.4: Klassendefinition mit innerer Instanz und Array als Eigenschaft	187

Listing 7.5: Bildung von Instanzen zur Verdeutlichung von Speichergrößen	190
Listing 7.6: Verwendung des sizeof -Operators mit primitiven Typen	193
Listing 7.7: Verwendung des sizeof -Operators mit komplexen Typen	194
Listing 7.8: Fehlerhafte Verwendung des sizeof -Operators	194
Listing 7.9: Verwendung des sizeof -Operators mit Klasse ohne Eigenschaften	195
Listing 7.10: Verwendung des sizeof -Operators mit einem Array fester Größe	195 f.
Listing 7.11: Deklaration eines Datenverbundes mit dem Schlüsselwort union	197
Listing 7.12: Definition eines Datenverbundes mit deklarierten Eigenschaften	198
Listing 7.13: Bildung und Verwendung der Instanz eines Datenverbundes	200
Listing 7.14: Anonymer Datenverbund mit dem Schlüsselwort union	202
Listing 7.15: Bitfeld in deklariertem Eigenschaft einer Klasse	203
Listing 7.16: Klassendefinition mit Bitfeldern in Eigenschaften	205
Listing 7.17: Planbarer Speicherbedarf einer Instanz mit Bitfeldern	206
Listing 7.18: Klassendefinition mit Bitfeldern als Flags	207
Listing 7.19: Definition und Instanziierung einer anonymen Klasse	208
Listing 7.20: Bitverschiebung nach links mit einer positiven Ganzzahl	211
Listing 7.21: Bitverschiebung nach rechts mit einer negativen Ganzzahl	212
Listing 7.22: Bitweise Und-Verknüpfung zweier positiver Ganzzahlen	214
Listing 7.23: Bitweise Oder-Verknüpfung zweier negativer Ganzzahlen	216
Listing 7.24: Bitweise Exklusive-Oder-Verknüpfung zweier Ganzzahlen	217
Listing 7.25: Verwendung des bitweisen Einerkomplements	219
Listing 7.26: Bitweise Operationen mit binären Zuweisungsoperatoren	221
Listing 7.27: Über- und Unterläufe ganzzahliger Werte	223
Listing 7.28: Promotion und Demotion eines primitiven Typs	224 f.
Listing 7.29: Implizite Konvertierung primitiver Typen	226
Listing 7.30: Implizite Konvertierung eines Wahrheitswertes	227
Listing 7.31: Fehlerhafte Konvertierung eines Gleitkommawertes	228
Listing 7.32: Implizite Konvertierung einer Ganzzahl nach bool	229
Listing 7.33: bool -Konvertierung in Verbindung mit einer Kontrollstruktur	229
Listing 7.34: Verwendung ganzzahliger Literale mit Suffixen	231

Anhang A: Verzeichnis der Listings (7.35 – 8.21)

