

HANSER



Stichwortverzeichnis

zu

„Der C++-Programmierer“ (3. Auflage)

von Ulrich Breymann

ISBN (Buch): 978-3-446-43894-1

ISBN (E-Book): 978-3-446-43953-5

Weitere Informationen und Bestellungen unter
<http://www.hanser-fachbuch.de/978-3-446-43894-1>

sowie im Buchhandel

© Carl Hanser Verlag München

Register

Symbole

* 43, 186, 409
*= 43, 339
+, += 43
++ 43, 340, 405, 409
+= 43
, 75, 211
-, -= 43
->* 229
-- 43, 343
.* 229
... *siehe* Ellipse
/ 43
/* ... */ 31
// 31
/= 43
:: 58, 152, 274
::* 229
|| 54
|= 43
! 43, 54
!= 53, 54, 409
; 33, 65
<, <= 43, 53
< 43
<< 43, 96, 222, 328, 383
<<= 43
=, == 43, 53, 54
>, >= 43, 53
>> 43, 94, 222, 387, 747

>>= 43
?: 66
[] 190, 193, 332
[][]
 Matrixklasse 366
 Zeigerdarstellung 213
132
%, %= 43
& Adress-Operator 54
&, &= Bit-Operatoren 43
&& logisches UND 54
&& Shell 612
&& R-Wert 598
\ 32, 51, 130, 316
\0 194, 197–199
\", \a, \b 51
\f, \n, \r, \t, \v 51, 94
\x 51
\\ 51
~ 43
^ 43
\$, \$^, \$@ 529
@D, @F 617
" 34, 194, 195

A

abgeleitete Klasse 261, 270, 274
 und virtueller Destruktor 287
Abhängigkeit (make) 527
 automatische Ermittlung 612

- abort() 892
- abs() 48, 706, 707, 877, 890, 892
- Abstrakter Datentyp 148, 957
- abstrakte Klasse 280, 957
- accumulate() 660
- acos() 707, 877, 890
- acosh() 707
- Adapter, Iterator- 825
- Additionsoperator 325
- adjacent_difference() 664
- adjacent_find() 691
- adjustfield 385
- Adresse 186
 - symbolische 34
- Adressierung, indirekte (valarray) 885
- Adressoperator 187
- advance() 824
- Aggregation 591, 957
- Aktualparameter 104
- <algorithm> 633, 829
- Algorithmus 28, 405
 - accumulate() 660
 - adjacent_difference() 664
 - adjacent_find() 691
 - all_of 672
 - any_of 673
 - binary_search() 693
 - copy(), -_backward() 728
 - copy_if() 730
 - copy_n() 730
 - count() 671
 - count_if() 671
 - equal() 705
 - equal_range() 694
 - fill() 658
 - fill_n() 658
 - find() 687
 - find_end() 690
 - find_first_of() 687
 - for_each() 728
 - generate(), _n() 659
 - includes() 696
 - inner_product() 662
 - inplace_merge() 686
 - iota() 659
 - is_heap(), -_until() 703
 - is_partitioned() 678
 - is_permutation() 675
 - is_sorted(), -_until 680
 - iter_swap() 731
 - lexicographical_compare() 676
 - lower_bound() 693
 - make_heap() 702
 - max() 737
 - max_element() 665
 - merge() 683
 - mergesort() 684
 - min() 737
 - min_element() 665
 - minmax() 737
 - minmax_element() 665
 - mismatch() 703
 - move(), -_backward() 760
 - next_permutation() 674
 - none_of 672
 - nth_element() 682
 - partial_sort(), -_copy 681
 - partial_sum() 663
 - partition() 677
 - partition_copy(), -_point() 678
 - pop_heap() 701
 - prev_permutation() 674
 - push_heap() 701
 - random_shuffle() 667
 - remove(), -_if(), -_copy(), -_copy_if() 734
 - replace(), -_if(), -_copy(), -_copy_if() 733
 - reverse(), -_copy() 671
 - rotate(), -_copy() 666
 - search() 689
 - search_n() 692
 - set_difference() 698
 - set_intersection() 697
 - set_symmetric_difference() 698
 - set_union() 697
 - shuffle() 668
 - sort() 679
 - sort_heap() 702
 - split() 634
 - stable_partition() 677
 - stable_sort() 680
 - swap() 731
 - swap_ranges() 731
 - transform() 731
 - unique(), -_copy() 668
 - upper_bound() 694
- Alias-Name 54, 187, 189
 - *this 210
- alignof() 362

- allgemeiner Konstruktor 155
 - allocator 756
 - all_of 672
 - alternative Funktions-Syntax 121
 - Anführungszeichen 34, 194, 195
 - Anker (bei regulärem Ausdruck) 417
 - anonymer Namespace 125
 - ANSI-Sequenzen zur Bildschirm-
ansteuerung 51
 - Anweisung 60
 - Anwendungsfall 178
 - any() 817
 - any_of 673
 - app 397
 - append() 858
 - apply() 874
 - Äquivalenzbeziehung 693
 - Äquivalenzklasse 534
 - arg() 706, 707
 - argc 208
 - Argumente *siehe* Parameter
 - argv[] 208
 - Arität (Template) 255
 - Arithmetik mit Iteratoren 408
 - Arithmetik mit Zeigern 191
 - arithmetische Operatoren 43
 - Array
 - char 196
 - von C-Strings 200
 - dynamisches 201, 214, 215, 330
 - Freigabe 205
 - als Funktionsparameter 211
 - mehrdimensionales 210, 214, 215
 - Matrixklasse 367
 - valarray 871
 - vs. Zeiger 191
 - <array> 778, 794
 - Array2d 216, 514
 - ASCII 839
 - Dateien 222
 - Tabelle 51, 897
 - asctime() 895
 - asin() 707, 877, 890
 - asinh() 707
 - assert() 133
 - mit Exception 316
 - assign() 783, 858
 - Assoziation (UML) 588
 - Assoziativität von Operatoren 55, 902
 - at() 83, 87, 785, 789, 795, 799, 857
 - atan() 707, 877, 890
 - atan2() 878, 890
 - atanh() 707
 - ate 397
 - atexit() 892
 - atof(), atol() 638, 892
 - atoi() 638, 653, 892
 - Atom-Uhr 492
 - atomic 436
 - Attribute 150, 958
 - Aufforderung 29, 958
 - Aufzählungstyp 78
 - Ausdruck
 - Auswertung 55, 56
 - Definition 39
 - mathematischer 48, 116
 - Ausgabe 93, 96, 383
 - benutzerdefinierter Typen 383
 - Datei- 96, 381
 - Formatierung 383
 - Weite der 384
 - Ausgabeoperator 328, 383
 - Ausnahme 309, 958
 - behandlung 309
 - Ausrichtung an Speichergrenzen 361
 - auto 90, 414, 566
 - automatische Variable 124
 - make 529
 - Autotools (GNU) 626
- ## B
- back() 784, 787, 789, 791, 795, 857
 - back_inserter(), _insert_iterator 826
 - Backslash
 - Zeichenkonstante \ 51
 - Zeilenfortsetzung 130, 316
 - Backspace 51
 - bad() 396
 - bad_alloc 315
 - badbit 395
 - bad_cast 299, 315
 - bad_function_call 315
 - bad_typeid 302, 315
 - bad_weak_ptr 315
 - base() 825
 - basefield 385
 - basic_-Streamklassen 381
 - basic_string 855
 - Basisklasse 261
 - und virtueller Destruktor 287

