

Index

- |, Bitweiser Operator, 52
- ||, logische Operatoren, 52
- ()
 - (), Cast-Operator, 95
 - Cast-Operator, 142, 292
- ()
 - Begrenzer, 27
 - Funktionsaufrufoperator, 45, 76
 - Operator, 202
 - Typumwandlungsoperator, 46
- Cast-Operator, 49
- *
- Begrenzer, 29, 92, 101
- Indirektionsoperator, 47, 94, 107
- Multiplikativer Operator, 49
- +
- Binärer Operator, 50
- unärer Operator, 47
- ,
- Begrenzer, 27
- Kommaoperator, 54
-
- Binärer Operator, 50
- unärer Operator, 47
- >, Elementzeigeroperator, 46, 49, 118, 214
- ., Elementoperator, 46, 118
- .*, Elementzeigeroperator, 49, 214
- ..., Ellipse, 29, 85, 315
- /, Multiplikativer Operator, 49
- /* */, Kommentar, 31
- //, Kommentar, 31
- ::, Scope-Operator, 45, 123, 125
- ;, Begrenzer, 27
- <, Relationaler Operator, 51
- <<
 - Ausgabeoperator, 206, 207
 - Bitweiser Schiebeoperator, 50
- <=, Relationaler Operator, 51
- =
 - Begrenzer, 29
 - Operator, 203, 209
 - Zuweisungsoperator, 210
- ==, Gleichheitsoperator, 51
- >, Relationaler Operator, 51
- >=, Relationaler Operator, 51
- >>
 - Bitweiser Schiebeoperator, 50
 - Eingabeoperator, 206, 208
- ?:, Bedingungsoperator, 53
- []
 - Begrenzer, 27
 - Indexoperator, 45, 103, 201
- #elif, 340
- #else, 340
- #endif, 340
- #error, 338
- #if, 340
- #ifdef, 340
- #ifndef, 340
- #include, 338
- #line, 339
- #pragma, 339
- %, Multiplikativer Operator, 49
- &
 - Adressoperator, 47, 93, 104, 209
 - Begrenzer, 29, 89, 176, 201
 - Bitweiser Operator, 52
- &&, logischer Operator, 36, 52
- _, *Underscore*, 19
- ~, Bit-Komplement-Operator, 48
- !, logischer Operator, 36, 48
- !=, Ungleichheitsoperator, 51
- !=, Zuweisungsoperator, 54
- ++
 - Postfix-Inkrement-Operator, 46
 - Präfix-Inkrement-Operator, 47
- - Postfix-Dekrement-Operator, 46

- Präfix-Dekrement-Operator, 47
- `^`, Bitweiser Operator, 52
- `for`-Anweisung, 67
- Call by Reference*, 112
- Call by Value*, 112
- `{}`, Begrenzer, 29
- 0, Ende von Zeichenketten, 109

- Abhängigkeit, 155, 162
- Ableiten, *siehe* Vererbung
- Abstrakte Basisklasse, *siehe* Polymorphismus
- Abstraktion und Klassen, 159
- Abweisungsschleife, 66
- Acht-Damen-Problem, Programm, 8
- Adresse, 91
- Aktualparameter, *siehe* Funktionen
- Alias, *siehe* Referenzen
- `Animal`, Übungsbeispiel, 296
- Annotated Reference Manual (ARM)*, 8
- ANSI, 7
- ANSI-Standard, 1, 2, 313
- Anweisungen, 57–71
 - Ausdrucksanweisung, 58
 - Blockanweisung, 59
 - `break`, 69
 - `case`, 64
 - `continue`, 70
 - Deklaration, 61
 - `do ... while`, 67
 - `for`, 67
 - Funktionsaufruf, 58, 76
 - `goto`, 70
 - `if ... else`, Selektion, 61
 - Iteration, 65–69
 - Nullanweisung, 58
 - `return`, 70, 74
 - Selektion, 61–65
 - Sprunganweisungen, 69–71
 - Sprungmarken, 59, 64
 - `switch`, 64
 - `switch` versus `CASE`, 64
 - `while`, 66
 - zusammengesetzte, 59
 - Zuweisung, 58
- Arbeitsteilung, 147
- Arrays, *siehe* Vektor
- Arten von Ausdrücken, 41
- `assert`, 201, 339
- Assoziativität, 41

- Aufrufreihenfolge, 263
- Aufzählungstyp, *siehe* `enum`
- Ausdrücke, 41–55
 - Andere, 49
 - Arten von, 41
 - Auswertungsreihenfolge, 41
 - Kurzschlussauswertung, 52
 - Nebeneffekte bei der Auswertung, 43
 - Postfix, 45
 - Primäre, 44
 - Unäre, 47
- Ausdrucksanweisung, 58
- Ausgabeoperator `<<`, 207
- Ausnahme, 309
- Ausnahmebehandlung, 309–322
 - Auslösen von Ausnahmen, 312
 - Ausnahmeklassen, 313
 - Ausnahmespezifikation, 316
 - `bad.exception`, 314, 317
 - `catch`, 311
 - `domain.error`, 314
 - `Ellipse ...`, 315
 - `exception`, 313
 - Exception Propagation*, 315
 - Handler, 310
 - `invalid.argument`, 314
 - Laufzeitfehler, 313
 - `length.error`, 314
 - `logic.error`, 314
 - Logischer Fehler, 313
 - `out.of.range`, 314
 - `overflow.error`, 314
 - Praxis, 318
 - Prinzip, 310
 - `range.error`, 314
 - Richtlinien, 319
 - `set.unexpected`, 317
 - Stack Unwinding*, 312
 - Standardausnahmen, 313
 - `throw`, 311–313
 - `throw()`, 316
 - `try`, 311
 - Unbehandelte Ausnahmen, 317
 - Unerwartete Ausnahmen, 317
 - `unexpected`, 317
 - `unexpected-Handler`, 317
- Ausprägung
 - Funktions-Template, 232
 - Klassen-Template, 237
- Auswertungsreihenfolge, 41