Listing 7.35: Verwendung von Gleitkommawerten mit Literalen und Suffixen	232
Listing 7.36: Literale und Suffixe verbunden mit dem sizeof -Operator	233
Listing 7.37: Verwendung von Konstruktoren primitiver Typen	233 f.
Listing 7.38: Verwendung des Konvertierungsoperators	235
Listing 7.39: Logische Und-Verknüpfungen von Wahrheitswerten	236
Listing 7.40: Operator der logischen Oder-Verknüpfung	237
Listing 7.41: Verwendung des unären Negierungsoperators	238 f.
Listing 7.42: Verknüpfte Bedingungen im Präprozessor	240
Listing 8.1: Signaturen von Methoden einer Klasse	242
Listing 8.2: Signatur einer Methode mit dem const -Qualifizierer	242
Listing 8.3: Implementierung von Methoden einer Klasse	244
Listing 8.4: Fehlerhaft implementierte Klassenmethoden	245
Listing 8.5: Prototyp einer Klassenmethode im Namensraum	246
Listing 8.6: Implementierung einer Klassenmethode explizit vom Namensraum	246
Listing 8.7: Implementierung einer Klassenmethode implizit vom Namensraum	247
Listing 8.8: Implementierung einer Klassenmethode direkt im Namensraum	247
Listing 8.9: Eingebettete Klassenmethode mit dem Schlüsselwort inline	248
Listing 8.10: Rückgabe von Werten aus implementierten Klassenmethoden	249 f.
Listing 8.11: Vorzeitiges Beenden von Methoden mit dem Schlüsselwort return	251
Listing 8.12: Rückgabe einer konstanten Instanz einer Klasse im Namensraum	252
Listing 8.13: Parameter von Methoden einer Klasse	253 f.
Listing 8.14: Zugriffe auf Parameter einer Klassenmethode über lokale Kopien	255
Listing 8.15: Definition von Standardparametern in Prototypen von Methoden	256
Listing 8.16: Verwendung von Standardparametern in Methodenaufrufen	257
Listing 8.17: Getter-Methoden und Setter-Methoden	259
Listing 8.18: Deklaration und Implementierung einer logischen Methode	260
Listing 8.19: Der Prototyp eines Konstruktors	262
Listing 8.20: Implementierung eines Konstruktors mit Initialisierungsliste	262
Listing 8.21: Implementierung eines Konstruktors ohne Initialisierungsliste	263

Anhang A: Verzeichnis der Listings (8.22 – 8.51)

Listing 8.22: Eingebetteter Konstruktor mit Initialisierungsliste	263
Listing 8.23: Bildung und Verwendung der Instanz einer Klasse	264
Listing 8.24: Bildung der Instanz einer Klasse über den Standardkonstruktor	264
Listing 8.25: Überschriebener und eingebetteter Standardkonstruktor	265
Listing 8.26: Instanz einer Klasse mit überschriebenem Standardkonstruktor	265
Listing 8.27: Prototyp eines Standardkonstruktors durch Standardparameter	266
Listing 8.28: Verwendung des Standardkonstruktors mit Standardparametern	267
Listing 8.29: Prototyp und Implementierung eines Konstruktors	267
Listing 8.30: Implizite Typkonvertierung durch einen Konstruktor	268
Listing 8.31: Prototyp eines Konstruktors mit dem Schlüsselwort explicit	269
Listing 8.32: Bildung der Instanz einer Klasse über expliziten Konstruktor	269
Listing 8.33: Expliziter und eingebetteter Konstruktor einer Klasse	270
Listing 8.34: Definition einer Klasse mit eingebetteten Objekten	272
Listing 8.35: Konstruktoraufrufe eingebetteter Objekte über Initialisierungslisten	272
Listing 8.36: Prototyp und Implementierung des Destruktors einer Klasse	274
Listing 8.37: Expliziter Aufruf einer Destruktormethode	274
Listing 8.38: Prototyp und Implementierung einer statischen Klassenmethode	276
Listing 8.39: Aufruf einer statischen Methode über den Namen der Klasse	277
Listing 8.40: Aufruf einer statischen Methode über ein Objekt	277
Listing 8.41: Überladene Methoden einer Klasse	279
Listing 8.42: Fehlerhaft überladene Methoden einer Klasse	280
Listing 8.43: Überladene und mehrdeutige Methoden einer Klasse	281
Listing 8.44: Mehrdeutigkeit beim Aufruf von Klassenmethoden	281 f.
Listing 8.45: Eindeutige Aufrufe überladener Klassenmethoden	282
Listing 8.46: Konstruktor-Prototyp mit Standardparametern	283
Listing 8.47: Instanziierung einer Klasse mit Standardparametern im Konstruktor	284
Listing 8.48: Überladene Methode einer Klasse durch Standardparameter	285
Listing 8.49: Überladene Methode einer Klasse mit zwei Parametern	285
Listing 8.50: Mehrdeutigkeit durch Überladung und Parameter	286
Listing 8.51: Berechnung der Fakultät einer Zahl durch Rekursion einer Methode	287

Anhang A: Verzeichnis der Listings (8.52 – 9.18)

