

Register

Symbole

* 47, 196, 433
*= 47, 353
+, += 47
++ 47, 354, 429, 433
+= 47
, 60, 80
-, -= 47
->* 241
-- 47, 357
.* 241
... *siehe Ellipse*
/ 47
/* ... */ 35
// 35
/= 47
:: 63, 158, 286
::* 241
; 37, 70
<, <= 47, 57
< 47
<< 47, 101, 234, 342, 401
<<= 47
<random> 695
=, == 47, 57, 58
>, >= 47, 57
>> 47, 99, 234, 405, 422
>>= 47
?: 71
[] 200, 203, 346
[][]
 Matrixklasse 383
 Zeigerdarstellung 224
138
%, %= 47
& Adress-Operator 59
&, &= Bit-Operatoren 47
&& logisches UND 58
&& Shell 582
&& R-Wert 452
\
 36, 55, 136, 330
 \0 204, 207–209
 \\", \\a, \\b 55
 \\f, \\n, \\r, \\t, \\v 55, 99
 \\x 55
 \\ \\ 55
 ~ 47
 ^ 47
 |= 47
 || 58
 ! 47, 58
 != 57, 58, 433
 \$<, \$^, \$@ 579
 @D, @F 587
 " 37, 204, 205

A

- abgeleitete Klasse 273, 282, 286
 - und virtueller Destruktor 299
- Abhängigkeit (make) 577
 - automatische Ermittlung 582
- abort() 928
- abs() 52, 734, 735, 913, 926, 928
- Abstrakter Datentyp 154, 955
- abstrakte Klasse 292, 955
- accumulate() 688
- acos() 734, 913, 926
- acosh() 734
- Adapter, Iterator- 856
- Additionsoperator 339
- adjacent_difference() 691
- adjacent_find() 718
- adjustfield 403
- Adresse 196
 - symbolische 38
- Adressierung, indirekte (valarray) 921
- Adressoperator 197
- advance() 854
- Aggregat 824
- Aggregation 659, 955
- Aktualparameter 110
- <algorithm> 661, 757, 861
- Algorithmus 32, 429
 - accumulate() 688
 - adjacent_difference() 691
 - adjacent_find() 718
 - all_of 700
 - any_of 700
 - binary_search() 720
 - copy(), copy_backward() 754
 - copy_if() 756
 - copy_n() 756
 - count(), count_if() 699
 - equal() 733
 - equal_range() 721
 - fill(), fill_n() 686
 - find() 714
 - find_end() 717
 - find_first_of() 715
 - for_each() 754
 - generate(), _n() 687
 - includes() 723
 - inner_product() 689
 - inplace_merge() 713
 - iota() 687
 - is_heap(), is_heap_until() 730
 - is_partitioned() 705
 - is_permutation() 702
 - is_sorted(), is_sorted_until() 707
 - iter_swap() 757
 - lexicographical_compare() 703
 - lower_bound() 721
 - make_heap() 729
 - make_pair() 785
 - make_tuple() 786
 - max(...) 763
 - max_element() 692
 - merge() 711
 - mergesort() 712
 - min(...) 763
 - min_element() 692
 - minmax() 763
 - minmax_element() 693
 - mismatch() 731
 - move(), move_backward() 783
 - next_permutation() 701
 - none_of 700
 - nth_element() 710
 - partial_sort(), partial_sort_copy 709
 - partial_sum() 690
 - partition() 705
 - partition_copy(), partition_point() 705
 - pop_heap() 728
 - prev_permutation() 701
 - push_heap() 729
 - remove(), _if(), _copy(), _copy_if() 760
 - replace(), _if(), _copy(), _copy_if() 759
 - reverse(), reverse_copy() 698
 - rotate(), rotate_copy() 694
 - search() 716
 - search_n() 719
 - set_difference() 725
 - set_intersection() 725
 - set_symmetric_difference() 726
 - set_union() 724

- shuffle() 695
- sort() 707
- sort_heap() 730
- split() 662
- stable_partition() 705
- stable_sort() 708
- swap(), swap_ranges() 757
- transform() 757
- unique(), unique_copy() 696
- upper_bound() 721
- Alias-Name 59, 197, 199
 - *this 220
- alignment_of 802
- alignof() 375, 802
- all_of 700
- allgemeiner Konstruktor 162
- allocator 778
- alternative Funktions-Syntax 127
- Anführungszeichen 37, 204, 205
- Anker (bei regulärem Ausdruck) 467
- anonymer Namespace 131
- Anweisung 65
- any() 848
- any_of 700
- app 417
- append() 892
- apply() (Tupel) 788
- apply() (valarray) 910
- Äquivalenzbeziehung 721
- Äquivalenzklasse 604
- arg() 734, 735
- argc 218
- Argument *siehe* Parameter
- Argument Dependent Lookup (ADL) 956
- argv[] 218
- Arität (Template) 265
- Arithmetik mit Iteratoren 432
- Arithmetik mit Zeigern 202
- arithmetische Operatoren 47
- Array
 - char 206
 - von C-Strings 210
 - dynamisches 211, 225, 226, 344
 - Freigabe 215
 - als Funktionsparameter 221
 - mehrdimensionales 220, 225, 226
 - Matrixklasse 384
 - valarray 907
 - vs. Zeiger 201
- <array> 806, 823
- Array2d 227, 564
- ASCII 871
 - Dateien 234
 - Tabelle 56, 933
- asctime() 931
- asin() 734, 913, 926
- asinh() 734
- assert() 139
 - mit Exception 330
- assign() 811, 892
- Assoziation (UML) 656
- Assoziativität von Operatoren 60, 938
- at() 88, 92, 813, 818, 824, 828, 891
- atan() 734, 913, 926
- atan2() 914, 926
- atanh() 734
- ate 417
- atexit() 928
- atof(), atol() 666, 928
- atoi() 666, 682, 928
- Atom-Uhr 543
- atomic 487
- Attribute 156, 956
- Aufforderung 33, 956
- Aufzählungstyp 83
- Ausdruck
 - Auswertung 59, 60
 - Definition 43
 - mathematischer 52, 122
- Ausgabe 98, 101, 401
 - benutzerdefinierter Typen 401
 - Datei- 102, 399
 - Formatierung 401
 - Weite der 402
- Ausgabeoperator 342, 401
- Ausnahme 323, 956
 - behandlung 323
- Ausrichtung an Speichergrenzen 374
- auto 95, 438, 637
- automatische Variable 130
 - make 579
- Autotools (GNU) 596