- Konstruktor 293
- Subobjekt 296
- virtuelle 295
- Bedingungsausdruck 63
 - make 623
- Bedingungsoperator ?: 66
- beg 398
- begin() C-Array 193
- begin() 410, 780
- Belegungsgrad 807
- benutzerdefinierte
 - Datentypen 78, 80
 - Klassen 328
 - Typen (Ausgabe) 383
 - Typen (Eingabe) 747
- Benutzungszählung 867
- Bereichsnotation 781
- Bereichsoperator :: 58, 152, 274
 - namespace 144
- Bibliothek
 - C 145, 887
 - C++ 755
- Bibliotheksmodul 124
 - dynamisch 623
 - statisch 622
- Bidirectional-Iterator 821
- Big Three/Five (Regel) 571
- Bildschirmsteuerung mit ANSI-
 - Sequenzen 51
- /bin/sh 530
- Binärdatei 223
- binäre Ein-/Ausgabe 221
- binärer Operator 325
 - optimiert 606
- binäres Prädikat 669, 704, 830
- binäre Zahlendarstellung 44
- binary 98, 397
- binary_negate 765
- binary_search() 693
- bind 766
- Binden 124
 - dynamisches *siehe* dynamisches Binden
 - statisches *siehe* statisches Binden
- Bit
 - Operatoren 43, 44
 - Verschiebung 44
 - pro Zahl 40
- Bitfelder 92
- <bitset> 778

- bitset 815
- bitweises ODER, UND, XOR 43
- Blitz (Matrix) 715
- Block 31, 33, 57, 58, 61, 176
 - und dyn. Objekte 204
- bool 53
- boolalpha 53, 384, 390
- Boost
 - create_directory() 742
 - exists() 740
 - Filesystem 739
 - is_any_of() 635
 - is_directory() 740
 - lexical_cast() 637
 - Matrix 715
 - remove_all() 740
 - split() 635
- Boost.Asio 485
- break 67, 75
- bsearch() 892
- Bucket 806
- bucket() 812, 814
- Byte 50
- Byte-Reihenfolge 493

C

- C++-Schlüsselwörter 899
- C-Arrays 189
- C-Header 887
- C-String 194
- call wrapper 437
- Callback-Funktion 226, 768
- calloc() 358
- capacity() 785, 857
- case 67
- <cassert> 133, 888
- cast *siehe* Typumwandlung
- catch 310
- cbegin() 780, 856
- <cctype> 732, 888
- cdecl 232
- ceil() 890
- cend() 780, 856
- cerr 94, 382
- <cerrno> 889
- <cfloat> 887
- char 49, 192
- char* 194
- char* const vs. const* char 189
- chmod 742

- <chrono> 429
- cin 34, 94, 382, 396
- <ciso646> 887
- class 151, 269
- class (bei Template-Parametern) 135
- clear() 396, 783, 800, 804, 859
- Client-Server
 - Beziehung 152
 - und callback 226
- <climits> 888
- <locale> 888
- clock(), clock_t 896
- clog 382
- close() 96
- Closure 353
- CMake 629
- <cmath> 48, 706, 755, 887, **889**
- code bloat 629
- <codecvt> 846
- Code-Formatierer 650
- collate 643, 843, 844
- combine() 838
- compare() 844, 862
- Compilationsmodell 630
- Compiler 32, 35, 123, 149, 151
 - befehle 900
 - direktiven 122, 128
 - und Templates 138
 - Typumwandlung 168
- <complex> 706
- Computerarithmetik 47
- configure 626
- conj() 707
- connect() 462
- const 49
 - Elementfunktionen 151, 174
 - globale Konstante 126
- const_cast<>() 300
- const char* vs. char* const 189
- constexpr 49, 141
- const_iterator 779
- const_local_iterator 812, 815
- const_pointer 784
- constraint, Vererben von 289
- const_reference 779
- const_reverse_iterator 782, 825
- const_pointer 798
- const& *siehe* Referenz auf const
- Container 404
 - implizite Datentypen 779
 - Container-Methoden 780
 - container_type 790, 791
 - continue 75
 - copy elision 596
 - copy semantics 600
 - copy() 728, 857
 - copy_backward() 728
 - copy_if() 730
 - copy_n() 730
 - cos(), cosh() 707, 877, 890
 - count() 800, 811, 814, 817
 - Algorithmus 671
 - set 804
 - count_if() 671
 - cout 33, 94, 382
 - crbegin(), crend() 782, 856
 - create_directory() (Boost.Filesystem) 742
 - critical section 435
 - CRLF 496
 - <csetjmp> 888
 - cshift() 874
 - <csignal> 888
 - <cstdarg> 890
 - <cstddef> 45, 188, **891**
 - <cstdint> 770
 - <cstdio> 740, 891
 - <cstdlib> 116, 225, 755, 889, **891**
 - c_str() 98, 234, 857
 - <cstring> 195, 227, 233, 235, **893**
 - <ctime> 342, **895**
 - ctype 642, 644, 844
 - cur 398
 - curr_symbol 848
 - <cwctype> 889
 - CXXFLAGS 529

D

 - dangling pointer 204
 - data race 435, 441
 - data() 784, 795, 857
 - Datagramm 491
 - __DATE__ 134
 - Datei
 - ASCII 222
 - binär 223
 - Ein-/Ausgabe 96, 381
 - kopieren 97
 - löschen 740
 - Öffnungsarten 397
 - öffnen 98

- Positionierung 398
 - schließen 97
 - umbenennen 741
 - Dateizugriffsrechte 742
 - Daten
 - als Attributwerte 958
 - static-Element- 243
 - Datenbankanbindung 511
 - Datenkapselung 959
 - Datensatz 80
 - Datentypen 34, 39
 - abstrakte *siehe* Abstrakter Datentyp
 - benutzerdefinierte 78
 - int und unsigned 65
 - logische 53
 - parametrisierte 135, 248
 - polymorphe 284
 - strukturierte 80
 - zusammengesetzte 78
 - date_order() 851
 - Datum
 - gültiges 343
 - Klasse 340
 - regulärer Ausdruck 721
 - dec 384, 390
 - decimal_point() 847, 848
 - decltype 259
 - declval 260
 - default (in switch-Anweisung) 67
 - default constructor 154
 - Default-Parameter *siehe* vorgegebene Parameter
 - default_random_engine 668, 725
 - = default 162
 - #define 129, 130
 - Definition 126, 959
 - von static-Elementdaten 243
 - von Objekten 156
 - Deklaration 33, 34, 126, 959
 - einer Funktion 103
 - Funktionszeiger 224
 - Lesen einer D. 230
 - in for-Schleifen 73
 - Deklarationsanweisung 60
 - Dekrementierung 43
 - Dekrementoperator 343
 - Delegation 305
 - delegierender Konstruktor 165
 - delete 201, 203, 246, 285, 287
 - überladen 354
 - delete [] 205
 - = delete 162, 569
 - <deque> 778, 788
 - deque 788
 - Dereferenzierung 186, 225
 - Destruktor 176, 294, 571
 - implizite Deklaration 176
 - und exit() 177
 - virtueller 285
 - detach() 431
 - Dezimalpunkt 45, 847
 - Dialog 473
 - diamond problem 296
 - difference_type 779
 - Differenz (Menge) 698
 - difftime() 896
 - digits, digits10 736
 - distance() 824
 - Distribution 531, 959
 - div(), div_t 892
 - divides 765
 - Division durch 0 318
 - dll 623
 - DNS 483
 - do while 70
 - Dokumentation (mit doxygen) 549
 - domain_error 315
 - double 45, 191
 - nicht als Laufvariable 74
 - korrekter Vergleich 565
 - downcast 299, 373
 - Dubletten entfernen 668
 - Durchschnitt (Menge) 697
 - dynamic_cast<>() 299
 - dynamisches Array 330
 - dynamisches Binden 224, 275, 959
 - dynamische Datenobjekte 201
 - dynamischer Typ 301
- ## E
- e, E 45
 - effizienter binärer Operator 606
 - egrep 415, 647
 - Ein- und Ausgabe 93
 - Eingabe 386
 - Datei- 96, 381
 - von Strings 95
 - benutzerdefinierter Typen 747
 - Einschränkung *siehe* constraint