Listing 8.52: Berechnung einer Fibonacci-Zahl durch rekursive Methodenaufrufe	288
Listing 8.53: Freundschaft einer Klassenmethode mit dem Schlüsselwort friend	289
Listing 8.54: Zugriff auf private Eigenschaft über befreundete Klassenmethode	290
Listing 8.55: Freundschaft einer Klasse über das Schlüsselwort friend	291
Listing 8.56: Freundschaft einer globalen Methode	292
Listing 8.57: Delegation an Methoden in einer Kontrollstruktur	293
Listing 8.58: Delegation an Methoden in einem arithmetischen Operator	294
Listing 8.59: Deklaration und Verwendung eines konstanten Objektes	296
Listing 8.60: Definition und Verwendung einer Wrapper-Klasse	297
Listing 8.61: Definition eines Datenverbundes mit Methoden	299
Listing 8.62: Implementierung von Methoden eines Datenverbundes	300
Listing 9.1: Primitive Datentypen für die Verwendung mit der Klasse Number	302
Listing 9.2: Typdefinition zur Verwendung mit dem sizeof -Operator	303
Listing 9.3: Aufzählungstyp für die Verwendung mit der Klasse Number	304
Listing 9.4: Datenverbund für die Verwendung mit der Klasse Number	305
Listing 9.5: Definition der Klasse Number	306
Listing 9.6: Prototypen überladener Konstruktoren der Klasse Number	307
Listing 9.7: Prototypen von Getter- und Setter-Methoden der Klasse Number	308
Listing 9.8: Weitere Prototypen von Getter-Methoden der Klasse Number	309
Listing 9.9: Prototypen logischer Methoden der Klasse Number	309
Listing 9.10: Prototyp einer statischen Methode der Klasse Number	310
Listing 9.11: Implementierung von Konstruktoren der Klasse Number	311 f.
Listing 9.12: Implementierung von Getter-Methoden der Klasse Number	313 ff.
Listing 9.13: Implementierung von Setter-Methoden der Klasse Number	315 ff.
Listing 9.14: Implementierung logischer Methoden der Klasse Number	318 f.
Listing 9.15: Implementierung einer statischen Methode der Klasse Number	320
Listing 9.16: Statische und konstante Eigenschaften der Klasse Number	321
Listing 9.17: Globale Initialisierung statischer und konstanter Eigenschaften	322
Listing 9.18: Verwendung des sizeof -Operators mit einem Aufzählungstyp	323

Anhang A: Verzeichnis der Listings (9.19 – 10.26)

Listing 9.19: Verwendung des sizeof -Operators mit einem Datenverbund	324
Listing 9.20: Speicherbedarf einer Instanz der Klasse <code>Number</code>	324
Listing 9.21: Verwendung der Klasse <code>Number</code>	325 f.
Listing 10.1: Deklaration eines eindimensionalen Zeigers	328
Listing 10.2: Deklaration eines zweidimensionalen Zeigers	329
Listing 10.3: Deklaration typloser Zeiger	330
Listing 10.4: Deklaration mehrerer Zeiger in einem Ausdruck	331
Listing 10.5: Verwendung des Adressoperators	331
Listing 10.6: Zeiger in Verbindung mit dem unären Adressoperator	332
Listing 10.7: Zuweisung und Initialisierung von Zeigern	333
Listing 10.8: Deklaration und Initialisierung eines Nullzeigers	333
Listing 10.9: Initialisierung eines Nullzeigers durch Typkonvertierung	334
Listing 10.10: Initialisierung eines mehrdimensionalen Nullzeigers	334
Listing 10.11: Validierung einer Zeigervariablen in bedingten Anweisungen	335
Listing 10.12: Validierung einer Zeigervariablen mit Negierungsoperator	336
Listing 10.13: Deklaration konstanter Zeiger	337
Listing 10.14: Deklaration von Zeigern mit dem Schlüsselwort const	337 f.
Listing 10.15: Dereferenzierung von Zeigern	339
Listing 10.16: Dereferenzierung von Zeigern mit dem Schlüsselwort const	340
Listing 10.17: Deklaration und Initialisierung von Zeigern auf Objekte	341
Listing 10.18: Zeiger in Verbindung mit dem Pfeiloperator	342
Listing 10.19: Deklaration eines Zeigers auf ein konstantes Objekt	343
Listing 10.20: Deklaration konstanter Zeiger auf konstante Objekte	344
Listing 10.21: Verwendung des this -Zeigers	345
Listing 10.22: Verwendung von Typdefinitionen mit Zeigern	346
Listing 10.23: Verwendung des sizeof -Operators mit Zeigern	347
Listing 10.24: Verwendung eines Zeigers als Rückgabewert einer Methode	348
Listing 10.25: Verwendung von Zeigern als Methodenparameter	350 f.
Listing 10.26: Verwendung eines Zeigers als Standardparameter	352