B

- back() 812, 815, 818, 820, 824, 891
- back_inserter(), _insert_iterator 857
- Backslash
 - Zeichenkonstante \ 56
 - Zeilenfortsetzung 136, 330
- Backspace 56
- bad() 415
- bad_alloc 329
- badbit 414
- bad_cast 314, 329
- bad_function_call 329
- bad_typeid 316, 329
- bad_weak_ptr 329
- base() 856
- basefield 403
- basic_string 889
- basic_Streamklassen 400
- Basisklasse 273
 - und virtueller Destruktor 299
 - Konstruktor 307
 - Subobjekt 310
 - virtuelle 309
- Bedingungsausdruck 68
 - make 593
- Bedingungsoperator ?: 71
- beg 418
- begin()
 - Container 434, 808
 - Namespace std 203, 855
 - string 890
- Belegungsgrad 836
- benutzerdefinierte
 - Datentypen 83, 85
 - Klassen 342
 - Literale 380
 - Typen (Ausgabe) 401
 - Typen (Eingabe) 422
- Benutzungszählung 902
- Bereichsnotation 809
- Bereichsoperator :: 63, 158, 286
 - namespace 149
- Bibliothek
 - C 150, 923
 - C++ 776
- Bibliotheksmodul 130
 - dynamisch 593
 - statisch 592
- Bidirectional-Iterator 851
- Big Three/Five (Regel) 231, 641
- /bin/sh 580
- Binärdatei 235
- binäre Ein-/Ausgabe 233
- binärer Operator 339
 - optimiert 460
- binäres Prädikat 697, 732, 862
- binäre Zahlendarstellung 48
- Binärzahl 45
- binary 104, 417
- binary_negate 790
- binary_search() 720
- bind 791
- Binden 129
 - dynamisches *siehe* dynamisches Binden
 - statisches *siehe* statisches Binden
- Bit
 - Operatoren 47, 48
 - Verschiebung 48
 - pro Zahl 43
- Bitfelder 98
- bitset 846
- bitweises ODER, UND, XOR 47
- Block 35, 37, 62, 63, 66, 186
 - und dyn. Objekte 215
- bool 58
- boolalpha 58, 402, 409
- Boost
 - create_directory() 769
 - exists() 766
 - Filesystem 765
 - is_any_of() 663
 - is_directory() 766
 - lexical_cast() 665
 - remove_all() 766
 - split() 663
- Boost.Asio 535
- break 72, 80
- bsearch() 928
- Bstatic (Makefile) 592
- Bucket 836
- bucket() 842, 845

- Byte 54
- Byte-Reihenfolge 543
- C**
- C++-Schlüsselwörter 935
- C-Arrays 199
- C-Funktionen einbinden 152
- C-Header 923
- C-String 204
- call wrapper 488
- Callback-Funktion 238, 792
- calloc() 371
- capacity() 813, 891
- case 72
- <cassert> 139, 924
- cast *siehe* Typumwandlung
- catch 324
- cbegin()
 - Container 808
 - Namespace std 855
 - string 890
- <cctype> 758, 924
- cdecl 244
- ceil() 926
- cend()
 - Container 808
 - Namespace std 855
 - string 890
- cerr 99, 401
- <cerrno> 925
- <cfloat> 923
- char 54, 202
- char* 204
- char16_t, char32_t 872
- char* const vs. const* char 199
- char16_t, char32_t 379
- chmod 768
- <chrono> 476
- cin 37, 99, 401, 415
- <ciso646> 923
- class 157, 281
- class (bei Template-Parametern) 141
- clear() 415, 811, 829, 833, 893
- Client-Server
 - Beziehung 158
 - und callback 238
- <climits> 924
- <locale> 924
- clock(), clock_t 932
- clog 99, 401
- close() 102
- CMake 599
- <cmath> 53, 777, 923, 925
- code bloat 599
- <codecvt> 878
- Code-Formatierer 678
- collate 671, 876
- combine() 870
- compare() 876, 896
- Compilationsmodell 600
- Compiler 35, 38, 128, 155, 157
 - befehle 936
 - direktiven 128
 - und Templates 144
 - Typumwandlung 179
- <complex> 735
- Computerarithmetik 52
- conditional 804
- configure 596
- conj() 734
- connect() 513
- const 54
 - Elementfunktionen 157, 159
 - globale Konstante 132
- const_pointer 827
- const_cast<>() 314
- const char* vs. char* const 199
- constexpr
 - Funktion 146
 - Konstante 54
 - Konstruktor / Methode 173
- const_iterator 807
- const_local_iterator 842, 845
- const_pointer 812
- const_reference 807
- const_reverse_iterator 810, 857
- const& *siehe* Referenz auf const
- constraint, Vererben von 303
- Container 428
 - implizite Datentypen 807
- Container-Methoden 808
- container_type 818, 820

- continue 80
 - copy elision 449
 - copy semantics 453
 - copy() 754, 891
 - copy_backward() 754
 - copy_if() 756
 - copy_n() 756
 - copysign() 926
 - cos(), cosh() 734, 913, 926
 - count() 829, 841, 845, 848
 - Algorithmus 699
 - set 834
 - count_if() 699
 - cout 37, 99, 401
 - __cplusplus 152
 - crbegin()
 - Container 810
 - Namespace std 855
 - string 890
 - create_directory() (Boost.Filesystem) 769
 - crend()
 - Container 810
 - Namespace std 855
 - string 890
 - critical section 486
 - CRLF 546
 - <csetjmp> 924
 - cshift() 910
 - <csignal> 924
 - <cstdarg> 926
 - <cstddef> 48, 198, 927
 - <cstdint> 795
 - <cstdio> 766, 927
 - <cstdlib> 122, 237, 777, 925, 927
 - c_str() 247, 891
 - <cstring> 204, 239, 245, 247, 929
 - <ctime> 356, 931
 - ctype 670, 672, 876, 877
 - cur 418
 - curr_symbol 881
 - <cwctype> 925
 - CXXFLAGS 579
- D**
- dangling pointer 214
 - data race 486, 492
 - data() 812, 824, 891
 - Datagramm 541
 - __DATE__ 139
 - date_order() 883
 - Datei
 - ASCII 234
 - binär 235
 - Ein-/Ausgabe 102, 399
 - kopieren 104
 - löschen 766
 - Öffnungsarten 417
 - öffnen 104
 - Positionierung 417
 - schließen 103
 - umbenennen 767
 - Dateizugriffsrechte 768
 - Daten
 - als Attributwerte 957
 - static-Element- 252
 - Datenbankanbindung 561
 - Datenkapselung 957
 - Datensatz 85
 - Datentypen 38, 43
 - abstrakte *siehe* Abstrakter Datentyp
 - benutzerdefinierte 83
 - int und unsigned 70
 - logische 58
 - parametrisierte 140, 258
 - polymorphe 297
 - strukturierte 85
 - zusammengesetzte 83
 - Datum
 - Klasse / Gültigkeit 354
 - regulärer Ausdruck 746
 - dec 402, 409
 - decay, decay_t 803
 - decimal_point() 879, 880
 - decltype 269
 - decltype(auto) 271, 782
 - declval 272
 - default (in switch-Anweisung) 72
 - default constructor 161
 - Default-Parameter *siehe* vorgegebene Parameter
 - = default 170