- Auswirkungen, 276
- auto, 130
- Automatisch generierte
 - Elementfunktionen, 208
- Backslash*, 24
- bad_exception, *siehe*
 - Ausnahmebehandlung
- Basisdatentypen, 33–40
- Basisklasse, *siehe* Vererbung
- Basisklassen
 - Spezifikation von, 269
 - Typumwandlungen, 141
- Batman, Klasse, 264
- Baum, binärer, 248
- Begrenzer, 27–30
 - (), 27
 - *, 29, 92, 101
 - ..., 29, 85, 315
 - ::, 27
 - =, 29
 - [], 27
 - &, 89
 - &, 29, 176, 201
 - ,, 27
 - !, 27
 - {}, 29
 - Alternative Symbole, 30
- Benutzerdefinierte Typumwandlung,
 - siehe* Typumwandlung
- Benutzt-Beziehung, *siehe* Vererbung
- Bezeichner, 19
- Bezugsrahmen, 121
- Binden (Linken), 10
- binSearch, Übungsbeispiel, 244
- bit field, 216
- Bitfelder, *siehe* bit field
- Bits, Löschen von, 52
- Bits, Setzen von, 52
- Block, 59
- Blockanweisung, 59, 62
- bool, 35, 141
- Boolesche Werte (Literele), 25
- break-Anweisung, 69
- C with Classes, 8
- C++-Programme, einfache, 9–16
- Calculator
 - Übungsbeispiel, 220
- Calculator
 - Schnittstelle, 222
- Call by Reference*, *siehe* Parameter
- Call by Value*, *siehe* Funktionen
- case, Marke, *siehe* Anweisungen
- Case Sensitive*, 20
- Cast-Operator (), 49, 95, 142, 292
- catch, 311
- CellWar, Übungsbeispiel, 297
- Chaining*, 263, 265, 266
- char, 36
- cin, *siehe* iostream
- class, *siehe* Klassen
- Class Scope*, 122
- Collection Class*, 183
- ComicCharacter, Klasse, 254, 286
- Comicfigurenliste, Programm, 277
- const, 39, 133, 171
 - Elementfunktion, 171
 - und mutable, 173
 - und Zeiger, 99
- const_cast,
 - Typumwandlungsoperator, 46, 143
- continue-Anweisung, 70
- Conversion, *siehe* Typumwandlung
- Copy-Konstruktor, *siehe* Konstruktor
- Counter, Übungsbeispiel, 217
- cout, *siehe* iostream
- Covarianter Rückgabewert, 279
- Dangling Pointer*, 98, 193
- Date, Übungsbeispiel, 217
- Datei globs.h, 227
- Dateipositionen, 332
- Datenkapsel, 147–157
 - in C++, 149
- Datentypen
 - bool, 35
 - char, 36
 - double, 37
 - enum, 38
 - float, 37
 - höhere, *siehe* Höhere
 - Datentypen
 - int, 36
 - long, 36
 - long double, 37
 - short, 36
 - void, 38
 - wchar_t, 36
- De-Initialisierung, *siehe* Destruktor
- Deep Copy*, 193, 194, 204

- Default*-Argumente, *siehe* Funktionen
- Default*-Konstruktor, 187
- `default`, Sprungmarke, 64
- Definition
 - Begriff, 33
 - One Definition Rule, 34
- Deklarationen, 33–40, 128–136
 - Anweisung, 61
 - Funktionsattribute, *siehe* Funktionsattribute
 - Initialisierung, *siehe* Initialisierung
 - Speicherklassenattribute, *siehe* Speicherklassenattribute
 - `typedef`, *siehe* `typedef`
 - Typqualifikatoren, *siehe* Typqualifikatoren
- `delete`, Operator, 49, 97, 202
- Destruktor, 168, 195
 - bei abgeleiteten Klassen
 - Chaining*, 265
 - Aufrufreihenfolge, 265
 - virtueller, 196, 210
- Dezimalzahl (Literal), 22
- `do ... while`-Anweisung, 67
- `domain_error`, *siehe* Ausnahmebehandlung
- Doppelpunkt `:`, Begrenzer, 27
- `double`, Datentyp, 37
- Duck, Klasse, 269
- Durchlaufschleife, *siehe* `do ... while`
- `dynamic_cast`,
 - Typumwandlungsoperator, 46, 145, 293
- Dynamische Bindung, 275–308
- Dynamische Methodenauflösung, 279

- Early Binding*, 275
- Effizienz, 147
- Ein-/Ausgabe
 - `iostream`, 10
- Ein-/Ausgabe über *Streams*, 323
- Eingabeoperator `>>`, 208
- Eingangsparameter, 114
- Einrückung, 62
- Einser-Komplement, 48
- Element, Klasse, 160
- Elementfunktion, 168–175, 185–213
 - automatisch generierte, 208
 - bei abgeleiteten Klassen, 262–269
 - `const`-Elementfunktion, 171, 173
- Destruktoren, *siehe* Destruktor
- Hilfsfunktion, 169
- Hintereinanderausführung, 175
- Implementierungsfunktion, 169
- `inline`, 169
- Iteratoren, 169
- Klassifikation, 168
- Konstruktor, *siehe* Konstruktor
- Managerfunktion, 169
- Modifikatoren, 169
- Operatoren, *siehe* Operatoren
- Rein virtuelle, 285
- Selektoren, 169
- `static const`, 181
- Überschreiben ererbter, 267
- Überschreiben nicht virtueller, 281
- virtuelle, 254, 276
 - Aufruf, 279
 - Deklaration, 278
 - Konstruktor, 280
- virtueller Destruktor, 281
- Zugriffsfunktion, 169
- Elementzeigeroperator `->`, 49, 214
- Elementzeigeroperator `.*`, 49, 214
- Element-Programmierrichtlinien, 63
- Ellipse, 29, 85, 315
- `else`-Zweig, 61
- Empfehlungen
 - 0 versus `NULL`, 95
 - Adressen von temporären Objekten, 93
 - Ausdrücke und Klammerung, 43
 - Ausnahmebehandlung, 319
 - Bedingte Bewertungen, 54
 - Beispiel, 5
 - Datenelemente und `mutable`, 175
 - Default-Zweige in `switch`-Anweisungen, 65
 - Deklarationen, 34
 - Elementfunktionen, 162
 - Elementfunktionen,
 - automatischer Aufruf, 187
 - Elementfunktionen, Definition in Schnittstelle, 171
 - Elementfunktionen, Definition und `inline`, 170