Anhang A: Verzeichnis der Listings (10.27 – 10.56)

Listing 10.27: Dynamische Speicherallokierung mit dem new -Operator	353 f.
Listing 10.28: Allokieren von dynamischem Speicher mit Speicherleck	355
Listing 10.29: Freigabe von Speicher mit dem delete -Operator	356
Listing 10.30: Verwendung des new -Operators und des delete -Operators	357
Listing 10.31: Gefährliche Verwendung des delete -Operators	358
Listing 10.32: Dynamische Speicherallokierung eines Zeigers auf dem Heap	359
Listing 10.33: Speicherfreigaben durch Zeigervariablen	360
Listing 10.34: Gefahr eines Speicherlecks durch new -Operator als Rückgabewert	361 f.
Listing 10.35: Speicherleck durch new -Operator als Parameter einer Methode	362
Listing 10.36: Vermeidung von Speicherlecks in Verbindung mit Methoden	363
Listing 10.37: Gleichheit und Ungleichheit von Adressen in Zeigervariablen	364
Listing 10.38: Vagabundierende Zeiger in Verbindung mit Zeigerkopien	366
Listing 10.39: Zeiger in Verbindung mit eindimensionalen Arrays fester Größen	367 f.
Listing 10.40: Lesender Zugriff auf ein Array fester Größe	368
Listing 10.41: Zeiger mit einem mehrdimensionalen Array fester Größe	369
Listing 10.42: Deklaration eines Array fester Größe verbunden mit einem Zeiger	370
Listing 10.43: Zeiger und Arrays fester Größen als Methodenparameter	372 ff.
Listing 10.44: Dynamische Allokierung eines Arrays mit dem new[] -Operator	376
Listing 10.45: Speicherfreigabe eines Arrays mit dem delete[] -Operator	377
Listing 10.46: Dynamisches Array in Verbindung mit dem sizeof -Operator	378
Listing 10.47: Zugriffe auf Elemente von Arrays über Indizierungsoperator	380 f.
Listing 10.48: Indirektion eines Zeigers über Indizierungsoperator mit Index 0	381
Listing 10.49: Zeiger in Verbindung mit Zeigerarithmetik	384 f.
Listing 10.50: Zeigerarithmetik bezogen auf ein Array fester Größe	387
Listing 10.51: Vermeidung eines Speicherlecks durch den delete[] -Operator	389
Listing 10.52: Allokierung eines zweidimensionalen und dynamischen Arrays	390 f.
Listing 10.53: Zugriffe auf Elemente eines zweidimensionalen Arrays	392
Listing 10.54: Speicherfreigabe eines zweidimensionalen Arrays	393
Listing 10.55: Zeiger als Eigenschaften in einer Praxisklasse	395
Listing 10.56: Bildung einer Instanz mit undefinierten Zeiger-Membem	395

Anhang A: Verzeichnis der Listings (10.57 – 10.86)