- #define 135, 136
 - Definition 132, 957
 - von static-Elementdaten 252
 - von Objekten 162
 - Deklaration 37, 38, 132, 957
 - einer Funktion 109
 - Funktionszeiger 236
 - Lesen einer D. 242
 - in for-Schleifen 78
 - Deklarationsanweisung 65
 - Dekrementierung 46, 47
 - Dekrementoperator 357
 - Delegation 320
 - delegierender Konstruktor 172
 - delete 211, 213, 255, 298, 299
 - überladen 366
 - delete [] 215
 - = delete 170, 639
 - <deque> 806, 817
 - deque 817
 - Dereferenzierung 196, 237
 - Destruktor 186, 308, 641
 - implizite Deklaration 186
 - und exit() 188
 - virtueller 297
 - detach() 482
 - Dezimalpunkt 49, 879
 - Dialog 524
 - diamond problem 310
 - difference_type 807
 - Differenz (Menge) 725
 - difftime() 932
 - digits, digits10 762
 - distance() 854
 - Distribution 581, 957
 - div(), div_t 928
 - divides 790
 - Division durch 0 332
 - dll 593
 - DNS 533
 - do while 75
 - Dokumentation (mit doxygen) 619
 - domain_error 329
 - double 50, 202
 - nicht als Laufvariable 79
 - korrekter Vergleich 635
 - downcast 313, 390
 - Drei (die großen Drei) 231, 641
 - Dubletten entfernen 696
 - Durchschnitt (Menge) 725
 - dynamic_cast<>() 313
 - dynamic_pointer_cast 903
 - dynamisches Array 344
 - dynamisches Binden 236, 287, 957
 - dynamische Datenobjekte 211
 - dynamischer Typ 316
- ## E
- e, E 49
 - effizienter binärer Operator 460
 - egrep 465
 - Ein- und Ausgabe 98
 - Einbinden von C-Funktionen 152
 - Eingabe 404
 - Datei- 102, 399
 - von Strings 100
 - benutzerdefinierter Typen 422
 - Einschränkung *siehe* constraint
 - Elementdaten, Zeiger auf 242
 - Elementfunktion 154, 276
 - als Funktionsobjekt 794
 - Zeiger auf 241
 - Ellipse
 - C-Funktionen 926
 - in catch-Klausel 326
 - printf() 265
 - template parameter pack 266
 - else 67
 - emplace() 810, 819, 821, 822
 - emplace_back() 813, 815, 818
 - emplace_front() 815, 818
 - empty() 808, 819, 821, 822, 891
 - enable_if 803
 - enable_if_t 804
 - end 418
 - end()
 - Container 434, 808
 - Namespace std 203, 855
 - string 890
 - #endif 136
 - endl 44, 55, 409
 - oder '\n'? 101

- ends 409
 - `ENTER` 36, 99
 - »enthält«-Beziehung 309
 - enum 83
 - Environment, env[] 218
 - EOF 406
 - eof() 324, 406, 415, 430
 - eofbit 414
 - epsilon() 762
 - equal() 733
 - equal_to 790
 - equal_range() 721, 830, 834, 841, 845
 - equalsIgnoreCase() 672
 - erase() 811, 829, 833, 841, 844, 893
 - ereignisgesteuerte Programmierung 510
 - Ergebnisrückgabe 110
 - errno 322, 766, 925
 - Exception 956
 - arithmetische Fehler 332
 - und Destruktor 323
 - Handling 323
 - Hierarchie 327
 - Speicherleck durch Exc. 642
 - <exception> 329
 - exception 327
 - Exception-Sicherheit 335
 - exists() (Boost.Filesystem) 766
 - exit() 122, 928
 - und Destruktor 188
 - Exklusiv-Oder (Menge) 726
 - exp() 734, 913, 926
 - explicit 169
 - explizite Instanziierung von
 - Templates 601, 602
 - Exponent 49–51
 - Expression Template 464
 - extern 130, 132, 134
 - extern "C" 152
 - extern template 601
 - external linkage 132
- F**
- f, F 49
 - Fünf (die großen Fünf) 641
 - fabs() 734, 926
 - Facette 876
 - fail() 415, 416
 - failbit 414, 423
 - fakultaet() 108
 - Fallunterscheidung 71
 - false 58
 - false_type 798
 - falsename() 879
 - Fehlerbehandlung 321
 - Ein- und Ausgabe 414
 - Fibonacci 264, 692
 - __FILE__ 139
 - fill() 402, 686
 - fill_n() 686
 - final 640
 - find() 829, 841, 844
 - Algorithmus 714
 - set 834
 - string 894
 - find_end() 717
 - find_first_of() 715
 - find...-Methoden (string) 895
 - findstring 593
 - fixed 402, 404, 409
 - Fixture 612
 - flache Kopie 230
 - flags() 403
 - flip() 814, 848
 - float 50
 - floatfield 403
 - floor() 926
 - flush() 404
 - flush (Manipulator) 409
 - fmod() 926
 - fmtflags 402
 - for 77
 - Kurzform 94
 - foreach (make) 588
 - for_each() 754
 - Formalparameter 110
 - Formatierung der Ausgabe 401
 - forward() 783
 - Forward-Iterator 851
 - <forward_list> 806
 - forward_list 778
 - frac_digits() 881
 - Fragmentierung (Speicher) 216

- Framework 511
 - free store 211
 - free() 370
 - frexp() 926
 - friend 251
 - front() 812, 815, 818, 820, 824, 891
 - front_inserter(), _insert_iterator 857
 - <fstream> 102
 - fstream 400, 418
 - Füllzeichen 402
 - function 793
 - <functional> 329, 789
 - Funktion 108
 - mit Gedächtnis (static) 111
 - mit initializer_list (variable Parameterzahl) 809
 - klassenspezifische 252
 - mathematische 53, 925
 - vorgegebene Parameterwerte 119
 - Parameterübergabe
 - per Referenz 117
 - per Wert 113
 - per Zeiger 216
 - rein virtuelle 292
 - mit Definition 293
 - static 252
 - alternative Syntax 127
 - Überschreiben
 - in abgeleiteten Klassen 286
 - virtueller Funktionen 291
 - virtuelle *siehe* virtuelle Funktionen
 - Funktionsobjekte 365, 410
 - function 792
 - mem_fn 794
 - Funktionspezifikation 127
 - Funktions-Template 140
 - Funktork *siehe* Funktionsobjekte
 - <future> 329
- G**
- g++ 40
 - Ganzzahlen 43
 - garbage collection 216
 - gegenseitige Abhängigkeit von Klassen 193
 - Genauigkeit 51
 - Generalisierung 274
 - generate(), generate_n() 687
 - generische Programmierung 428
 - GET (http) 547
 - get() 99, 234, 405, 406
 - getaddrinfo() 534
 - getenv() 928
 - getline() für Strings 101, 897
 - getline(char*,...) 406
 - getloc() 426
 - get_money() (Manipulator) 408
 - get_monthname() 883
 - getnameinfo() 534
 - get_time() (Manipulator) 408
 - get_time(), get_weekday(), get_year() (locale) 883
 - GGT 74
 - schnell 181
 - Gleichheitsoperator bei Vererbung 391
 - Gleichverteilung 750
 - Gleitkommazahl 53
 - Syntax 49
 - global 63
 - Namensraum 150
 - Variable 130, 134
 - gmtime() 932
 - GNU 958
 - GNU Autotools 596
 - good() 415
 - goodbit 414
 - Grafische Benutzungsschnittstellen 509
 - greater, greater_equal 790
 - greedy (regex-Auswertung) 468
 - Grenzwerte von Zahltypen 762
 - Groß- und Kleinschreibung 36, 40
 - größter gemeinsamer Teiler *siehe* GGT
 - grouping() 879, 880
 - gslice 917
 - gslice_array 919
 - GTK+ 509
 - guard (Threads) 487
 - Gültigkeitsbereich 121
 - Block 62
 - Datei 131
 - Funktion 110