- Elementfunktionen, Operatoren, 200
- Elementfunktionen, Selektoren (`const`), 173
- Explizite Konstruktoren, 192
- Funktionen, Überladen, 85
- Geheimnisprinzip und `friend`, 180
- Größe der Basisdatentypen, 37
- Indexprüfungen, 103
- Initialisierung versus Zuweisung, 30
- Initialisierung von Objekten (Initialisierungslisten), 189
- Klassen, `protected`, 258
- Klassen, automatisch erzeugte Elementfunktionen, 209
- Klassen, Geheimnisprinzip, 257
- Klassen, Geheimnisprinzip (Sichtbarkeit), 167
- Klassen, getrennte Übersetzung (Schnittstellen), 167
- Klassen, kanonische Form, 206, 209, 210
- Klassen, Reihenfolge der Elemente, 179
- Klassen, Speicherverwaltung, 196
- Klassen, Speicherverwaltung versus Initialisierung und De-Initialisierung, 196
- Klassen, Vollständigkeit der Schnittstellen, 168
- Kommaoperator, 69
- Kommentare, Arten, 31
- Kommentare, sparsame Verwendung, 32
- Konstruktoren, `Copy`-Konstruktoren, 195
- Konstruktoren, Sichtbarkeit, 191
- Makronamen, 336
- Makros und `inline`, 81
- Namensräume und Gültigkeitsbereiche, 128
- Namensräume und interne Objekte (`static`), 131
- Objekte, Gültigkeitsbereich, 75
- Optimierungen, 81
- Parameterübergabe, Vektoren, 109
- Parameterbelegung und getrennte Übersetzung, 83
- Rückgabewerte, Referenzierung lokaler Objekte, 117
- Rückgabewerte, Zeiger und Referenzen, 116
- Referenzen, 90, 91
- Rekursion, 88
- Schleifen, `continue`-Anweisung, 70
- Schleifenarten, 66
- Schleifenarten, `do while`, 67
- Schleifenvariablen, lokale Deklaration, 68
- Speicherverwaltung, 98
- Speicherverwaltung mit `new` und `delete`, 105
- Standardbibliothek, 186
- Templates, Ausprägung von Funktions-Templates, 232, 233
- Templates, Ausprägungszeitpunkt, 244
- Typumwandlung, 143
- Typumwandlung und `const_cast`, 144
- Vektoren, Indirektionsoperatoren, 108
- Vektoren, Zeiger, 104
- Vererbung, `virtual`, 268, 279
- Vererbung, *Run-Time Type Information*, 293
- Vererbung, nicht virtuelle Methoden, 282
- Vererbung, virtuelle Destruktoren, 281
- Verwendung von `const`, 40
- Verwendung von `const`, versus `#define`, 40
- Verwendung von `goto`s, 71
- Verwendung von `l`, `long`-Zahlen, 23
- Zeiger, 93
- Enthält-Beziehung, *siehe* Vererbung
- `enum`, Datentyp, 38
- Error*, 309
- Escape-Sequenz, 24
- `exception`, *siehe* Ausnahmebehandlung

- Exception Handling, *siehe*
 - Ausnahmebehandlung
- explicit und Konstruktoren, 191
- explicit, 211
- Explizite Qualifizierung, 232
- Explizite Typumwandlung, *siehe*
 - Typumwandlung
- Exponent, 37
- Export von Daten, 162
- extern, 132, 133

- false, 25
- Fehler, 309
- Felder, *siehe* Vektor
- Fließkommadatentyp, 37
- Fließkommakonstanten, 40
- Fließkommazahlen, 23
- float, Datentyp, 37
- for-Anweisung, 67
- Formalparameter, *siehe* Funktionen
- friend, 179, 199
 - Template-Klassen, 238
 - Vererbung, 257
- Function Casts* und Konstruktoren, 191
- Funktion, 73–88
 - Aufruf, 76
 - Definition, 74, 75
 - Deklaration, 74
 - Deklaration, 74
 - inline, 80
 - Namensauflösung, 84
 - Operatorfunktion, 80, 198
 - Parameter, 73, 76
 - Default*-Argumente, 81
 - Aktualparameter, 76
 - Eingangsparameter, 114
 - Formalparameter, 76
 - Vektoren, 108
 - Parameterübergabe, 112–115
 - Call by Value*, 261
 - Call by Reference*, 112
 - Call by Value*, 112
 - Gegenüberstellung der Möglichkeiten, 113
 - Referenzen, 113
 - Zeiger, 113
 - Prototyp, 74
 - Prozedur, 79
 - Rückgabewert, 73, 176, 201
 - Rekursion, 86
 - Einsatz, 87
 - Signatur, 76, 83, 232
 - Templates, *siehe* Templates
 - Typ, 76
 - Überladen, 83, 84
 - Überladen versus vorbelegte Parameter, 85
- Funktion-Zeiger-Typumwandlungen, 139
- Funktionsattribute, 136
 - inline, *siehe* inline
 - virtual, *siehe* virtual
- Funktionsaufruf, 58
- Funktionsdefinition, 74
- Funktionsdeklaration, 74
- Funktionsnotation von Operatoren, 198