Listing 10.57: Initialisierte Nullzeiger im überschriebenen Standardkonstruktor	396
Listing 10.58: Prototyp eines Konstruktors der Praxisklasse Line2D	396
Listing 10.59: Dynamische Bildung innerer Instanzen im Konstruktor	397
Listing 10.60: Instanz der Klasse Line2D mit Speicherlecks	397
Listing 10.61: Freigabe innerer Instanzen im Destruktor	398
Listing 10.62: Verwendung eines inneren Objektes über Zeiger-Member	399
Listing 10.63: Prototyp einer Methode der Praxisklasse Line2D	399
Listing 10.64: Instanzen als Parameter über dereferenzierte Zeiger-Member	400
Listing 10.65: Bildung und Verwendung einer Instanz der Klasse Line2D	400
Listing 10.66: Speicherbedarf einer Instanz mit Zeiger-Membem	401
Listing 10.67: Methoden-Prototyp zur Ermittlung des gesamten Speicherbedarfs	401
Listing 10.68: Bedarf an Speicher innerer Instanzen für äußeres Objekt	402
Listing 10.69: Gesamter Speicherbedarf einer Instanz der Klasse Line2D	402
Listing 10.70: Verwendung des Schlüsselwortes nullptr	406
Listing 10.71: Deklaration und Initialisierung einer Referenz	408
Listing 10.72: Fehlerhafte Deklarationen von Referenzen	409
Listing 10.73: Schreibende und lesende Zugriffe auf Variable über Referenz	409 f.
Listing 10.74: Deklaration und Verwendung einer Referenz auf eine Referenz	411
Listing 10.75: Typdefinition in Verbindung mit einer Referenz	412
Listing 10.76: Referenzvariable mit dem Schlüsselwort const	413
Listing 10.77: Referenz auf konstanten Speicherbereich eines Objektes	413
Listing 10.78: Referenz als Parameter in Signatur einer Methode	414
Listing 10.79: Veränderung einer Variablen über Referenz in Methode	415
Listing 10.80: Vergleichbare Methode mit Zeiger als Parameter	415
Listing 10.81: Konstante Referenz als Parameter einer Methode	417
Listing 10.82: Veränderung der statischen Methode der Klasse Number	418
Listing 10.83: Methodenaufruf mit Referenzen als Parameter	418
Listing 10.84: Konstante Referenzen als Methodenparameter in Praxisklasse	419
Listing 10.85: Rückgabe des Aufrufers über dereferenzierten this -Zeiger	420
Listing 10.86: Verkettung von Methodenaufrufen in einem Ausdruck	421

Anhang A: Verzeichnis der Listings (10.87 – 11.9)

Listing 10.87: Objektkopien mit dem Standard-Kopierkonstruktor	422
Listing 10.88: Flache Kopie eines Objektes mit Zeiger-Membren	423
Listing 10.89: Prototyp eines überschriebenen Kopierkonstruktors	424
Listing 10.90: Implementierung eines Kopierkonstruktors mit flacher Kopie	425
Listing 10.91: Prototyp und Implementierung eines Kopierkonstruktors	425
Listing 10.92: Implementierung eines Kopierkonstruktors für tiefe Kopien	426
Listing 10.93: Aufruf eines Kopierkonstruktors im Konstruktor einer Klasse	427
Listing 10.94: Tiefe Kopie eines Objektes mit Zeiger-Membren	428
Listing 10.95: Objektkopien mit dem Standard-Zuweisungsoperator	429
Listing 10.96: Kopien von Objekten mit dem Schlüsselwort operator	430
Listing 10.97: Prototyp eines überschriebenen Zuweisungsoperators	431
Listing 10.98: Implementierung eines Zuweisungsoperators mit flacher Kopie	432
Listing 10.99: Verkettung von Zuweisungen mit Objektkopien	432
Listing 10.100: Prototyp und Implementierung eines Zuweisungsoperators	433
Listing 10.101: Gefahr vagabundierender Zeiger durch flache Objektkopie	433
Listing 10.102: Implementierung eines Zuweisungsoperators für tiefe Kopien	434
Listing 10.103: Tiefe Objektkopien durch Zuweisungen	435 f.
Listing 10.104: Vagabundierende Referenz auf dynamischen Speicherbereich	437
Listing 10.105: Vagabundierende Referenz auf temporären Rückgabewert	437
Listing 10.106: Vermeidung einer vagabundierenden Referenz durch static	438
Listing 11.1: Array fester Größe mit druckbaren Zeichen in Initialisierungsliste	440
Listing 11.2: Initialisierung einer Zeigervariablen mit Zeichenketten-Literal	441
Listing 11.3: Initialisiertes Array fester Größe mit Zeichenketten-Literal	442
Listing 11.4: Bildung und Verwendung einer dynamischen Zeichenkette	443
Listing 11.5: Zeiger auf Array fester Größe mit Zeichenketten-Literalen	444
Listing 11.6: Dynamische Zeichenkette mit terminierter Null	446
Listing 11.7: Globale Methode zur Ermittlung der Länge von Zeichenketten	447
Listing 11.8: Array fester Größe mit Zeichenketten-Literal und Größenangabe	448
Listing 11.9: Verwendung von Zeichenketten-Literalen in Deklarationen	449 f.