- Klassen 157
 - und new 214
- GUI 509
- H**
- hängender Zeiger *siehe* Zeiger,
 - hängender
- hardware_concurrency() 477
- has_sort_function 798
- has_facet() 870
- hash() 877
- Hash-Funktion 836, 838
- hasher 838, 843
- has_infinity 762
- »hat«-Beziehung 659
- Header 38
 - C++-Standard 150
 - der Standardbibliothek 776
- Header (Http) 546
- Header-Datei 127
 - Inhalt 133
- Heap 727
- hex 402, 409
- Hexadezimalzahl 45
- Host Byte Order 543
- I**
- iconv 875
- IDE 40
- Identität von Objekten 156, 958
- IEC 60559, IEEE 754 50
- if 67
- #ifdef, #ifndef 135
- ifeq 593
- ifstream 102, 400
- ignore() 406
- imag() 734, 735
- imbue() 868
- Implementation 128
- Implementationsdatei, Inhalt 133
- Implementationsvererbung 317
- implizite Deklaration
 - Destruktor 186
 - Konstruktor 161
 - Zuweisungsoperator 349, 389
- in 417
- #include 36, 128, 134
- includes() 723
- Indexoperator 87, 200, 203, 224, 346
- index_sequence 788
- indirekte Adressierung (mit
 - indirect_array) 921
- infinity() 762
- Initialisierung
 - array 824
 - direkte I. der Attribute 162
 - und virtuelle Basisklassen 311
 - C-Array 202, 221
 - mit Element-Initialisierungsliste 163
 - Konstante in Objekten 164, 254
 - globaler Konstanten 132, 134
 - mit {}-Liste 62, 170, 809, 855
 - mit konstruktor-interner Liste 254
 - von Objekten 161
 - mit {}-Liste 162
 - von Referenzen 942
 - Reihenfolge 163
 - in for-Schleife 77
 - von static-Elementdaten 252
 - struct 86
 - und Vererbung 279
 - und Zuweisung 48, 165
- initializer_list 171, 809
- Inklusions-Modell 600
- Inkrementierung 46, 47
- Inkrementoperator 354
- inline 146, 159
- inner_product() 689
- inplace_merge() 713
- Input-Iterator 850
- insert() 811, 829, 833, 835, 840, 844,
 - 893
- insert_iterator 858
- inserter() 858
- Installation der Software 948
- Instanz 34, 958
- Instanziierung von Templates 260
 - explizite 601, 602
 - ökonomische (bei vielen
 - Dateien) 599
- int 37, 43
 - int32_t, int_least32_t usw. 49

- int-Parameter in Templates 261
 - int2string() 667
 - integer_sequence 788
 - integral_constant 798
 - integral promotion 61
 - Interface (UML) 654
 - internal 402, 409
 - internal linkage 132
 - Internet-Anbindung 531
 - Interrupt (Thread) 493
 - Intervall (und Notation) 809
 - Introspektion 512
 - invalid_argument 329
 - <iomanip> 408, 410
 - <ios> 408
 - ios 400, 414, 416
 - ios_base 400
 - ios_base::binary 104
 - ios_base-Fehlerstatusbits 414
 - ios_base-Flags 402, 403
 - ios_base-Manipulatoren 409
 - ios::failure 415
 - ios-Flags zur Dateipositionierung 418
 - ios-Methoden 403, 404, 415
 - iostate 414
 - <iostream> 38, 400, 401, 405, 408
 - iota() 687
 - IP4-Adresse (regulärer Ausdruck) 748
 - IPv4, IPv6 533
 - is() 877
 - isalnum(), isalpha() 875, 925
 - is_any_of() 663
 - is_arithmetic 800
 - is_base_of 802
 - isblank() 925
 - is_bounded 762
 - is_class 797
 - iscntrl() 875, 925
 - isdigit() 126, 168, 875, 925
 - is_directory() (Boost.Filesystem) 766
 - is_exact 762
 - isgraph() 875, 925
 - is_heap(), is_heap_until() 730
 - is_iec559, is_integer 762
 - islower() 875, 925
 - is_modulo 762
 - ISO 10646 872
 - ISO-8859-1, ISO-8859-15 871
 - is_partitioned() 705
 - is_permutation() 702
 - isprint() 875, 925
 - ispunct() 875
 - is_same 802
 - is_signed 762
 - is_sorted(), is_sorted_until() 707
 - isspace() 875, 925
 - ist-ein*-Beziehung 274, 283, 303
 - istream 400, 404
 - IStream-Iterator 858
 - istream::seekg(), tellg() 417
 - istream::ws 409
 - istringstream 400, 419
 - isupper() 875, 925
 - isxdigit() 875, 925
 - iter_swap() 757
 - Iterator 429, 433, 849
 - Adapter 856
 - Bidirectional 851
 - Forward 851
 - Input 850
 - Insert 857
 - Output 850
 - Random Access 851
 - Reverse 856
 - Stream 858
 - Tag 851
 - Zustand 434
 - <iterator> 849
 - iterator 807
 - iterator_category 850
- J**
- Jahr 931
 - join() 479, 482
 - Jota 687
- K**
- Kardinalität 656
 - Kategorie (locale) 876
 - key_comp() 829, 833
 - key_compare 827, 832

- key_equal 838, 843
 - key_type 827, 832, 838, 843
 - Klammerregeln 59
 - Klasse 33, 155, 958
 - abgeleitete *siehe* abgeleitete Klasse
 - abstrakte 292
 - Basis- *siehe* Basisklasse
 - Deklaration 157
 - konkrete 293
 - Ober- *siehe* Oberklasse
 - für einen Ort 156
 - Unter- *siehe* Unterklasse
 - für rationale Zahlen 176
 - Klassenname 316
 - klassenspezifische
 - Daten 252
 - Funktionen 252
 - Konstante 256
 - Klassen-Template 258
 - Klassifikation 274, 959
 - Kleinschreibung 36
 - Kollisionsbehandlung 836
 - Kommandointerpreter 577, 580
 - Kommandozeilenparameter 218
 - Kommaoperator 60, 80, 267
 - Kommentar 35
 - komplexe Zahlen 734
 - Komplexität 964
 - Komposition 659
 - konkrete Klasse 293
 - Konsole auf UTF-8 einstellen 871
 - Konstante 53
 - globale 132, 134
 - klassenspezifische 256
 - konstante Objekte 159
 - Konstruktor 157, 161
 - allgemeiner *siehe* allgemeiner Konstruktor
 - implizite Deklaration 161
 - delegierender 172
 - erben 284
 - Kopier- *siehe* Kopierkonstruktor
 - vorgegebene Parameterwerte 163
 - Typumwandlungs- *siehe* Typumwandlungskonstruktor
 - Kontrollabstraktion 429
 - Kontrollstrukturen 65
 - Konvertieren von Datentypen *siehe* Typumwandlung
 - Kopie, flache/tiefe 230
 - Kopieren
 - von Dateien 104
 - von Objekten 230
 - von Zeichenketten 208
 - Kopierkonstruktor 165, 346, 641
 - Auslassung durch Compiler 167
 - Kreuzreferenzliste 683
 - kritischer Bereich 486
 - Kurzform-Operatoren 46
- L**
- l, L 49
 - L-Wert 66, 200, 206, 347, 959
 - Länge eines Vektors 690
 - Lambda-Funktionen 441
 - LANG 868
 - late binding 236
 - Laufvariable 77, 78
 - Laufzeit 196
 - und Funktionszeiger 236
 - und new 211
 - und Polymorphie 287
 - Typinformation 315
 - LD_LIBRARY_PATH 595
 - ldd 594
 - ldexp() 926
 - ldiv(), ldiv_t 928
 - left 402, 409
 - length() 891
 - length_error 329
 - less, less_equal 790
 - lexical_cast() 665
 - lexicographical_compare() 703
 - lexikografischer Vergleich 785, 809, 959
 - <limits> 44, 50, 762
 - __LINE__ 139
 - Linken 131, 150
 - dynamisches 593, 959
 - internes, externes 132
 - statisches 592, 959
 - Linker 39