- Elementdaten, Zeiger auf 230
 - Elementfunktion 148, 264
 - als Funktionsobjekt 769
 - Zeiger auf 229
 - Ellipse
 - C-Funktionen 890
 - in catch-Klausel 312
 - printf() 256
 - template parameter pack 257
 - else 62
 - emplace() 781
 - emplace_back() 785, 787, 789
 - emplace_front() 787, 789
 - empty() 780, 790, 792, 793, 857
 - end 398
 - end() C-Array 193
 - end() 410, 780
 - #endif 130
 - endl 34, 40, 51, 390
 - ends 390
 - ENTER 32, 94
 - »enthält«-Beziehung 295
 - enum 78
 - Environment, env[] 208
 - Eof 388
 - eof() 311, 388, 396
 - eofbit 395
 - epsilon() 736
 - equal() 705
 - equalsIgnoreCase() 644
 - equal_range() 694, 800, 804, 811, 814
 - equal_to 765
 - erase() 783, 799, 803, 811, 814, 859
 - ereignisgesteuerte Programmierung 460
 - Ergebnisrückgabe 104
 - errno 308, 740, 889
 - Exception 958
 - arithmetische Fehler 318
 - und Destruktor 310
 - Handling 309
 - Hierarchie 313
 - Speicherleck durch Exc. 572
 - <exception> 315
 - exception 313
 - Exception-Sicherheit 321
 - exists() (Boost.Filesystem) 740
 - exit() 116, 892
 - und Destruktor 177
 - Exklusiv-Oder (Menge) 698
 - exp() 707, 877, 890
 - explicit 162
 - explizite Instanziierung von
 - Templates 631, 632
 - Exponent 45, 46
 - Expression Template 610
 - extern 124, 126, 128
 - extern template 631
 - external linkage 126
- ## F
- f, F 45
 - fabs() 707, 890
 - Facette 843
 - fail() 396, 397
 - failbit 395, 748
 - fakultaet() 102
 - Fallunterscheidung 66
 - false 53
 - falsename() 847
 - false_type 773
 - Fehlerbehandlung 307
 - Ein- und Ausgabe 394
 - Fibonacci 255, 665
 - __FILE__ 134
 - fill() 384, 658
 - fill_n() 658
 - final 570
 - find() 800, 811, 814
 - Algorithmus 687
 - set 804
 - string 860
 - find_...-Methoden (string) 861
 - findstring 623
 - find_end() 690
 - find_first_of() 687
 - fixed 384, 386, 390
 - Fixture 542
 - flache Kopie 236
 - flags() 385
 - flip() 786, 817
 - float 45
 - floatfield 385
 - floor() 890
 - flush() 386
 - flush (Manipulator) 390
 - fmod() 890
 - fmtflags 384
 - for 72
 - Kurzform 89
 - foreach (make) 618

- for_each() 728
 - Formalparameter 104
 - Formatierung der Ausgabe 383
 - forward() 761
 - Forward-Iterator 821
 - <forward_list> 778
 - forward_list 757
 - frac_digits() 848
 - Fragmentierung (Speicher) 206
 - Framework 461
 - free store 201
 - free() 357
 - frexp() 890
 - friend 242
 - front() 784, 787, 789, 791, 795, 857
 - front_inserter(), -insert_iterator 826
 - <fstream> 96
 - fstream 381, 399
 - Füllzeichen 384
 - function 768
 - <functional> 315, 765
 - Funktion 102
 - mit Gedächtnis (static) 105
 - mit initializer_list (variable Parameterzahl) 781
 - klassenspezifische 243
 - mathematische 48, 889
 - vorgegebene Parameterwerte 113
 - Parameterübergabe
 - per Referenz 111
 - per Wert 107
 - per Zeiger 206
 - rein virtuelle 280
 - mit Definition 281
 - static 243
 - alternative Syntax 121
 - Überschreiben in abgeleiteten Klassen 274
 - Überschreiben virtueller Funktionen 278
 - virtuelle *siehe* virtuelle Funktionen
 - Funktionsobjekte 351, 392
 - function 768
 - mem_fn 769
 - Funktionspezifikation 121
 - Funktions-Template 135
 - Funktork *siehe* Funktionsobjekte
 - <future> 315
- G**
- g++ 36
 - Ganzzahlen 40
 - garbage collection 205
 - gegenseitige Abhängigkeit von Klassen 182
 - Genauigkeit 46
 - Generalisierung 262
 - generate(), generate_n() 659
 - generische Programmierung 404
 - GET (http) 496
 - get() 94, 222, 387, 388
 - getaddrinfo() 484
 - getenv() 892
 - getline() für Strings 95, 863
 - getline(char*, ...) 388
 - getloc() 402
 - get_money() (Manipulator) 391
 - get_monthname() 851
 - getnameinfo() 484
 - get_time() (Manipulator) 391
 - get_time(), get_weekday(), get_year() (locale) 851
 - GGT 69
 - schnell 170
 - Gleichheitsoperator bei Vererbung 374
 - Gleichverteilung 725
 - Gleitkommazahl 49
 - Syntax 45
 - global 58
 - Namensraum 145
 - Variable 125, 128
 - gmtime() 896
 - GNU 960
 - GNU Autotools 626
 - good() 396
 - goodbit 395
 - Grafische Benutzungsschnittstellen 459
 - greater, greater_equal 765
 - greedy (regex-Auswertung) 418
 - Grenzwerte von Zahltypen 736
 - Groß- und Kleinschreibung 32, 37
 - größter gemeinsamer Teiler *siehe* GGT
 - grouping() 847, 848
 - gslice 881
 - gslice_array 883
 - GTK+ 459
 - guard (Threads) 436
 - Gültigkeitsbereich 115
 - Block 57