Anhang A: Verzeichnis der Listings (11.10 – 12.12)

Listing 11.10: Verwendung von Zeichenketten-Literalen in Parametern	451
Listing 11.11: Vergleich druckbarer Zeichen	452
Listing 11.12: Vergleich einer Zeichenkette mit Operator der Gleichheit	453
Listing 11.13: Globale Methode zur Prüfung zweier Zeichenketten auf Gleichheit	453 f.
Listing 11.14: Prototyp einer neuen Getter-Methode der Klasse Number	455
Listing 11.15: Implementierung der neuen Getter-Methode der Klasse Number	457
Listing 11.16: Verwendung der neuen Getter-Methode der Klasse Number	459
Listing 11.17: Zeichenketten-Literale verbunden mit dem Präprozessor	460
Listing 11.18: Verwendung von Parametern in der main-Methode	461 f.
Listing 12.1: Ableitung zweier Klassen von einer gemeinsamen Basisklasse	465
Listing 12.2: Implementierung einer Methode der Basisklasse	466
Listing 12.3: Bildung und Verwendung von Instanzen abgeleiteter Klassen	467
Listing 12.4: Deklaration von Klassennamen in Verbindung mit Vererbung	467
Listing 12.5: Vererbung von Sichtbarkeit mit private und protected	469
Listing 12.6: Zugriffe auf geerbte und gekapselte Eigenschaft	470
Listing 12.7: Vererbung mit den Schlüsselwörtern class und struct	471
Listing 12.8: Definition der abgeleiteten Praxisklasse Point3D	472
Listing 12.9: Eigenschaften und Sichtbarkeit in Vererbungshierarchie	474
Listing 12.10: Speicherbedarf von der Instanz einer geerbten Klasse	475
Listing 12.11: Fehlerhafter Konstruktoraufruf der Basisklasse über Instanz	476
Listing 12.12: Aufruf eines Basiskonstruktors in Initialisierungsliste	477
Listing 12.13: Prototypen von Konstruktoren der Praxisklasse Point3D	477
Listing 12.14: Aufrufe von Basiskonstruktoren der Klasse Point2D	478
Listing 12.15: Bildung von Instanzen der abgeleiteten Klasse Point3D	479
Listing 12.16: Definition einer abstrakten Basisklasse	481
Listing 12.17: Aufrufe öffentlicher Methoden über Instanz einer Kindklasse	483
Listing 12.18: Prototypen von Methoden der Praxisklasse Point3D	484
Listing 12.19: Getter- und Setter-Methode der Praxisklasse Point3D	484
Listing 12.20: Überschreiben einer Methode mit neuer Sichtbarkeit in Kindklasse	485

Anhang A: Verzeichnis der Listings (12.21 – 12.43)