- linksassoziativ 60
- list 814
- <list> 779, 806
- Liste
 - Initialisierungs- 163, 254
 - Initialisierungs- (bei C-Arrays) 221
- Liste (Klasse) 434
- Literal 205, 959
 - benutzerdefiniert 380
 - String 379
 - Zahl- 54
 - Zeichen- 54, 872
- load factor 836
- load_factor() 842, 845
- local_iterator 838, 842, 843, 845
- <locale> 867
- localtime() 356, 932
- lock_guard 487
- log(), log10() 734, 913, 926
- logic_error 328, 329
- logical_and, _not, _or 790
- logischer Datentyp 58
- logische Fehler 330
- logische Negation 58
- logisches UND bzw. ODER 58
- lokal (Block) 62
- lokale Objekte 199
- long 43
- long double 50
- lower_bound() 721, 830, 834
- lvalue *siehe* L-Wert
- M**
- magic number 960
- main() 35, 37, 121
- MAKE 587
- make 575
 - automatische Ermittlung von Abhängigkeiten 582
 - parallelisieren 591
 - rekursiv 586
 - Variable 579
- Makefile 129, 577
- make_heap() 729
- make_index_sequence 788
- make_integer_sequence 788
- make_pair() 785
- make_shared 643, 903
- make_tuple() 786
- make_unique 464, 643, 901
- Makro 136
- malloc() 370
- Manipulatoren 407
- Mantisse 50
- <map> 806, 826
- mapped_type 827, 838
- mask_array 920
- match_results 472
- mathematischer Ausdruck 52
- mathematische Funktionen 925
- Matrix
 - C-Array 220
 - C-Array, dynamisch 225
 - Klasse 227
 - Klasse, mit zusammenhängendem Speicher 736
 - Vektor von Vektoren 383
- max() 762, 763, 910
- max(initializer_list<T>) 763
- max_size() 808
- max_bucket_count() 842, 845
- max_element() 692
- max_exponent, max_exponent10 762
- max_load_factor() 842, 845
- mehrdimensionales Array 220
- mehrdimensionale Matrizen 383
- Mehrfachvererbung 276, 305, 308
- MeinString (Klasse) 245
- mem_fn() 794
- member function *siehe* Elementfunktion
- memchr() 930
- memcmp(), memcpy(), memmove(), 930
- <memory> 329, 779
- memory leak 215, 642
- memset() 931
- Mengenoperationen auf sortierten Strukturen 723
- merge() 711, 816
- mergesort() 712
- messages 876, 885

- Meta-Objekt-Compiler 512
- Metaprogrammierung 262
- Methode 33, 34, 154, 960
 - Regeln zur Konstruktion
 - von Prototypen 631
- MIME 960
- min() 762, 763, 910
- min(initializer_list<T>) 763
- min_element() 692
- min_exponent, min_exponent10 762
- minmax(...) 763
- minmax(initializer_list<T>) 763
- minmax_element() 693
- minus 790
- Minute 931
- mischen 711
- mismatch() 731
- mkdir() 768
- mktime() 932
- modf() 926
- modulare Gestaltung 127
- Modulo 47
- modulus 790
- Monat 931
- monetary 876, 880
- money_get 881
- money_punct 880
- money_put 882
- Monitor 498
- move semantics 453
- move() R-Wert 457
- move(), move_backward()
 - Container-Bereich 783
- moving constructor 456
- mt19937 695, 749
- multimap 831
- multiplies 790
- Multiplikationsoperator 353
- Multiplizität (UML) 656
- multiset 835
- mutable
 - Attribut 159
 - Lambda-Funktion 446
- mutex 486
- N**
- '\n' oder endl? 101
- Nachbedingung 127, 960
- Nachkommastellen 49
 - precision 421
- Name 40
 - einer Klasse 316
- name() 316, 870
- Namenskonflikte bei Mehrfach-
 - vererbung 308
- Namenskonventionen 40
- Namespace
 - anonym 131
 - in Header-Dateien 139
 - Verzeichnisstruktur 585
- namespace 37, 149
- namespace std 64
- narrow() 878
- ationale Sprachumgebung 868
- NDEBUG 139, 330
- negate 790
- Negation
 - bitweise 47, 48
 - logische 58
- Negationsoperator (überladen) 416
- negative_sign() 881
- neg_format() 881
- Network Byte Order 543
- Netzwerkprogrammierung 531
- neue Zeile 56, 67
- new 211, 213, 255
 - Fehlerbehandlung 333
 - überladen 366
- <new> 329, 905
- new_handler 333
- new Placement-Form 905
- next_permutation() 701
- noboolalpha 409
- noexcept 326
- none() 848
- none_of 700
- norm() 734, 735
- Normalverteilung 752
- noshowbase, -point, -pos 409
- noskipws 409
- not1, not2 790
- not_equal_to 790
- nothrow 335

- notify_one(), _all() 492, 651
- nounitbuf, nouppercase 409
- npos 890
- NRVO *siehe* RVO
- nth_element() 710
- NTP – Network Time Protocol 543
- NULL 198, 927
- nullptr 198
- Null-Zeiger und new 335
- numeric 876, 879
- numeric_limits 50, 762
- numerische Auslöschung 52
- numerische Umwandlung 663, 667, 897
- num_get, num_put 879
- NummeriertesObjekt (Klasse) 253
- numpunct 879
- O**
- O-Notation 964
- Oberklasse 273, 286, 960
 - erben von 276
 - Subobjekt einer 279
 - Subtyp einer 282
 - Zugriffsrechte vererben 280
- Oberklassenkonstruktor 276, 279
- object slicing 283
- Objekt 32, 155, 157, 961
 - dynamisches 211
 - als Funktions- 364
 - Identität *siehe* Identität von Objekten
 - Initialisierung 161
 - konstantes 159
 - temporäres 169
 - Übergabe per Wert 166
 - verkettete Objekte 213
 - verwitwetes 215
 - vollständiges 310, 311, 964
- Objektcode 39
- Objekthierarchie 309
- Objektorientierung 153
- oct 402, 409
- ODER
 - bitweises 47
 - logisches 58
- Öffnungsarten für Streams 417
- offsetof 927
- ofstream 102, 400
- Oktalzahl 45
- omanip 410
- one definition rule 133
- open() 102
- Open Source 961
- Operator
 - arithmetischer 47
 - binärer 339
 - Bit- 47
 - für char 57
 - als Funktion 338
 - Kurzform 46
 - für logische Datentypen 58
 - Präzedenz 60, 937
 - relationale 47, 58, 781
 - Syntax 338
 - Typumwandlungs- 358
 - überladen (<<) 401
 - überladen (>>) 422
 - unärer 339
 - für ganze Zahlen 46
- operator!() 416
- operator delete() 366
- operator new() 366
- operator string() 358
- operator()() 365
- operator*() 353, 359, 433
- operator*=() 353
- operator++() 354, 356, 433
- operator++(int) 637
- operator+=() 341
- operator->() 359
- operator<<() 342, 401, 846, 847
- operator<<=() 847
- operator=() 350
- operator==() 433, 848
- virtual 391
- operator>>() 405, 846, 847
- operator>>=() 847
- operator[]() 385, 388, 813, 818, 824, 851
- operator&() 846
- operator&=() 847
- operator^() 847