- Ganzzahldatentyp, 36
- Garbage Collection*, 98
- Gebundenes Template, 238
- Geheimnisprinzip, 162
- Generische Funktion, 230
- Geschachtelte Klassen, 183
 - Templates, 238
- Getrennte Übersetzung, 241
- Gleitkomma-Integral-Typumwandlung, 140
- Gleitkomma-Promotion, 139, 140
- Gleitkomma-Typumwandlung, 140
- goto-Anweisung, 70
- Grammatik, 17
- Gültigkeit, 133
- Gültigkeitsbereich, 60, 121, 122, 315
 - Block, 122, 130
 - Funktion, 122
 - Klasse, 122
 - Namensraum, 122, 127
 - Substatement*, 63, 66
 - Vererbung, 261
- Gültigkeitsbereich & Sichtbarkeit, Programm, 218
- Gültigkeitsbereich, Übungsbeispiel, 218

- Höhere Datentypen, 89–119
 - Referenzen, *siehe* Referenzen
 - Vektoren, *siehe* Vektor
 - Zeiger, *siehe* Zeiger
- Handler, 310
 - set.unexpected, 317

- Handler, 315
 - unexpected, 317
- Has-Beziehung, *siehe* Vererbung
- Hello world, Programm, 10
- Hexadezimalzahl (Literal), 22
- Hilfsfunktion, 169
- Hintereinanderausführung von Methoden, 175
- if ... else, Selektion, 61
- if und Blockanweisungen, 62
- Implementierungsfunktion, 169
- Implementierungsvererbung, 270
- Initialisierung, 137
 - Objekte, 186, 188
 - versus Zuweisung, 30
- Initialisierungsliste, 188
- inline, 136
 - inline-Funktionen, 80
 - Elementfunktion, 169
- int, Datentyp, 36
- Integerkonstanten, 40
- Integertyp, 35
- Integral-Promotion, 139
- Integral-Typumwandlungen, 140
- Integraler Typ, 35
- Interaktive Testtreiber, 155
- Interface-Vererbung, 270, 282
- invalid.argument, *siehe* Ausnahmebehandlung
- iostream, 10
 - iostream, cin, 11
 - iostream, cout, 11
- is-A-Beziehung, *siehe* Vererbung
- ISO-Standard, 1, 2, 313
- istream, 325
- Iteration, 57, 65–69
- Iterationsanweisung
 - for, 67
- Iteratoren, 169, 183, 246
 - Robustheit, 247
- Kanonische Form von Klassen, 209
- Kellerbar, Übungsbeispiel, 303
- Klasse
 - Abstraktion, 159
 - Ausgabeoperator <<, 207
 - Benutzerdefinierte Typumwandlung, *siehe* Typumwandlung
 - Bitfelder (bit field), 216
 - Calculator, Schnittstelle, 222
 - ComicCharacter, 254, 286
 - Definition, 117
 - Duck, 269
 - Eingabeoperator >>, 208
 - Element, 160
 - Elementfunktion, *siehe* Elementfunktion
 - explicit, 211
 - friend, *siehe* friend
 - Gültigkeitsbereich, *siehe* Gültigkeitsbereich Klasse
 - Geheimnisprinzip, 162
 - Geschachtelte, 183
 - Implementierung, 164
 - Initialisierung, 186
 - inline-Elementfunktion, 169
 - Kanonische Form, 209
 - Klienten, 255
 - MapSite, 302
 - Methoden, 162
 - Objekt, 160
 - Polymorphe Klasse, 278
 - private, 163
 - protected, 163
 - public, 117
 - public, 163
 - Rational, Schnittstelle, 225
 - Schnittstelle, 164
 - SimObject, Schnittstelle, 307
 - Simulation, Schnittstelle, 306
 - Stack, Ausgabeoperator, 207
 - Stack, Eingabeoperator, 208
 - Stack, Implementierung, 165, 177
 - Stack, mit const-Elementfunktionen, 171
 - Stack, Schnittstelle, 164, 176
 - Stack-Template, Implementierung, 236
 - Stack-Template, Schnittstelle, 235
 - static const-Elemente, 181
 - static-Elemente, 180
 - static-Elementfunktionen, 181
 - SuperHero, 255, 268
 - SuperHero, Superman und Batman, 264
 - Templates, *siehe* Templates