Listing 12.21: Aufruf einer überschriebenen Methode der Kindklasse	486
Listing 12.22: Aufruf einer geerbten Methode über Instanz der Kindklasse	487
Listing 12.23: Prototyp einer überschriebenen Methode der Klasse Point3D	487
Listing 12.24: Implementierung einer Methode mit Aufruf der Basismethode	488
Listing 12.25: Methodenaufruf über Instanz der Kindklasse	488
Listing 12.26: Überladene Methoden in abgeleiteter Kindklasse	489
Listing 12.27: Aufrufe überladener Methoden über Instanz der Kindklasse	490
Listing 12.28: Geerbte und statische Eigenschaft in Kindklassen	491
Listing 12.29: Zugriffe auf geerbte und statische Eigenschaft	491
Listing 12.30: Aufruf einer geerbten und statischen Methode	492
Listing 12.31: Polymorpher Zeiger auf dynamische Instanz der Kindklasse	493
Listing 12.32: Prototyp einer Klassenmethode mit dem Schlüsselwort virtual	494
Listing 12.33: Überschriebene Methode mit dem Schlüsselwort virtual	495
Listing 12.34: Aufruf einer virtuellen Methode über polymorphen Zeiger	495
Listing 12.35: Speicherbedarf einer Instanz mit verborgenem Zeiger	497
Listing 12.36: Destruktor einer Klasse mit dem Schlüsselwort virtual	497
Listing 12.37: Prototyp und Implementierung eines virtuellen Destruktors	498
Listing 12.38: Abstrakte Basisklasse durch reinvirtuelle Methode	499
Listing 12.39: Fehlerhafte Bildung der Instanz einer abstrakten Klasse	499
Listing 12.40: Polymorpher Zeiger als Parameter einer Methode	500
Listing 12.41: Polymorphie verbunden mit einer Referenz	501
Listing 12.42: Polymorphe Referenzen als Parameter einer Methode	501
Listing 12.43: Finalisierte Praxisklassen mit dem final -Spezifizierer	502

Anhang B: Verzeichnis der Abbildungen

Abbildung 1: Compilieren von Quelldateien	27
Abbildung 2: Linken von Objektdateien	28
Abbildung 3: Instanziierung primitiver und komplexer Datentypen	30
Abbildung 4: Darstellung der Kapselung von Daten	31
Abbildung 5: Präprozessor beim Übersetzen von Quellcodes	65
Abbildung 6: Problem mehrfach eingebundener Header-Dateien	83
Abbildung 7: Abarbeitung von Methoden im Programmstapel	107
Abbildung 8: Sichtbarkeit von Variablen in Scopes	109
Abbildung 9: Verschachtelte Namensräume	157
Abbildung 10: Namenlose und verschachtelte Namensräume	159
Abbildung 11: Interne Darstellung einer vorzeichenlosen Ganzzahl	176
Abbildung 12: Interne Darstellung standardisierter Gleitkommatypen	182
Abbildung 13: Signifikante Bit eines Speicherbereiches (MSB und LSB)	185
Abbildung 14: Interne Darstellung von Objekten in Verbindung mit Füllbyte	191
Abbildung 15: Verhältnis von Speicherausrichtung zu Speicherzugriffen	191
Abbildung 16: Organisation von Elementen eines Datenverbundes im Speicher	199
Abbildung 17: Interne Darstellung eines Speicherbereiches durch ein Bitfeld	204
Abbildung 18: Anordnung von Elementen mit Bitfeldern im Speicher	206
Abbildung 19: Linke Bitverschiebung mit einer vorzeichenbehafteten Ganzzahl	210
Abbildung 20: Rechte Bitverschiebung mit einer vorzeichenbehafteten Ganzzahl	212
Abbildung 21: Darstellung der bitweisen Und-Verknüpfung zweier Ganzzahlen	214
Abbildung 22: Darstellung der bitweisen Oder-Verknüpfung zweier Ganzzahlen	215
Abbildung 23: Bitweise Exklusive-Oder-Verknüpfung zweier Ganzzahlen	216
Abbildung 24: Darstellung des bitweisen Einerkomplements	218
Abbildung 25: Über- und Unterläufe ganzzahliger Wertebereiche	222
Abbildung 26: Distanz zweier Punkte im zweidimensionalen Koordinatensystem	278
Abbildung 27: Eindimensionaler Zeiger auf eine Hardware-Adresse	328
Abbildung 28: Zweidimensionaler Zeiger auf Hardware-Adresse	329

Anhang B: Verzeichnis der Abbildungen (29 – 44)