- operator^() 846
 - operator|() 846
 - operator|=() 847
 - operator~() 848
 - operator!==(()) 433, 848
 - Optimierung durch Vermeiden
 - temporärer Objekte 167
 - Ort (Klasse) 156
 - ostream 342, 400, 401
 - OStream-Iterator 858
 - ostream::endl, ends, flush() 409
 - ostream::seekp(), tellp() 417
 - ostreamstream 400, 419
 - out 417
 - out_of_range 329
 - Output-Iterator 850
 - overflow 46, 52
 - overflow_error 329
 - override 295, 638
- P**
- pair 784
 - Parameter einer Funktion 109
 - Parameter Pack 266
 - Parameterübergabe
 - per Referenz 117
 - per Wert 113
 - per Zeiger 216
 - parametrisierte Datentypen 140, 258
 - »part-of«-Beziehung 659
 - partial_sum() 690
 - partial_sort(), partial_sort_copy 709
 - partielle Spezialisierung von
 - Templates 850
 - partition() 705
 - partition_copy(), partition_point() 705
 - patsubst 581
 - peek() 407
 - perfect forwarding 783
 - Performance 447
 - Permutationen 701
 - PHONY 578
 - π 53, 174
 - Placement new/delete 905
 - plus 790
 - POD (plain old data type) 961
 - pointer 812, 827, 832, 838, 843
 - Pointer, smarte 359
 - polar() 734
 - polymorpher Typ 297, 316
 - Polymorphismus 287, 962
 - pop() 819, 821, 822
 - pop_back() 813, 815, 818
 - pop_front() 815, 818
 - pop_heap() 728
 - portabel (Zeichensatz) 873
 - pos_format() 881
 - Positionierung innerhalb einer Datei 418
 - positive_sign() 881
 - POSIX 868
 - POST (http) 551
 - postcondition *siehe* Nachbedingung
 - Postfix-Operator 354
 - pos_type 417
 - pow() 734, 914
 - Prädikat
 - Algorithmus mit P. 862
 - binäres 697, 732, 862
 - unäres 699
 - Präfix-Operator 354
 - Präprozessor 128, 134, 582
 - Präzedenz von Operatoren 60, 937
 - precision() 404
 - precondition *siehe* Vorbedingung
 - prev_permutation() 701
 - PRINT (Makro) 138
 - printf() 265
 - Priority-Queue 821
 - private 157, 280
 - private Vererbung 317
 - Programm
 - ausführbares 39
 - Strukturierung 107
 - Programmierrichtlinien 642
 - Projekte 129
 - protected 280
 - protected-Vererbung 320
 - Prototyp
 - Funktions- 108
 - einer Methode 156
 - Regeln zur Konstruktion 631

- ptrdiff_t 927
 - public 157, 276, 280
 - push() 819, 820, 822
 - push_heap() 729
 - push_back() 813, 815, 818
 - vector 91
 - push_front() 815, 818
 - put() 102, 234, 401
 - putback() 406
 - put_money(), _time() (Manipulatoren) 408
- Q**
- qsort() 237, 928
 - Qt 511
 - QThread 528
 - Quantifizierer 468
 - Queue 820
 - <queue> 779, 806, 820
 - quicksort() 141
 - quoted() 408
- R**
- R-Wert 66, 200, 959
 - Referenz 453
 - race condition 486, 507
 - radix 762
 - RAII 487, 612, 643, 962
 - rand() 753, 928
 - RAND_MAX 928
 - Random-Access-Iterator 851
 - random_device 749
 - range_error 329
 - <ratio> 794
 - rationale Zahlen (Klasse) 176
 - rbegin()
 - Container 810, 856
 - Namespace std 855
 - string 890
 - rdstate() 415
 - read() 233
 - real() 734, 735
 - realloc() 371
 - Rechengenauigkeit 50, 79
 - rechtsassoziativ 60
 - reelle Zahlen 49
 - ref(), reference_wrapper 489, 796
 - reference
 - bitset 846
 - Container 807
 - vector<bool> 813
 - reference collapsing rules 452
 - reference counting 902
 - Referenz 59, 125
 - auf Basisklasse 290
 - auf const 117, 165
 - auf istream 405, 422
 - auf Oberklasse 282, 289
 - auf ostream 342, 401
 - Parameterübergabe per 117
 - Rückgabe per 347
 - auf R-Wert 453
 - Referenzsemantik 230, 447
 - <regex> 329, 472
 - regex_iterator 471
 - regex_match() 473
 - regex_replace() 474, 676
 - regex_search() 473, 675
 - reguläre Ausdrücke 465
 - Reihenfolge
 - Auswertungs- 52, 60
 - der Initialisierung 163
 - umdrehen 698
 - rein virtuelle Funktion 292
 - mit Definition 293
 - reinterpret_cast<>() 202, 233, 315
 - Rekursion 114
 - Template-Metaprogrammierung 263, 266
 - rekursiver Abstieg 122
 - rekursiver Make-Aufruf 586
 - rel_ops 781
 - relationale Operatoren 47, 58, 781
 - remove()
 - Algorithmus 760
 - Datei/Verzeichnis (C) 766
 - Liste 815
 - remove_if() 760, 815
 - remove_reference 782, 802
 - remove_reference_t 803
 - remove_all() (Boost.Filesystem) 766
 - rename() Datei/Verzeichnis (C) 767
 - rend()

- Container 810, 856
 - Namespace std 855
 - string 890
 - replace() 759, 894
 - replace_copy(), -_if(), -_copy_if() 759
 - reserve() 813, 891
 - reset() 848
 - resetiosflags() 408
 - resize() 813, 815, 818, 891
 - return 37, 166
 - return value optimization *siehe* RVO
 - reverse() (list) 816
 - reverse(), reverse_copy() 698
 - reverse_iterator 810
 - Reverse-Iterator 856
 - Reversible Container 810
 - rfind() 895
 - right 402, 409
 - »rohes« Stringliteral 379
 - rotate(), rotate_copy() 694
 - round_error(), round_style() 762
 - RTTI 315
 - rule of three/five 641
 - runtime_error 329
 - rvalue *siehe* R-Wert
 - RVO 167, 449, 460
- S**
- scan_is(), scan_not() 877
 - Schleifen 73
 - und Container 94
 - do while 75
 - for 77
 - und Strings 207
 - Tabellensuche 90
 - terminierung 75
 - while 73
 - Schlüsselwörter 935
 - Schnittmenge 725
 - Schnittstelle 128, 962
 - einer Funktion 112
 - Regeln zur Konstruktion 631
 - scientific 402, 404, 409
 - scope 63
 - scoped locking 487
 - search() 716
 - search_n() 719
 - sed 589, 590
 - seekg(), seekp() 417
 - Seiteneffekt 68, 109, 207, 209
 - im Makro 140
 - Seitenvorschub 56
 - Sekunde 931
 - Selektion 67
 - sentinel 90, 202
 - sentry 426
 - Sequenz 67
 - Sequenzkonstruktor 170, 346
 - 2-dim. Matrixklasse 739
 - Sequenzmethoden 811
 - Server-Client
 - Beziehung 158
 - und callback 238
 - <set> 806, 831
 - set() 847
 - set_new_handler() 333
 - setbase() 408
 - set_difference() 725
 - setf() 403
 - setfill() 408
 - set_intersection() 725
 - setiosflags() 408
 - setjmp() 924
 - setprecision() 408
 - setstate() 415
 - set_symmetric_difference() 726
 - set_terminate() 330
 - set_union() 724
 - setw() 408
 - SFINAE 799, 962
 - shared_lock 501
 - shared_timed_mutex 501
 - shared_ptr 364, 902
 - für Arrays 644
 - SHELL 580
 - shift() 910
 - short 43
 - short string optimization 464
 - showbase, showpoint, showpos 402, 409
 - showContainer() 686
 - shrink_to_fit() 813, 818, 892