- Datei 125
- Funktion 104
- Klassen 151
- und new 204
- GUI 459
- H**
- hängender Zeiger *siehe* Zeiger, hängender
- hardware_concurrency() 425
- has_facet() 838
- hash() 844
- Hash-Funktion 806, 808
- hasher 809, 813
- has_infinity 736
- has_sort_function 774
- »hat«-Beziehung 591
- Header 34
 - C++-Standard 145
 - der Standardbibliothek 754
- Header (Http) 496
- Header-Datei 122, 123
 - Inhalt 127
- Heap 699
- hex 384, 390
- Hexadezimalzahl 42
- Host Byte Order 493
- I**
- iconv 842
- IDE 36
- Identität von Objekten 150, 960
- IEC 60559, IEEE 754 46
- if 62
- #ifdef, #ifndef 129
- ifeq 623
- ifstream 96, 381
- ignore() 388
- imag() 706, 707
- imbue() 836
- Implementation 123
- Implementationsdatei, Inhalt 127
- Implementationsvererbung 303
- implizite Deklaration
 - Destruktor 176
 - Konstruktor 154
 - Zuweisungsoperator 335, 372
- in 98, 397
- #include 32, 122, 128
- includes() 696
- Indexoperator 82, 190, 193, 213, 332
- indirekte Adressierung (mit indirect_array) 885
- infinity() 736
- Initialisierung
 - = bzw. {} 57
 - array 795
 - direkte I. der Attribute 155
 - und virtuelle Basisklassen 297
 - C-Array 191, 211
 - Konstante in Objekten 157, 244
 - globaler Konstanten 126, 128
 - mit konstruktor-interner Liste 156, 244
 - mit Liste (Container) 781
 - von Objekten 154
 - mit {}-Liste 156
 - von Referenzen 922
 - Reihenfolge 157, 235
 - in for-Schleife 72
 - von static-Elementdaten 243
 - struct 81
 - und Vererbung 267
 - und Zuweisung 159
- Initialisierung vs. Zuweisung 44
- initializer_list 164, 781
- Inklusions-Modell 630
- Inkrementierung 43
- Inkrementoperator 340
- inline 140, 153
- inner_product() 662
- inplace_merge() 686
- Input-Iterator 820
- insert() 783, 799, 803, 805, 810, 813, 859
- insertter() 827
- insert_iterator 826
- Installation der Software 950
- Instanz 30, 960
- Instanziierung von Templates 251
 - explizite 631, 632
 - ökonomische (bei vielen Dateien) 629
- int 33, 40
- int-Parameter in Templates 251
- int2string() 639
- integral promotion 56
- integral_constant 773
- Interface (UML) 586
- internal 384, 390
- internal linkage 126
- Internet-Anbindung 481

- Interrupt (Thread) 442
 - Intervall (und Notation) 781
 - Introspektion 462
 - invalid_argument 315
 - <iomanip> 390, 391
 - <ios> 390
 - ios 381, 394, 396
 - ios_base 381
 - ios_base::binary, ios_base::in 98
 - ios_base-Fehlerstatusbits 395
 - ios_base-Flags 384, 385
 - ios_base-Manipulatoren 390
 - ios::failure 396
 - ios-Flags zur Dateipositionierung 398
 - ios-Methoden 385, 386, 396
 - iostate 394
 - <iostream> 34, 382, 383, 387, 390
 - iota() 659
 - IP4-Adresse (regulärer Ausdruck) 723
 - IPv4, IPv6 483
 - is() 845
 - isalnum(), isalpha() 843, 889
 - is_any_of() 635
 - isblank() 889
 - is_bounded 736
 - isctrl() 843, 889
 - isdigit() 121, 161, 843, 889
 - is_directory() (Boost.Filesystem) 740
 - is_exact 736
 - isgraph() 843, 889
 - is_heap(), is_heap_until() 703
 - is_iec559, is_integer 736
 - islower() 843, 889
 - is_modulo 736
 - ISO 10646 840
 - ISO-8859-1, ISO-8859-15 839
 - isprint() 843, 889
 - ispunct() 843
 - is_signed 736
 - is_sorted(), _until() 680
 - isspace() 843, 889
 - ist-ein-Beziehung 262, 271, 288
 - istream 381, 386
 - istream-Iterator 827
 - istream::seekg(), tellg() 398
 - istream::ws 390
 - istreamstream 381, 400
 - isupper() 843, 889
 - is_class 773
 - is_floating_point 775
 - is_partitioned() 678
 - is_permutation() 675
 - isxdigit() 843, 889
 - Iterator 405, 409, 819
 - Adapter 825
 - Bidirectional 821
 - Forward 821
 - Input 820
 - Insert 826
 - Output 820
 - Random Access 821
 - Reverse 825
 - Stream 827
 - Zustand 410
 - <iterator> 819
 - iterator 779
 - iterator_category 820
 - Iterator-Tags 821
 - iter_swap() 731
- J**
- Jahr 895
 - join() 428, 431
 - Jota 659
- K**
- Kardinalität 588
 - Kategorie (locale) 843
 - key_comp() 800, 804
 - key_compare 798, 803
 - key_equal 809, 813
 - key_type 798, 803, 809, 813
 - Klammerregeln 55
 - Klasse 29, 149, 960
 - abgeleitete *siehe* abgeleitete Klasse
 - abstrakte 280
 - Basis- *siehe* Basisklasse
 - Deklaration 151
 - konkrete 280
 - Ober- *siehe* Oberklasse
 - für einen Ort 150
 - Unter- *siehe* Unterklasse
 - für rationale Zahlen 166
 - Klassenname 302
 - klassenspezifische
 - Daten 243
 - Funktionen 243
 - Konstante 247

- Klassen-Templete 248
 - Klassifikation 262, 960
 - Kleinschreibung 32
 - Kollisionsbehandlung 806
 - Kommandointerpretierer 527, 530
 - Kommandozeilenparameter 208
 - Kommaoperator 75, 211
 - Kommentar 31
 - komplexe Zahlen 706
 - Komplexität 965
 - Komposition 591
 - konkrete Klasse 280
 - Konsole auf UTF-8 einstellen 839
 - Konstante 49
 - globale 126, 128
 - klassenspezifische 247
 - konstante Objekte 174
 - Konstruktor 151, 154
 - allgemeiner *siehe* allgemeiner Konstruktor
 - implizite Deklaration 154
 - delegierender 165
 - erben 272
 - Kopier- *siehe* Kopierkonstruktor
 - vorgegebene
 - Parameterwerte 156
 - Typumwandlungs- *siehe* Typumwandlungskonstruktor
 - Kontrollabstraktion 405
 - Kontrollstrukturen 60
 - Konvertieren von Datentypen *siehe* Typumwandlung
 - Kopie, flache/tiefe 236
 - Kopieren
 - von Dateien 97
 - von Objekten 236
 - von Zeichenketten 198
 - Kopierkonstruktor 158, 333, 571
 - Auslassung durch Compiler 160
 - Kreuzreferenzliste 655
 - kritischer Bereich 435
 - Kurzform-Operatoren 42
- L**
- l, L 45
 - L-Wert 61, 190, 196, 333, 961
 - Länge eines Vektors 662
 - Lambda-Funktionen 352
 - LANG 836
 - late binding 224
 - Laufvariable 72, 73
 - Laufzeit 186
 - und Funktionszeiger 224
 - und new 201
 - und Polymorphie 275
 - Typinformation 301
 - ldd 624
 - ldexp() 890
 - ldiv(), ldiv_t 892
 - LD_LIBRARY_PATH 625
 - left 384, 390
 - length() 857
 - length_error 315
 - less, less_equal 765
 - lexical_cast() 637
 - lexicographical_compare() 676
 - lexikografischer Vergleich 763, 780, 961
 - <limits> 40, 45, 736
 - __LINE__ 134
 - Linken 125, 145
 - dynamisches 623, 961
 - internes, externes 126
 - statisches 622, 961
 - Linker 36
 - linksassoziativ 55
 - list 786
 - <list> 757, 778
 - Liste
 - Initialisierungs- 156, 244
 - Initialisierungs- (bei C-Arrays) 211
 - Liste (Klasse) 410
 - Literal 49, 195
 - load factor 807
 - load_factor() 812, 815
 - <locale> 835
 - localtime() 342, 896
 - local_iterator 809, 812, 813, 815
 - lock_guard 436
 - log(), log10() 707, 877, 890
 - logical_and, _not, _or 765
 - logic_error 314, 315
 - logischer Datentyp 53
 - logische Fehler 316
 - logische Negation 54
 - logisches UND bzw. ODER 54
 - lokal (Block) 57
 - lokale Objekte 189
 - long 40
 - long double 45