- `this`, 175
- Typumwandlung mit
 - Konstruktoren, 211
- Typumwandlung mit
 - Umwandlungsoperatoren, *siehe* Typumwandlung
- `union`, 118
- Varianten (`union`), 215
- `Vehicle`, 170
- `Vehicle` mit `const-`
 - Elementfunktionen, 172
- Vererbung, *siehe* Vererbung
- `xcpt`, 311
- Zeiger auf Klasselemente, 213
- Zugriffsschutz, 178
- Zuweisungsoperator, 203, 209
- Klassen, 117, 159–216
- Klassentopologie, *siehe* Vererbung
- Klassifikation von Elementfunktionen, 168
- Klienten, von Klassen, 255
- Kommaoperator und `for`, 68
- Kommentare, 31
- Kompilieren, 9
- Konstanten, 39
- Konstruktor, 168, 186
 - Copy*-Konstruktor, 192, 209, 210, 263
 - Deep Copy*, 193
 - Shallow Copy*, 193
 - Default*-Konstruktor, 186, 187, 209, 210, 263
- Aufruf, 187
- bei abgeleiteten Klassen, 262
 - Chaining*, 263
 - Aufrufreihenfolge, 263
 - Copy*-Konstruktor, 263
- `explicit`, 191, 211
- Function Casts*, 191
- Initialisierung durch Zuweisung, 188
- Initialisierungsliste, 188
- Sichtbarkeit, 190
- Standard-*Copy*-Konstruktor, 193
- Typumwandlung, 200, 211
- Überladen, 189
- Kontrakte, *siehe Programming by Contract*
- Kurzschlussauswertung, 52
- Löschen von Bits, 52
- Lösungen, 6
- Label*, *siehe* Sprungmarken
- Late Binding*, *siehe* Vererbung
- Laufzeit-Typinformation, 293
 - `dynamic_cast`, 293
 - `typeid`, 294
- Erweiterung und Einsatz, 295
- Laufzeitfehler, *siehe*
 - Ausnahmebehandlung
- Lebensdauer, 60, 133
- `length_error`, *siehe*
 - Ausnahmebehandlung
- Lesbarkeit, 63
- Lexikalische Elemente von C++, 17–32
- Linken, 10
- `List`, Übungsbeispiel, 223, 245
- Literale, 21
 - Fließkommazahlen, 23
 - Ganze Zahlen, 21
 - Wahrheitswerte, 25
 - Zeichen, 23
 - Zeichenkette, 25
- Local Scope*, *siehe* Gültigkeitsbereich
- `logic_error`, *siehe*
 - Ausnahmebehandlung
- Logischer Fehler, *siehe*
 - Ausnahmebehandlung
- Logischer Negationsoperator `!`, 48
- Lokale Objekte, 122
- Lokaler Gültigkeitsbereich, *siehe* Gültigkeitsbereich
- `long`, Datentyp, 36
- `long double`, Datentyp, 37
- LValue-RValue*-Typumwandlungen, 138
- LValue (Left Value)*, 44
- Makro, 81
 - `#elif`, 340
 - `#else`, 340
 - `#endif`, 340
 - `#error`, 338
 - `#ifdef`, 340
 - `#ifndef`, 340
 - `#if`, 340
 - `#include`, 338
 - `#line`, 339

- #pragma, 339
- assert, 339
- Managerfunktion, 169
- Mantisse, 37
- MapSite, Klasse, 302
- Maskieren von Bits, 48
- Maze, Übungsbeispiel, 300
- Mehrdeutigkeit von Methoden, 288
- Mehrfach-*Includes*, 150, 167
- Mehrfache Basisklassen, 289
- Mehrfachvererbung, *siehe* Vererbung
- Memberwise Assignment*, 118
- Methode, *siehe* Elementfunktion
- Mixed Case*, 20
- Mixin*-Klassen, 291
- Modifikatoren, 169
- Modul *Stack*, Implementierung, 152
- Modul *Stack*, Schnittstelle, 151
- Module, 147–157
 - Abhängigkeit, 155
 - Datenkapsel, 149
 - Geheimnisprinzip, 149
 - Implementierung, 152
 - in C++, 149
 - Schnittstelle, 148, 150
 - Testbarkeit, 156
- Multiplikative Operatoren: *, / und %, 49
- mutable, 133, 173

- Namenlose Namensräume, *siehe* Namensräume
- Namensauflösung, 84, 126
- Namensgebung, 20
- Namenskonflikte, 183
- Namensräume, 124–128
 - Alias, 125
 - Deklaration, 124
 - Gültigkeitsbereich, 122
 - Globaler Gültigkeitsbereich, 127
 - Namenlose, 127
 - Offen, 125
 - Using*-Deklaration, 125
 - Using*-Direktive, 125
- Namespace*, *siehe* Namensräume
- Namespace Alias (Namensräume), 125
- Namespace Pollution*, 185
- Nebeneffekte, bei der Auswertung von Ausdrücken, 43
- new, Operator, 49, 95, 96, 202, 206

- Non-Terminalsymbole einer Grammatik, 18
- NULL, 95
- Null, Zeichenketten (Ende), 109
- Nullanweisung, 58
- Nullzeiger, *siehe* NULL

- Objekt
 - Begriff, 10
 - Identität, 289
 - Initialisierung, 188
 - Variable eines Klassentyps, 160
- Oktalzahl (Literal), 22
- One Definition Rule, 34
- online*-Beispiele, 6
- Operanden, 41
- Operator, 26
 - (), 202
 - Überladen, 197
 - Additiver Operator +, 50
 - Additiver Operator −, 50
 - Adressoperator &, 47, 93, 104, 209
 - als Elementfunktion, 199
 - als Funktionen, 199
 - Alternative Symbole, 30
 - Ausgabeoperator <<, 206, 207
 - Bedingungsoperator: ?:, 53
 - Bit-Komplement-Operator ~, 48
 - Bitweise Operatoren: &, ^ und |, 52
 - Bitweiser Schiebeoperator <<, 50
 - Bitweiser Schiebeoperator >>, 50
 - Cast-Operator (), 49, 95, 142, 292
 - delete, 49, 97, 202
 - Divisionsoperator /, 49
 - Eingabeoperator >>, 206, 208
 - Elementoperator −>, 46, 118
 - Elementoperator ., 46, 118
 - Elementzeigeroperator −>, 49, 214
 - Elementzeigeroperator .* , 49, 214
 - Funktionsaufrufoperator (), 45, 76
 - Funktionsnotation, 198
 - Gleichheitsoperatoren: ==, !=, 51