- shuffle() 695
- Sichtbarkeit 62
 - dateiübergreifend 130
- Sichtbarkeitsbereich (namespace) 149
- sign() 926
- Signal 512, 515
- Signalton 56
- Signatur 120, 276, 286–288, 290
- signed char 54
- sin(), sinh() 734, 913, 926
- single entry/ single exit 81
- Singleton 742
- size() 87, 92, 808
- size_t 48, 927
- size_type 807
- sizeof 201
 - nicht bei dynamischen Arrays 226
- Skalarprodukt 689
- skipws 402, 409
- sleep() 481
- sleep_for(), sleep_until() 476
- slice 914
- slice_array 916
- Slot 513, 515
- Smart Pointer 359
 - und Exceptions 642
- Socket 536
- Sommerzeit 931
- Sonderzeichen 56
- sort_heap() 730
- Sortieren
 - mit qsort() 237
 - mit sort() 707
 - stabiles 707
 - durch Verschmelzen 712
- Speicherklasse 130
- Speicherleck 215, 642
- Speicherplatzfreigabe 198
- Speicherverwaltung (eigene) 374
- Spezialisierung
 - von Klassen 274
 - von Templates 143
- Spezifikation einer Funktion 127
- splICE() 816
- split() 662
- Sprachumgebung 868
- SQL 561
- sqrt() 52, 734, 913, 926
- srand() 753, 928
- SSH 625
- <sstream> 419
- stable_partition() 705
- stable_sort() 708
- Stack 62
 - Klasse 258
- <stack> 779, 806, 818
- stack unwinding 323
- Standard Ein-/Ausgabe 99
- Standardbibliothek
 - C 150, 923
 - C++ 776
- Standardheader 151, 779
- Standardklassen 779
- Standard-Typumwandlung 61
 - Zeiger 240
- start() 914, 917
- static 130, 131
 - Attribute und Methoden 252
 - in Funktion 111
- static_assert 140
- static_cast<>() 56, 313
- static und -Bstatic (Makefile) 592
- statisches Binden 287, 963
- Statusabfrage einer Datei 415
- std 37, 64, 150
- <stdexcept> 329
- stdio 402
- Stelligkeit (Template) 265
- STL 427
- stod() 664
- stoi() 663, 664
 - und verwandte numerische
Konversionsfunktionen 897
- strcat(), strchr(), strcmp() 929
- strncat(), strncmp() 930
- strpbrk(), strrchr(), strstr() 930
- strcpy() 210, 929
- strcspn() 929
- stream 399
- Stream-Iterator 858
- Stream-Öffnungsarten 417
- streamsize 406

- strerror() 766, 925, 929
 - Streuspeicherung 836
 - strftime() 932
 - stride() 914, 917
 - String 91
 - in Zahl umwandeln 663
 - string 91, 889
 - append() 892
 - assign() 892
 - at() 92, 891
 - back() 891
 - begin() 890
 - capacity() 891
 - cbegin(), cend(), crbegin(), crend() 890
 - clear() 893
 - compare() 896
 - copy() 891
 - c_str() 891
 - data() 891
 - empty() 891
 - end() 890
 - erase() 893
 - find() 894
 - find...-Methoden 895
 - front() 891
 - insert() 893
 - length() 92, 891
 - max_size() 891
 - operator+=() 892
 - rbegin(), rend() 890
 - replace() 894
 - reserve() 891
 - resize() 891
 - rfind() 895
 - shrink_to_fit() 892
 - size() 92, 891
 - substr() 895
 - swap() 891
 - verketteten 92
 - <string> 779, 889
 - String-Klasse MeinString 245
 - Stringlänge 207, 208
 - Stringliteral 379, 872
 - stringstream 664
 - strlen() 205, 208, 929
 - strncpy() 210, 930
 - strtod(), strtol(), strtoul() 665, 928
 - strtok() 682, 930
 - struct 85, 281
 - Strukturanalyse 623
 - Stunde 931
 - Subobjekt 276, 282, 307
 - in virtuellen Basisklassen 309
 - verschiedene 308
 - Substitutionsprinzip 303
 - substr() 895
 - Subtyp 277, 282, 303, 963
 - Subversion 625
 - Suffix 49
 - sum() 910
 - swap() 350, 808, 891
 - Algorithmus 757
 - swap-Trick 231
 - swap_ranges() 757
 - switch 71
 - symmetrische Differenz
 - sortierter Strukturen 726
 - Synchronisation 485
 - Syntaxdiagramm 124, 125
 - ?: Bedingungsoperator 71
 - do while-Schleife 76
 - enum-Deklaration 83
 - for-Schleife 77
 - Funktionsaufruf 110
 - Funktionsdefinition 110
 - Funktionsprototyp 109
 - Funktions-Template 141
 - if-Anweisung 67
 - mathematischer Ausdruck 123
 - operator-Deklaration 338
 - struct-Definition 85
 - switch-Anweisung 72
 - Typumwandlungsoperator 358
 - while-Schleife 73
 - system() 928
 - system_clock 476
 - system_error 329, 415
 - Szenario 188
- T**
- Tabellensuche 90

- Tabulator 56
- Tag (des Monats) 931
- tan(), tanh() 734, 913, 926
- target (make) 577
- Taschenrechnersimulation 122
- Tastaturabfrage 99
- TCP 532
- »Teil-Ganzes«-Beziehung 659
- tellg(), tellp() 417
- Template
 - Alias 244
 - für Funktionen 140
 - #include 144
 - Instanziierung von T. 260
 - explizite 601, 602
 - ökonomische (bei vielen Dateien) 599
 - int-Parameter 261
 - für Klassen 258
 - Konstante 174
 - Metaprogrammierung 262
 - Spezialisierung 143
 - partielle 850
 - static Attribut 376
 - variable Parameterzahl 265, 740
- Template Method (Design-Muster) 301
- temporäres Objekt 169
 - Vermeidung 167
- terminate() 329
- terminate_handler 330
- Test Driven Development 605
- test() 848
- Test-Suite 607
- Textersetzung 136
- this 220
- this_thread 479
- this->, *this bei Zugriff auf Oberklassenelement 353
- thousands_sep() 879, 880
- Thread 475
- thread (API) 480
- Thread-Sicherheit 507
- thread_group 483
- ThreadGroup 484
- throw 324
- tie() 426
- tiefe Kopie 230
- __TIME__ 139
- time 876, 883
- time() 356, 932
- time_get 883
- time_put 884
- time_t 356, 931
- tm 356, 931
- tolower() 670, 875, 878, 924
- top() 819, 822
- to_string() (bitset) 848
- to_string() (Zahlkonvertierung) 897
- to_ulong() 848
- toupper() 670, 873, 875, 878, 924
- trailing return type 127
- traits 849
- traits::eof() 406
- transform() 757, 877
- tree (Programm) 584, 770
- Trennung von Schnittstellen und Implementation 130
- true 58
- true_type 798
- trunename() 879
- trunc 417
- try 324
- Tupel, <tuple> 740, 786
- tuple_cat() 741, 787
- Typ 963
 - polymorpher bzw. dynamischer Typ 316
- type cast *siehe* Typumwandlung
- Type Traits 797
 - Abfrage von Eigenschaften 800
 - Abfrage numerischer Eigenschaften 802
 - decay 803
 - enable_if 803
 - Funktionsweise 798
 - Typbeziehungen 802
 - Typumwandlungen 802
- typedef 243
- typeid() 316, 394
- type_info 316
- <typeinfo> 329
- typename (bei Template-Parametern) 141
- Typinformation 297