lower_bound() 693, 800, 804

lvalue *siehe* L-Wert

M

main() 31, 33, 116

MAKE 617

make 525

 automatische Ermittlung von

 Abhängigkeiten 612

 parallelisieren 621

 rekursiv 616

 Variable 529

Makefile 124, 527

make_heap() 702

make_pair() 763

make_shared() 868

make_tupel() 764

Makro 130

malloc() 357

Manipulatoren 389

Mantisse 46

<map> 778, 796

mapped_type 798, 809

mask_array 884

match_results 421

mathematischer Ausdruck 48

mathematische Funktionen 889

Matrix

 C-Array 210

 C-Array, dynamisch 214

 Klasse 216

 Klasse, mit zusammenhängendem
 Speicher 708

 Vektor von Vektoren 366

max() 736, 737, 874

max_bucket_count() 811, 814

max_element() 665

max_exponent, max_exponent10 736

max_load_factor() 812, 815

max_size() 780

mehrdimensionales Array 210

mehrdimensionale Matrizen 366

Mehrfachvererbung 264, 291, 294

MeinString (Klasse) 233

member function *siehe* Element-
funktion

memchr() 894

memcmp(), memcpy(), memmove(), 894

<memory> 315, 757

memory leak 204, 572

memset() 895

mem_fn() 769

Mengenoperationen auf sortierten
Strukturen 695

merge() 683, 788

mergesort() 684

messages 844, 853

Meta-Objekt-Compiler 462

Methode 30, 148, 961

 Regeln zur Konstruktion

 von Prototypen 561

MIME 962

min() 736, 737, 874

min_element() 665

min_exponent, min_exponent10 736

MinGW 526

minmax() 737

minmax_element() 665

minus 765

Minute 895

mischen 683

mismatch() 703

mkdir() 742

mktime() 896

modf() 890

modulare Gestaltung 122

Modulo 43

modulus 765

Monat 895

monetary 844, 847

money_get 848

money_punct 848

money_put 850

Monitor 447

move semantics 600

move() R-Wert 604

move(), *backward()* Container-
Bereich 760

moving constructor 602

M_PI 351, 706, 746

multimap 802

multiplies 765

Multiplikationsoperator 339

Multiplizität (UML) 588

multiset 805

mutable 175

N

Nachbedingung 121, 962

Nachkommastellen 45

- precision 746
 - Name 37
 - einer Klasse 302
 - name() 302, 838
 - Namenskonflikte bei Mehrfach-
 - vererbung 294
 - Namenskonventionen 36
 - Namespace
 - anonym 125
 - in Header-Dateien 133
 - Verzeichnisstruktur 615
 - namespace 32, 143
 - namespace std 59
 - narrow() 846
 - nationale Sprachumgebung 836
 - NDEBUG 133, 316
 - negate 765
 - Negation
 - bitweise 43, 45
 - logische 54
 - Negationsoperator (überladen) 397
 - negative_sign() 848
 - neg_format() 848
 - Network Byte Order 493
 - Netzwerkprogrammierung 481
 - neue Zeile 51, 62
 - new 201, 203, 246
 - Fehlerbehandlung 319
 - überladen 354
 - <new> 315, 869
 - new Placement-Form 869
 - new_handler 319
 - next_permutation() 674
 - noboolalpha 390
 - noexcept 313
 - none() 817
 - none_of 672
 - norm() 706, 707
 - Normalverteilung 726
 - noshowbase, -point, -pos 390
 - noskipws 390
 - not1, not2 765
 - not_equal_to 765
 - nothrow 321
 - notify_one(), -_all() 441, 582
 - nounitbuf, nouppercase 390
 - npos 856
 - NRVO *siehe* RVO
 - nth_element() 682
 - NTP – Network Time Protocol 493
 - NULL 188, 891
 - nullptr 188
 - Null-Zeiger und new 321
 - numeric 844, 846
 - numeric_limits 45, 736
 - numerische Auslöschung 47
 - numerische Umwandlung 635, 639, 863
 - num_get, num_put 846
 - NummeriertesObjekt (Klasse) 243
 - numpunct 847
- ## O
- O-Notation 965
 - Oberklasse 261, 274, 962
 - erben von 264
 - Subobjekt einer 267
 - Subtyp einer 270
 - Zugriffsrechte vererben 268
 - Oberklassenkonstruktor 264, 267
 - object slicing 271
 - Objekt 28, 149, 151, 962
 - dynamisches 201
 - als Funktions- 351
 - Identität *siehe* Identität von Objekten
 - Initialisierung 154
 - konstantes 174
 - temporäres 162
 - Übergabe per Wert 159
 - verkettete Objekte 203
 - verwitwetes 204
 - vollständiges 296, 297, 965
 - Objektcode 36
 - Objekthierarchie 295
 - Objektorientierung 147
 - oct 384, 390
 - ODER
 - bitweises 43
 - logisches 54
 - Öffnungsarten für Streams 397
 - offsetof 891
 - ofstream 96, 381
 - Oktalzahl 42
 - omanip 391
 - one definition rule 127
 - open() 96
 - OpenMP 583
 - Open Source 963
 - Operator
 - arithmetischer 43