- Indexoperator [], 45, 103, 201
- Indirektionsoperator *, 47, 94, 107
- Kommaoperator ,, 54
- logische Operatoren && und ||, 52
- Logischer Negationsoperator !, 48
- Modulo-Operator %, 49
- Multiplikationsoperator *, 49
- new, 49, 95, 202, 206
 - Placement, 96
- Operatorfunktion, *siehe* Funktion
- Postfix-Dekrement-Operator --, 46
- Postfix-Inkrement-Operator ++, 46
- Präfix-Dekrement-Operator --, 47
- Präfix-Inkrement-Operator ++, 47
- Relationaler Operator <=, 51
- Relationaler Operator <, 51
- Relationaler Operator >=, 51
- Relationaler Operator >, 51
- Scope-Operator ::, 45
- sizeof, 48
- Typumwandlung, 200
- Typumwandlungsoperator (), 46
- Unärer Minus-Operator -, 47
- Unärer Plus-Operator +, 47
- Zusammengesetzte
 - Zuweisungsoperatoren, 54
- Zuweisungsoperator =, 54, 203, 209, 210
- Operatoren, 41
- out_of_range, *siehe*
 - Ausnahmebehandlung
- overflow_error, *siehe*
 - Ausnahmebehandlung
- Parameter, *siehe* Funktionen
- Parameterübergabe, *siehe* Funktionen
 - Call by Reference, 112
 - Call by Value, 112
 - Gegenüberstellung Call by Value, Call by Reference, 113
- Placement, *siehe* new
- Pointer, *siehe* Zeiger
- Polymorphismus, 275–308
 - Abstrakte Basisklasse, 285
 - Protokoll, 285
 - Rein virtuelle Methode, 285
 - Covarianter Rückgabewert, 279
 - Dynamische Bindung, 276
 - Early Binding, 275
 - Mehrfachvererbung, *siehe* Vererbung
 - Polymorphe Klasse, 278
 - Repräsentation polymorpher Objekte, 283
 - Statische Bindung, 275
 - Virtual Function Table, 283
 - Virtuelle Elementfunktion, *siehe* Elementfunktion
- Postfix-Ausdrücke, 45
- Postfix-Dekrement-Operator --, 46
- Postfix-Inkrement-Operator ++, 46
- Präfix-Inkrement-Operator ++, 47
- Primäre Ausdrücke, 44
- Priorität, 42
- private
 - private-Basisklasse, 269
- private, 160, 163, 178
- Programm
 - Stack-Testprogramm, 166
 - Xcpt-Programm, 311
 - Acht-Damen-Problem, 8
 - Datei globs.h, 227
 - Dateipositionen, 332
 - Gültigkeitsbereich & Sichtbarkeit, 218
 - Hello world, 10
 - Klasse Calculator, Schnittstelle, 222
 - Klasse ComicCharacter, 254, 286
 - Klasse Duck, 269
 - Klasse MapSite, 302
 - Klasse Rational, Schnittstelle, 225
 - Klasse SimObject, Schnittstelle, 307
 - Klasse Simulation, Schnittstelle, 306
 - Klasse Stack, mit const-Elementfunktionen, 171

- Klasse `Stack`, Schnittstelle, 164
- Klasse `SuperHero`, 255, 268
- Klasse `Vehicle`, 170
- Klasse `Vehicle` mit `const-Elementfunktionen`, 172
- Klassen `SuperHero`, `Superman` und `Batman`, 264
- Polymorphe `Comicfigurenliste`, 277
- Quadratische Gleichung, 11
- Stack, Ausgabeoperator, 207
- Stack, Eingabeoperator, 208
- Stack-Klasse, Implementierung, 165, 177
- Stack-Klasse, Schnittstelle, 176
- Stack-Modul, Implementierung, 152
- Stack-Modul, Schnittstelle, 151
- Stack-Template, Implementierung, 236
- Stack-Template, Schnittstelle, 235
- Statische Klassenelemente, 182
- Testprogramm `Stack`, 153
- Testprogramm `Stack-Template`, 237
- Turm, 14
- Programme, einfache, 9–16
- Programming by Contract*, 282
- Promotion
 - Gleitkomma, 139
 - Integral, 139
- `protected`, 163, 178, 257
- `protected-Basisklasse`, 269
- Protokoll, 285
- Prototyp, *siehe* Funktionen
- Prototyp (Funktion), 74
- Prozedur, *siehe* Funktion
- `public`
 - `public-Basisklasse`, 269
- `public`, 117, 163, 178
- Pure Virtual*, 285
- Quadratische Gleichung, Programm, 11
- Qualifikations-Typumwandlungen, 139
- Rückgabewerte
 - Referenzen, 115
 - Zeiger, 115
- `range_error`, *siehe* Ausnahmebehandlung
- `Rational`
 - Übungsbeispiel, 224
 - Schnittstelle, 225
- `Rational II`, Übungsbeispiel, 322
- Referenzen, 89, 176
 - Parameterübergabe, *siehe* Funktionen
 - Rückgabewerte, 115
 - Schreibweise, 89
- Regeln einer Grammatik, 17
- `register`, 130
- `reinterpret_cast`, Typumwandlungsoperator, 46, 145
- Rekursion, *siehe* Funktionen
- Relationale Operatoren: `<`, `>`, `<=` und `>=`, 51
- `return`-Anweisung, 70, 74
- Robustheit, *siehe* Iteratoren
- RTTI*, *siehe* Laufzeit-Typinformation
- Run-Time Type Information*, *siehe* Laufzeit-Typinformation
- RValue (Right Value)*, 44
- Sätze einer Grammatik, 17
- Söhne, *siehe* Vererbung
- Schablone, *siehe* Templates
- Schaltvariable, 64
- Schlüsselwörter, 20
- Schleifen, 65
- Schleifenvariable, 68
- Schutz, 162
- Scope*, 60
- `Scope-Operator ::`, 123, 125
- Selektion, 57, 61–65
 - `if`, 61
 - `switch`, 64
- Selektoren, 169
- Sequenz, 57
- `set_unexpected`, *siehe* Ausnahmebehandlung
- Setzen von Bits, 52
- Shallow Copy*, 193, 204