Abbildung 29: Lokale Kopie eines Zeigers in einer Methode	349
Abbildung 30: Dynamische Speicherallokierung eines Zeigers auf dem Heap	360
Abbildung 31: Vagabundierende Zeiger in Verbindung mit Zeigerkopien	365
Abbildung 32: Zeiger auf verschiedene Speicherbereiche	379
Abbildung 33: Zeiger in Verbindung mit Zeigerarithmetik	386
Abbildung 34: Zeiger verbunden mit einem zweidimensionalen Datenfeld	391
Abbildung 35: Flache Kopien verbunden mit Zeiger-Membem	403
Abbildung 36: Tiefe Kopien in Verbindung mit Zeiger-Membem	404
Abbildung 37: Referenz auf deklarierte Variable	407
Abbildung 38: Übergabe eines Objektes durch call-by-value und call-by-reference	416
Abbildung 39: Beispiel der internen Darstellung von Zeichenketten-Literalen	450
Abbildung 40: Beispiel zur bitweisen Prüfung binärer Zustände	456
Abbildung 41: Ableitung zweier Klassen von einer gemeinsamen Basisklasse	463
Abbildung 42: Darstellung einer Vererbungshierarchie	473
Abbildung 43: Konstruktoraufrufe im Programmablauf	480
Abbildung 44: Destruktoraufrufe im Programmablauf	482
Abbildung 45: Polymorpher Zeiger auf Speicher einer Instanz der Kindklasse	496

Anhang C: Verzeichnis der Tabellen

Tabelle 1: Ganzzahlige Datentypen der Sprache C++	37
Tabelle 2: Gleitkommatypen der Sprache C++	40
Tabelle 3: Binäre und arithmetische Operatoren	47
Tabelle 4: Operatoren für unäre Vorzeichen	50
Tabelle 5: Inkrement und Dekrement in Verbindung mit Präfix und Postfix	51
Tabelle 6: Binäre und arithmetische Zuweisungsoperatoren	53
Tabelle 7: Wertetabelle eines konstanten Arrays fester Größe	60
Tabelle 8: Wertetabelle eines zweidimensionalen Arrays fester Größe	62
Tabelle 9: Präprozessor-Direktiven verbunden mit bedingten Programmcodes	79
Tabelle 10: Vordefinierte Präprozessor-Symbole	80
Tabelle 11: Sichtbarkeitsbereiche in Klassendefinitionen	90
Tabelle 12: Übersicht definierbarer Bereiche (Scopes)	112
Tabelle 13: Binäre Vergleichsoperatoren	131
Tabelle 14: Punktoperator, Scope-Operator und ternärer Operator	152
Tabelle 15: Speichergrößen primitiver und ganzzahliger Datentypen	174
Tabelle 16: Binäre Codierung vorzeichenbehafteter Ganzzahlen	179
Tabelle 17: Speichergrößen von Gleitkommatypen	181
Tabelle 18: Übersicht bitweiser Operatoren	209
Tabelle 19: Wertetabellen für bitweise Verknüpfungen	217
Tabelle 20: Binäre und bitweise Zuweisungsoperatoren	220
Tabelle 21: Übersicht der Suffixe zur Verwendung mit Literalen	230
Tabelle 22: Wertetabellen für logische Verknüpfungen	238
Tabelle 23: Operatoren der Aussagenlogik	239
Tabelle 24: Primitive Datentypen für die Verwendung mit der Klasse Number	302
Tabelle 25: Enumeratoren und Datentypen für die Klasse Number	304
Tabelle 26: Zeigervariablen und Arrays fester Größen als Methodenparameter	371
Tabelle 27: Operatoren der Zeigerarithmetik	382

Anhang C: Verzeichnis der Tabellen (28 – 30)

Tabelle 28 : Operatoren in Verbindung mit Zeigern	405
Tabelle 29 : Weitere Operatoren für die Verwendung mit Zeigern	406
Tabelle 30 : Schlüsselwörter in Verbindung mit der Vererbung von Sichtbarkeit	468