- binärer 325
 - Bit- 43
 - für char 53
 - als Funktion 324
 - Kurzform 44
 - für logische Datentypen 54
 - Präzedenz 55, 902
 - relationale 43, 53, 759
 - Syntax 324
 - Typumwandlungs- 344
 - überladen (<<) 383
 - überladen (>>) 747
 - unärer 325
 - für ganze Zahlen 42
 - operator !() 397
 - operator delete() 354
 - operator new() 354
 - operator string() 345
 - operator () () 351
 - operator*() 339, 346, 409
 - operator*=() 339
 - operator++() 340, 342, 409
 - operator++(int) 567
 - operator+=() 327
 - operator->() 346
 - operator|() 815
 - operator|=() 816
 - operator!=() 409, 817
 - operator<<() 328, 383, 816, 817
 - operator<<=() 817
 - operator=() 336
 - operator==() 409, 817
 - virtual 374
 - operator>>() 387, 816, 817
 - operator>>=() 817
 - operator[]() 369, 371, 784, 789, 795, 821
 - operator&() 815
 - operator&=() 816
 - operator^=() 816
 - operator^~() 815
 - operator~() 817
 - Optimierung durch Vermeiden
 - temporärer Objekte 160
 - Ort (Klasse) 150
 - ostream 328, 381, 383
 - OStream-Iterator 827
 - ostream::endl, ends, flush() 390
 - ostream::seekp(), tellp() 398
 - ostreamstringstream 381, 400
 - out 397
 - out_of_range 315
 - Output-Iterator 820
 - overflow 42, 47
 - overflow_error 315
 - override 283, 568
- ## P
- pair 762
 - Parameter einer Funktion 103
 - Parameterübergabe
 - per Referenz 111
 - per Wert 107
 - per Zeiger 206
 - parametrisierte Datentypen 135, 248
 - »part-of«-Beziehung 591
 - partial_sort(), _copy 681
 - partial_sum() 663
 - partielle Spezialisierung von
 - Templates 820
 - partition() 677
 - partition_copy(), _point() 678
 - patsubst 531
 - peek() 388
 - perfect forwarding 761
 - Performance 593
 - Permutationen 674
 - PHONY 528
 - π 49, 706
 - Placement new/delete 869
 - plus 765
 - POD (plain old data type) 963
 - pointer 784, 798, 803, 809, 813
 - Pointer, smarte 346
 - polar() 707
 - polymorpher Typ 284, 301
 - Polymorphismus 275, 963
 - pop() 790, 792, 793
 - pop_back() 785, 787, 789
 - pop_front() 787, 789
 - pop_heap() 701
 - portabel (Zeichensatz) 841
 - pos_format() 848
 - Positionierung innerhalb einer Datei 398
 - positive_sign() 848
 - POSIX 836
 - POST (http) 501
 - postcondition *siehe* Nachbedingung
 - Postfix-Operator 340
 - pos_type 398
 - pow() 707, 878

- Prädikat
 - Algorithmus mit P. 830
 - binäres 669, 704, 830
 - unäres 672
 - Präfix-Operator 340
 - Präprozessor 122, 128, 612
 - Präzedenz von Operatoren 55, 902
 - precision() 386
 - precondition *siehe* Vorbedingung
 - prev_permutation() 674
 - PRINT (Makro) 132
 - printf() 256
 - Priority-Queue 792
 - private 151, 268
 - private Vererbung 303
 - Programm
 - ausführbares 36
 - Strukturierung 101
 - Programmierrichtlinien 572
 - Projekte 124
 - protected 268
 - protected-Vererbung 306
 - Prototyp
 - Funktions- 102
 - einer Methode 150
 - Regeln zur Konstruktion 561
 - ptrdiff_t 891
 - public 151, 264, 268
 - push() 790, 791, 793
 - push_back() 785, 787, 789
 - vector 86
 - push_front() 787, 789
 - push_heap() 701
 - put() 96, 222, 383
 - putback() 388
 - put_money(), _time() (Manipulatoren) 391
- Q**
- qsort() 225, 892
 - Qt 461
 - QThread 477
 - Quantifizierer 418
 - Queue 791
 - <queue> 757, 778, 791
 - quicksort() 135
- R**
- R-Wert 61, 190, 961
 - Referenz 599
 - race condition 435, 456
 - radix 736
 - RAII 402, 436, 542, 573, 963
 - rand() 724, 892
 - Random 724
 - Random-Access-Iterator 821
 - random_shuffle() 667
 - RAND_MAX 892
 - range_error 315
 - <ratio> 770
 - rationale Zahlen (Klasse) 166
 - rbegin() 782, 825, 856
 - rdstate() 396
 - read() 221
 - real() 706, 707
 - realloc() 358
 - Rechengenauigkeit 46, 74
 - rechtsassoziativ 55
 - reelle Zahlen 45
 - ref(), reference_wrapper 438, 772
 - reference
 - bitset 816
 - Container 779
 - vector<bool> 785
 - reference collapsing rules 599
 - reference counting 867
 - Referenz 54, 119
 - auf Basisklasse 278
 - auf const 111, 158
 - auf istream 387, 747
 - auf Oberklasse 270, 277
 - auf ostream 328, 383
 - Parameterübergabe per 111
 - Rückgabe per 333
 - auf R-Wert 599
 - Referenzsemantik 236, 593
 - <regex> 315, 422
 - regex_iterator 421
 - regex_match() 423
 - regex_replace() 423, 648
 - regex_search() 423, 647
 - reguläre Ausdrücke 415
 - Reihenfolge
 - Auswertungs- 47, 56
 - der Initialisierung 157, 235
 - umdrehen 671
 - rein virtuelle Funktion 280
 - mit Definition 281
 - reinterpret_cast<>() 192, 221, 301
 - Rekursion 108

- Template-Metaprogrammierung 253, 257
 - rekursiver Abstieg 116, 117
 - rekursiver Make-Aufruf 616
 - relationale Operatoren 43, 53, 759
 - rel_ops 759
 - remove()
 - Algorithmus 734
 - Datei/Verzeichnis (C) 740
 - Liste 787
 - remove_all() (Boost.Filesystem) 740
 - remove_if() 734, 787
 - rename() Datei/Verzeichnis (C) 741
 - rend() 782, 825, 856
 - replace() 733, 860
 - replace_copy(), _if(), _copy_if() 733
 - reserve() 239, 785, 857
 - reset() 817
 - resetiosflags() 391
 - resize() 785, 787, 789, 857
 - return 34, 160
 - return value optimization (RVO) 595
 - reverse() (List) 787
 - reverse(), _copy() 671
 - Reverse-Iterator 825
 - reverse_iterator 782
 - Reversible Container 782
 - rfind() 861
 - right 384, 390
 - rotate(), _copy() 666
 - round_error(), _style() 736
 - RTTI 301
 - rule of three/five 571
 - runtime_error 315
 - rvalue *siehe* R-Wert
 - RVO 595
- S**
- scan_is(), scan_not() 845
 - Schleifen 68
 - in Containern 89
 - do while 70
 - for 72
 - und Strings 196
 - Tabellensuche 85
 - terminierung 70
 - while 68
 - Schlüsselwörter 899
 - Schnittmenge 697
 - Schnittstelle 123, 964
 - einer Funktion 106
 - Regeln zur Konstruktion 561
 - scientific 384, 386, 390
 - scope 58
 - scoped locking 436
 - search() 689
 - search_n() 692
 - sed 620
 - seekg(), seekp() 398
 - Seiteneffekt 63, 103, 197, 199
 - im Makro 134
 - Seitenvorschub 51
 - Sekunde 895
 - Selektion 62
 - sentinel 85, 192
 - sentry 402
 - Sequenz 61
 - Sequenz-Methoden 783
 - Sequenzkonstruktor 163, 332
 - 2-dim. Matrixklasse 711
 - Server-Client
 - Beziehung 152
 - und callback 226
 - <set> 778, 802
 - set() 817
 - setbase() 391
 - set_difference() 698
 - setf() 385
 - setfill() 391
 - set_intersection() 697
 - setiosflags() 391
 - setjmp() 888
 - setprecision() 391
 - setstate() 396
 - set_symmetric_difference() 698
 - set_terminate() 316
 - set_union() 697
 - set_new_handler() 319
 - setw() 391
 - shared_ptr 350, 867
 - für Arrays 573
 - SHELL 530
 - shift() 874
 - short 40
 - showbase, showpoint, showpos 384, 390
 - showContainer() 658
 - shrink_to_fit() 239, 785, 789, 858
 - shuffle() 668
 - Sichtbarkeit 57
 - dateiübergreifend 124