- short, Datentyp, 36
- Short Circuit*, 52
- Sichere Typumwandlung, 138
- Sichtbarkeit, 121
 - von Konstruktoren, 190
- Signatur, 76, 83, 173, 279
- signed char, Datentyp, 36
- SimObject
 - Schnittstelle, 307
- Simulation
 - Schnittstelle, 306
- Simulations-Framework, 303
- Single Source-Prinzip*, 85, 230
- size_t, 49
- sizeof, Operator, 48
- Slicing Problem*, 114, 260, 261
- Spaghettiprogrammierung, 71
- Speicherklasse, 133
 - extern, 132
 - mutable, 133
 - static, 130
- Speicherklassenattribute, 129–133
 - auto, 130
 - extern, 132
 - mutable, 133
- Speichermanagement, 95
- Spezifikatoren, alternative, 35
- Sprachbeschreibung, *siehe*
 - Grammatik
- Sprunganweisungen, 57, 69–71
- Sprungmarken, 59
- Stack
 - Ausgabeoperator, 207
 - Eingabeoperator, 208
 - Implementierung, 165
 - Klasse, Implementierung, 177
 - Klasse, mit const-Elementfunktionen, 171
 - Klasse, Schnittstelle, 176
 - Schnittstelle, 164
 - Template, Implementierung, 236
 - Template, Schnittstelle, 235
 - Template, Testprogramm, 237
 - Testprogramm, 166
- Stack Unwinding*, 312
- Standard Template Library*, 2, 247
- Standard-Typumwandlung, *siehe*
 - Typumwandlung
- static, 130
- static, 130, 133
 - const Klasselemente, 181
 - static-Elementfunktionen, 181
 - Klasselemente, 180
- static_cast,
 - Typumwandlungsoperator, 46, 145
- Statische Bindung, 230, 275
- Statische Klasselemente,
 - Programm, 182
- Statische Methodenauflösung, 280
- Streams*, 323
- Strukturierte Datentypen, 117–119
 - Klassen, *siehe* Klassen
- Strukturierte Programmierung, 57
- Strukturierung, 148
- Subklasse, *siehe* Vererbung
- Substatement*, *siehe*
 - Gültigkeitsbereich
- SuperHero, Klasse, 255, 264, 268
- Superklasse, *siehe* Vererbung
- Superman, Klasse, 264
- switch-Anweisung, 64
- Symbole (einer Grammatik), 17
- Template, 229–249
 - Funktion, 230
 - Überladen von
 - Funktions-Templates, 233
 - Ausprägung, 232
 - Ausprägungszeitpunkt, 233
 - Explizite Qualifizierung, 232
 - Funktions-Template versus
 - Template-Funktion, 232
 - Parameter, 232
 - Klasse, 234
 - Ausprägung, 237
 - Compiler-Probleme, 241
 - Definition, 234
 - Element-Templates, 243
 - Explizite Ausprägung, 239
 - friend, 238
 - Gebundenes Template, 238
 - Geschachtelte Klassen, 238
 - Klassen-Template versus
 - Template-Klasse, 234
 - Notation, 235
 - Spezialisierung, 240
 - Template-Typen, 240
 - Ungebundenes Template, 238

- Übersetzungseinheiten, 241
- Single Source*-Prinzip, 230
- Temporäre Objekte, 77
- Terminalsymbole einer Grammatik, 18
- Terminator, 25
 - Stringendezeichen, 110
- Testen, 163
- Testprogramm Stack, 153
- `this`, 45, 175, 204
- `throw()`, *siehe*
 - Ausnahmebehandlung
- Tree, Übungsbeispiel, 248
- Trigraph-Sequenzen, 30
- `true`, 25
- `try`, 311
- TString II, Übungsbeispiel, 322
- Turm, Programm, 14
- Typ einer Funktion, 76
- `typedef`, 136
- `typeid`, 46, 294
- Typen, 134
- Typidentifikation, 142
- Typidentifikationsoperator
 - `typeid`, 46
- Typinformation, 296
- Typqualifikator, 133
 - `const`, 133
 - `volatile`, 133
- Typumwandlung, 138–145
 - Benutzerdefinierte, 84, 210
 - `explicit`, 211
 - Konstruktor, 211
 - Umwandlungsoperator, 212
 - Explizite, 142
 - `const_cast`, 143
 - `dynamic_cast`, 145
 - `reinterpret_cast`, 145
 - `static_cast`, 145
 - C-Stil, 142
 - Funktionsstil, 142
 - Sichere, 138
 - Standard, 138–141
 - LValue-RValue*, 138
 - `bool`, 141
 - Basisklassen, 141
 - Funktion-Zeiger, 139
 - Gleitkomma, 140
 - Gleitkomma-Integral, 140
 - Gleitkomma-Promotion, 140
 - Integral-Promotion, 84, 139
 - Qualifikation, 139
 - Vektor-Zeiger, 139
 - Zeiger, 141
- Unsichere, 138
- Typumwandlungsoperator
 - `()`, 46, 49, 142
 - `const_cast`, 46, 143
 - `dynamic_cast`, 46, 145
 - `reinterpret_cast`, 46, 145
 - `static_cast`, 46, 145
- Übungsbeispiele
 - Animal, 296
 - `binSearch`, 244
 - Calculator, 220
 - CellWar, 297
 - Counter, 217
 - Date, 217
 - Gültigkeitsbereich, Sichtbarkeit und Lebensdauer, 218
 - Kellerbar, 303
 - List, 223, 245
 - Maze, 300
 - Rational, 224
 - Rational II, 322
 - Tree, 248
 - TString II, 322
- Übungsbeispiele, 6
- Umwandlungsoperatoren, 212
- Unäre Ausdrücke, 47
- Unärer Minus-Operator `-`, 47
- Unärer Plus-Operator `+`, 47
- Underscore*, 19
- `unexpected`, *siehe*
 - Ausnahmebehandlung
- `unexpected-Handler`, *siehe*
 - Ausnahmebehandlung
- Ungebundenes Template, 238
- `union`, 118, 215
- Unsichere Typumwandlung, 138
- `unsigned char`, Datentyp, 36
- Unterklasse, *siehe* Vererbung
- UPN (umgekehrte polnische Notation), 220
- Uses*-Beziehung, *siehe* Vererbung
- Using*-Deklaration, 125
- Using*-Direktive, 125
- Überladen, 189
 - Funktions-Templates, 233
- Überladen von Funktionen, 83
- Überladen von Operatoren, 197
- Übersetzen, 9