- zur Laufzeit 315
- Typumwandlung
 - cast 56, 198, 237, 312
 - cast-Schreibweise 57
 - durch Compiler 179
 - const_cast<>() 314
 - dynamic_cast<>() 313
 - mit explicit 169
 - implizit 71, 169
 - mit Informationsverlust 121
 - reinterpret_cast<>() 315
 - Standard- 61
 - Zeiger 240
 - static_cast<>() 313
- Typumwandlungskonstruktor 168, 179
- Typumwandlungsoperator 358
 - ios 416

U

- u16string, u32string 379
- UCS 872
- UDP 532, 541
- Überladen
 - von Funktionen 120
 - von Operatoren *siehe* operator
- Überlauf 46, 52
- Überschreiben
 - von Funktionen in
 - abgeleiteten Klassen 286
 - virtueller Funktionen 291
- Übersetzung 128, 129
- Übersetzungseinheit 132
- uint32_t und Verwandte 49
- Umgebungsvariable 218
- UML 273, 653
- Umleitung der Ausgabe auf Strings 419
- unärer Operator 339
- unäres Prädikat 699
- unary_negate 790
- UND
 - bitweises 47, 48
 - logisches 58
- #undef 136
- undefined behaviour 215
- underflow 52
- underflow_error 329

- unexpected() 327
- Unicode 399, 871
- uniform_int_distribution 750
- uniform_real_distribution 751
- union 97
- unique() Sequenzen 816
- unique(), -_copy() Algorithmen 696
- unique_lock 487
- unique_ptr 364, 464, 899
 - für Arrays 645
- Unit-Test 603
- unitbuf 402, 404, 409
- unordered_map 837
- unordered_multimap 842
- unordered_multiset 845
- unordered_set 843
- unsigned 44
- unsigned char 54
- Unterklasse 273, 963
- upper_bound() 721, 830, 834
- uppercase 402, 409
- URI, URL 532, 676
- URL-Codierung 546
- use case 188
- use_facet() 670–672, 870
- using
 - Deklaration 318
 - Namespace
 - Deklaration 149
 - Direktive 149
 - protected Funktion 317
 - statt typedef 243
- UTC 932
- UTF-8 871
- <utility> 272, 457, 462, 757, 782, 786

V

- Valarray
 - arithmetische Operatoren 911
 - Bit-Operatoren 912
 - logische Operatoren 912
 - mathematische Funktionen 913
 - relationale Operatoren 912
- <valarray> 907
- value_comp() 829, 834
- value_compare 827, 832

- value_type 807, 827, 832, 838
 - Variable 38
 - automatische 130
 - globale 130, 134
 - make 579
 - Variablenname 40
 - variadic templates 265, 740
 - vector 812
 - at() 88
 - push_back() 91
 - size() 87
 - <vector> 779, 806, 812
 - vector<bool> 813
 - Vektor 87
 - Klasse 343
 - Länge (geom.) 690
 - Verbundanweisung 66
 - Vereinigung (Menge) 724
 - Vererbung 273, 963
 - der abstrakt-Eigenschaft einer Klasse 293
 - von constraints 303
 - der Implementierung 318
 - Mehrfach- 305
 - private 317
 - protected 320
 - von Zugriffsrechten 280
 - und Zuweisungsoperator 389
 - Vergleich von double-Werten 635
 - verschmelzen (*merge*) 711
 - Versionskontrolle 624
 - Vertrag 304, 963
 - verwitwetes Objekt 215
 - Verzeichnis
 - anlegen 768
 - anzeigen 769
 - löschen 766
 - umbenennen 767
 - Verzeichnisbaum
 - anzeigen 770
 - make 585
 - Verzweigung 67
 - virtual 287, 289, 299
 - virtuelle Basisklasse 309
 - virtueller Destruktor 297
 - virtuelle Funktionen 287, 290
 - private 301
 - rein- 292
 - void 198
 - als Funktionstyp 109
 - void* 237
 - Typumwandlung nach 198
 - Typumwandlungsoperator 416
 - volatile 964
 - vollständiges Objekt 310, 311, 964
 - Vorbedingung 127, 964
 - vorgegebene Parameterwerte
 - in Funktionen 119
 - in Konstruktoren 163
 - Vorkommastellen 49
 - Vorrangregeln 59, 937
 - Vorwärtsdeklaration 193
- ## W
- Wächter (Tabellenende) 90, 202
 - Wahrheitswert 58
 - zufällig erzeugen 751
 - wait() 491, 499
 - Warteschlange 820
 - wchar_t 55, 872, 927
 - weak_ptr 904
 - Webserver 552
 - Weite der Ausgabe 402
 - Wert
 - eines Attributs 956
 - Parameterübergabe per 113
 - Wertebereich ganzer Zahlen 45
 - Wertsemantik 230, 429, 447
 - Performanceproblem 449
 - what() 328
 - while 73, 207, 208
 - whitespace *siehe* Zwischenraumzeichen
 - wide character 55
 - widen() 878
 - Widget 515
 - width() 402
 - Wiederverwendung
 - durch Delegation 320
 - durch Vererbung 283
 - wildcard 581
 - Winterzeit 931

- Wochentag 931
- wofstream 874
- Worttrennung 36
- Wrapperklasse für Iterator 856
- write() 233, 401
- ws 409
- wstring 379, 874, 889
- wxWidgets 510
- X**
- XOR, bitweises 47
- Y**
- yield() 481
- Z**
- Zahl in String umwandeln 667
- Zahlenbereich 44, 50
- Zeichen 54
- Zeichenkette 37, *siehe auch* string
 - C-String 204
 - Kopieren einer 208
- Zeichenklasse 468
- Zeichenkonstante 54
- Zeichenliteral 872
- Zeichensatz 871
- Zeiger 195, 211
 - vs. Array 201
 - auf Basisklasse 290, 297
 - auf Elementdaten 242
 - auf Elementfunktionen 241
 - auf Funktionen 236
 - hängender 214
 - intelligente *siehe* Smart Pointer
 - Null-Zeiger 198
 - auf Oberklasse 282, 289
 - auf ein Objekt (Mehrfachvererbung) 308
 - auf lokale Objekte 199
 - Parameterübergabe per 216
- Zeigerarithmetik 202
- Zeigerdarstellung
 - von [] 203
 - von [][] 224
- Zeile
 - einlesen *siehe* getline()
 - neue 56
- Zeit-Server 544
- Zeitkomplexität 964
- Ziel (make) 577
- Ziffernzeichen 54
- Zufallszahlen 749
- Zugriffsspezifizierer und -rechte 280
- zusammengesetzte Datentypen 83
- Zusicherung 139, 330
- Zustand 964
 - eines Iterators 434
- Zuweisung 37, 66, 69, 964
 - in abgeleiteten Klassen 282
 - und Initialisierung 165
- Zuweisungsoperator 166, 349, 641
 - implizite Deklaration 349, 389
 - und Vererbung 389
- zweidimensionale Matrix 736
- Zweierkomplement 44
- Zwischenraumzeichen 99, 205, 405