- Sichtbarkeitsbereich (namespace) 143
- Signal 462, 465
- Signalton 51
- Signatur 114, 264, 274–276, 278
- signed char 50
- sin(), sinh() 707, 877, 890
- single entry/ single exit 76
- Singleton 717
- size() 82, 87, 780
- sizeof 193
 - nicht bei dynamischen Arrays 215
- size_t 45, 891
- size_type 779
- Skalarprodukt 662
- skipws 384, 390
- sleep() 430
- sleep_for(), sleep_until() 429
- slice 878
- slice_array 880
- Slot 462, 465
- Smart Pointer 346
 - und Exceptions 572
- Socket 486
- Sommerzeit 895
- Sonderzeichen 51
- Sortieren
 - mit qsort() 225
 - mit sort() 679
 - stabiles 680
 - durch Verschmelzen 684
- sort_heap() 702
- Speicherklasse 124
- Speicherleck 204, 572
- Speicherplatzfreigabe 188
- Speicherverwaltung (eigene) 361
- Spezialisierung
 - von Klassen 262
 - von Templates 137
- Spezifikation einer Funktion 121
- splICE() 788
- split() 634
- Sprachumgebung 836
- SQL 511
- sqrT() 48, 707, 877, 890
- srand() 724, 892
- SSH 555
- <sstream> 400
- stable_partition() 677
- stable_sort() 680
- Stack 57
 - Klasse 248
- <stack> 757, 778, 790
- stack unwinding 310
- Standard Ein-/Ausgabe 94
- Standardbibliothek
 - C 145, 887
 - C++ 755
- Standardheader 146, 757
- Standardklassen 757
- Standard-Typumwandlung 56
 - Zeiger 228
- start() 878, 881
- static 124, 125
 - Attribute und Methoden 243
 - in Funktion 105
 - Makefile 622
- static_assert 134
- static_cast<>() 52, 299
- statisches Binden 275, 964
- Statusabfrage einer Datei 396
- std 32, 59, 144
- <stdexcept> 315
- stdio 384
- Stelligkeit (Template) 255
- STL 403
- stod() 636
- stoi() 635, 636
- stoi() und verwandte numerische Konversionsfunktionen 863
- strcat(), strchr(), strcmp() 893
- strncat(), strncmp() 894
- strpbrk(), strrchr(), strstr() 894
- strcpy() 200, 893
- strcspn() 893
- stream 381
- Stream-Iterator 827
- Stream-Öffnungsarten 397
- strerror() 740, 889, 893
- Streuspeicherung 806
- strftime() 896
- stride() 878, 881
- String 86
- string 86, 855
 - append() 858
 - assign() 858
 - at() 87, 857
 - back() 857
 - begin() 856
 - capacity() 857

- cbegin(), cend(), crbegin(), crend() 856
 - clear() 859
 - compare() 862
 - copy() 857
 - c_str() 857
 - data() 857
 - empty() 857
 - end() 856
 - erase() 859
 - find() 860
 - find_...-Methoden 861
 - front() 857
 - insert() 859
 - length() 87, 857
 - max_size() 857
 - operator+=() 858
 - rbegin(), rend() 856
 - replace() 860
 - reserve() 857
 - resize() 857
 - rfind() 861
 - shrink_to_fit() 858
 - size() 87, 857
 - substr() 861
 - swap() 857
 - verketteten 87
 - <string> 757, 855
 - String in Zahl umwandeln 635
 - String-Klasse MeinString 233
 - Stringlänge 197, 198
 - Stringliteral 840
 - stringstream 636
 - strlen() 195, 198, 893
 - strncpy() 894
 - strtod(), strtol(), strtoul() 637, 892
 - strtok() 653, 894
 - struct 80, 269
 - Strukturanalyse 553
 - Stunde 895
 - Subobjekt 264, 270, 293
 - in virtuellen Basisklassen 295
 - verschiedene 294
 - Substitutionsprinzip 288
 - substr() 861
 - Subtyp 265, 270, 288, 964
 - Subversion 555
 - Suffix 45
 - sum() 874
 - swap() 336, 780, 857
 - Algorithmus 731
 - swap-Trick 219, 578
 - swap_ranges() 731
 - switch 66
 - symmetrische Differenz
 - sortierter Strukturen 698
 - Synchronisation 434
 - Syntaxdiagramm 117, 119
 - ? : Bedingungsoperator 66
 - do while-Schleife 70
 - enum-Deklaration 78
 - for-Schleife 72
 - Funktionsaufruf 104
 - Funktionsdefinition 103
 - Funktionsprototyp 103
 - Funktions-Template 135
 - if-Anweisung 62
 - mathematischer Ausdruck 117
 - operator-Deklaration 324
 - struct-Definition 80
 - switch-Anweisung 66
 - Typumwandlungsoperator 344
 - while-Schleife 68
 - system() 892
 - system_error 315, 396
 - Szenario 178
- ## T
- Tabellensuche 85
 - Tabulator 51
 - Tag 895
 - tan(), tanh() 707, 877, 890
 - target (make) 527
 - Taschenrechnersimulation 116
 - Tastaturabfrage 94
 - TCP 482
 - »Teil-Ganzes«-Beziehung 591
 - tellg(), tellp() 398
 - Template
 - für Funktionen 135
 - #include 138
 - Instanziierung von T. 251
 - explizite 631, 632
 - ökonomische (bei vielen Dateien) 629
 - int-Parameter 251
 - für Klassen 248
 - Spezialisierung 137
 - partielle 820
 - static Attribut 363