- Varianten (union), 215
- Vaterklasse, *siehe* Vererbung
- Vehicle, Klasse, 170, 172
- Vektor, 103
 - als Parameter, 108
 - Indizierung, 103
 - Mehrdimensionale, 103
 - Zeichenkette, *siehe* Zeichenkette
 - Zusammenhang mit Zeigern, 104
- Vektor-Zeiger-Typumwandlung, 139
- Vererbung, 251–273
 - Has-Beziehung, 271
 - Interface-Vererbung, 270
 - Is-A-Beziehung, 253
 - Ableiten, 253
 - Ableiten einer Klasse, 254
 - Abstrakte Basisklasse, *siehe* Polymorphismus
 - Basisklasse, 253
 - Benutzt-Beziehung, 253, 271
 - Call by Value, 261
 - Copy-Konstruktor, *siehe* Konstruktor
 - Destruktor, *siehe* Destruktor
 - Direkte Basisklasse, 253
 - Elementfunktion, 262–269
 - Konstruktoren, *siehe* Konstruktor
 - Konstruktor
 - Überschreiben, 267
 - Zuweisungsoperator, 266
 - Enthält-Beziehung, 253, 271
 - friend, 257
 - Gültigkeitsbereiche, 261
 - Has-Beziehung, 253
 - Hierarchien, 252
 - Implementierungsvererbung, 270
 - Interface-Vererbung, 282
 - Is-A-Beziehung, 251, 258, 271
 - Klassentopologie, 253
 - Late Binding, 276
 - Mehrfachvererbung, 288
 - Mehrdeutigkeit von Methoden, 288
 - Mehrfache Basisklassen, 289
 - Mixin-Klassen, 291
 - Objektidentität, 289
 - Parameterübergabe, 260
 - Programming by Contract, 282
 - protected, 257
 - Söhne, 253
 - Slicing Problem, 260
 - Spezifikation von Basisklassen, 269
 - Subklasse, 253
 - Superklasse, 253
 - Unterklasse, 253
 - Uses-Beziehung, 253, 271
 - Vaterklasse, 253
 - Vererbungs-Beziehung, 253
 - Vererbungshierarchie, 253
 - Virtual Function Table, 283
 - Virtuelle Basisklassen, 291
 - Virtuelle Elementfunktion, 254
 - Wurzel, 253
 - Wurzelklasse, 253
 - Zugriffsmöglichkeiten, 255
 - auf private-Elemente, 255
 - auf public-Elemente, 255
- Vererbungs-Beziehung, 253
- Verkettung von Zeichenketten, 25
- virtual, 136
- Virtual Function Table, 283
- Virtuelle Basisklassen, *siehe* Vererbung
- Virtueller Destruktor, 196
- void
 - Datentyp, 38
 - Zeiger, 100
- volatile, 133
- Vorbelegte Parameter, *siehe* Funktionen
- vtbl, 283
- Wahrheitswerte (Literale), 25
- wchar_t, 24, 36
- while-Anweisung, 66
- Whitespace, 29
- Wide Character Set (wchar_t), 24
- Wiederverwendbarkeit, 230
- Wiederverwendung, 251
- Wurzelklasse, *siehe* Vererbung
- xcpt, Klasse, 311
- Zählschleife, *siehe* for-Anweisung
- Zeichen
 - Konstanten, 40
 - Literale, 23
 - Zeichenkette, 109
 - 0, 109

- Literale, 25, 109
- Zeiger, 91
 - void-Zeiger, 100
 - Addition, 106
 - Adressbildung, 93
 - als Offsets, 213
 - Anlegen von dynamischen
 - Objekten, 95
 - auf Funktionen, 100
 - auf Klassenelemente, 213
 - Dangling Pointer*, 98
 - Deklaration von, 92
 - Garbage Collection*, *siehe*
 - Garbage Collection*
 - Indirektionsoperator *, 94, 107
 - Konstanten, 99
 - Null, 97
 - NULL versus 0, 95
 - Nullzeiger, *siehe* Null
 - Parameterübergabe, 112
 - Rückgabewerte, 115
 - Schreibweise, 92
 - Subtraktion, 106
 - Typumwandlungen, 141
 - Vergleich, 107
 - Zeigerarithmetik, 106
 - Zerstören von
 - Speicherobjekten, 97
 - Zusammenhang mit Vektoren,
 - 104
- Übersetzungseinheiten
 - Klassen-Templates, 241
- Zugriffsfunktion, 169
- Zugriffsschutz, 149, 178
- Zusammengesetzte Anweisung,
 - siehe* Blockanweisung
- Zusammengesetzte
 - Zuweisungsoperatoren, 54
- Zuweisung, 58
- Zuweisungsoperator, 54
 - bei abgeleiteten Klassen, 266
 - in Klassen, 203, 209