- variable Parameterzahl 255, 712
 - Template-Metaprogrammierung 253
 - temporäres Objekt 162
 - Vermeidung 160
 - terminate() 315
 - terminate_handler 316
 - Test Driven Development 535
 - test() 817
 - Test-Suite 537
 - Textersetzung 130
 - this 210
 - this->, *this bei Zugriff auf Oberklassenelement 339
 - this_thread 428
 - thousands_sep() 847, 848
 - Thread 425
 - thread (API) 430
 - Thread-Sicherheit 456
 - ThreadGroup 433
 - thread_group 432
 - throw 310
 - tie() 402
 - tiefe Kopie 236
 - __TIME__ 134
 - time 844, 850
 - time() 342, 896
 - time_get 850
 - time_put 852
 - time_t 342, 895
 - tm 342, 895
 - tolower() 642, 843, 845, 888
 - top() 790, 793
 - to_string() (bitset) 817
 - to_string() (Zahlkonvertierung) 863
 - to_ulong() 817
 - toupper() 642, 841, 843, 845, 888
 - trailing return type 121
 - traits 819
 - transform() 731, 844
 - tree (Programm) 614, 744
 - Trennung von Schnittstellen und Implementation 124
 - true 53
 - true_name() 847
 - true_type 773
 - trunc 397
 - try 310
 - Tupel, <tuple> 713, 764
 - tuple_cat() 714
 - Typ 964
 - polymorpher bzw. dynamischer Typ 301
 - type cast *siehe* Typumwandlung
 - <type_traits> 773
 - typedef 231
 - typeid() 301, 377
 - type_info 301
 - <typeinfo> 315
 - typename (bei Template-Parametern) 135
 - type_traits 773
 - Typinformation 285
 - zur Laufzeit 301
 - Typumwandlung
 - cast 52, 188, 225, 298
 - cast-Schreibweise 52
 - durch Compiler 168
 - const_cast<>() 300
 - dynamic_cast<>() 299
 - mit explicit 162
 - implizit 66, 162
 - mit Informationsverlust 115
 - reinterpret_cast<>() 301
 - Standard- 56
 - Zeiger 228
 - static_cast<>() 299
 - Typumwandlungskonstruktor 161, 168
 - Typumwandlungsoperator 344
 - ios 396
- ## U
- UDP 482, 491
 - Überladen
 - von Funktionen 114
 - von Operatoren *siehe* operator
 - Überlauf 42, 47
 - Überschreiben
 - von Funktionen in abgeleiteten Klassen 274
 - virtueller Funktionen 278
 - Übersetzung 122, 124
 - Übersetzungseinheit 126
 - Umgebungsvariable 208
 - UML 261, 585
 - Umleitung der Ausgabe auf Strings 400
 - unärer Operator 325
 - unäres Prädikat 672
 - unary_negate 765
 - UND
 - bitweises 43, 44
 - logisches 54

#undef 130
 undefined behaviour 205
 underflow 47
 underflow_error 315
 unexpected() 313
 Unicode 381, 839
 union 91
 unique() Sequenzen 787
 unique(), _copy() Algorithmen 668
 unique_lock 436
 unique_ptr 865
 für Arrays 574
 Unit-Test 533
 unitbuf 384, 386, 390
 unordered_map 808
 unordered_multimap 812
 unordered_multiset 815
 unordered_set 812
 unsigned 40
 unsigned char 50
 Unterklasse 261, 964
 upper_bound() 694, 800, 804
 uppercase 384, 390
 URI, URL 482, 648
 URL-Codierung 496
 use case 178
 use_facet() 642–644, 838
 using
 -Deklaration 304
 Namespace
 Deklaration 144
 Direktive 144
 protected Funktion 302
 using (statt typedef) 231
 UTC 896
 UTF-8 839
 <utility> 260, 604, 608, 760, 763

V

Valarray
 arithmetische Operatoren 875
 Bit-Operatoren 876
 logische Operatoren 876
 mathematische Funktionen 877
 relationale Operatoren 876
 <valarray> 871
 value_comp() 800, 804
 value_compare 798, 803
 value_type 779, 798, 803, 809
 Variable 34
 automatische 124
 globale 125, 128
 make 529
 Variablenname 36
 variadic templates 255, 712
 vector **784**
 at() 83
 push_back() 86
 size() 82
 <vector> 757, 778, 784
 vector<bool> 785
 Vektor 82
 Klasse 329
 Länge (geom.) 662
 Verbundanweisung 61
 Vereinigung (Menge) 697
 Vererbung 965
 der abstrakt-Eigenschaft einer
 Klasse 281
 von constraints 289
 der Implementierung 304
 Mehrfach- 291
 private 303
 protected 306
 von Zugriffsrechten 268
 und Zuweisungsoperator 372
 Vergleich von double-Werten 565
 verschmelzen (*merge*) 683
 Versionskontrolle 554
 Vertrag 289, 965
 verwitwetes Objekt 204
 Verzeichnis
 anlegen 742
 anzeigen 743
 löschen 740
 umbenennen 741
 Verzeichnisbaum
 anzeigen 744
 make 615
 Verzweigung 62
 virtual 275, 277, 287
 virtuelle Basisklasse 295
 virtueller Destruktor 285
 virtuelle Funktionen 275, 278
 rein- 280
 void 188
 als Funktionstyp 103
 void* 225
 Typumwandlung nach 188
 Typumwandlungsoperator 397

vollständiges Objekt 296, 297, 965
Vorbedingung 121, 965
vorgegebene Parameterwerte
 in Funktionen 113
 in Konstruktoren 156
Vorkommastellen 45
Vorrangregeln 55, 902
Vorwärtsdeklaration 182

W

Wächter (Tabellenende) 85, 192
Wahrheitswert 53
wait() 440, 448
Warteschlange 791
wchar_t 50, 840, 891
weak_ptr 868
Webserver 502
Weite der Ausgabe 384
Wert
 eines Attributs 958
 Parameterübergabe per 107
Wertebereich 41
Wertsemantik 236, 405, 593
 Performanceproblem 595
what() 314
while 68, 197, 198
whitespace *siehe* Zwischenraum-
 zeichen
wide character 50
widen() 846
Widget 465
width() 384
Wiederverwendung
 durch Delegation 305
 durch Vererbung 271
wildcard 531
Winterzeit 895
Wochentag 895
wofstream 842
Worttrennung 32
Wrapperklasse für Iterator 825
write() 221, 383
ws 390
wstring 842, 855

X

XOR, bitweises 43

Y

yield() 430

Z

Zahl in String umwandeln 639
Zahlenbereich 40, 46
Zeichen 49
Zeichenkette 34, *siehe auch* string
 C-String 194
 Kopieren einer 198
Zeichenklasse 418
Zeichenkonstante 49
Zeichenliteral 840
Zeichensatz 839
Zeiger 185, 201
 vs. Array 191
 auf Basisklasse 278, 285
 auf Elementdaten 230
 auf Elementfunktionen 229
 auf Funktionen 224
 hängender 204
 intelligente *siehe* Smart Pointer
 Null-Zeiger 188
 auf Oberklasse 270, 277
 auf ein Objekt (Mehrfachvererbung) 294
 auf lokale Objekte 189
 Parameterübergabe per 206
Zeigerarithmetik 191
Zeigerdarstellung
 von [] 193
 von [][] 213
Zeile
 einlesen *siehe* getline()
 neue 51
Zeit-Server 494
Zeitkomplexität 965
Ziel (make) 527
Ziffernzeichen 49
Zufallszahlen 724
Zugriffsspezifizierer und -rechte 268
zusammengesetzte Datentypen 78
Zusicherung 133, 316
Zustand 965
 eines Iterators 410
Zuweisung 34, 61, 64, 965
 in abgeleiteten Klassen 270
 und Initialisierung 159
Zuweisungsoperator 159, 335, 571
 implizite Deklaration 335, 372
 und Vererbung 372
zweidimensionale Matrix 708
Zweierkomplement 40
Zwischenraumzeichen 94, 195, 387