

Register

Symbole

* 46, 214, 453
*= 46, 372
+, +=, -, -= 46
++ 46, 374, 449, 453
, 60, 82, 503
-> 217
->* 263
-- 46, 378
.* 263
... *siehe* Ellipse
/, /= 46
/* ... */ 31
// 31
:: 62, 174, 311
::* 263
; 33, 71
<, <=, >, >= 46, 57
<=> 46, 388
<< 46, 106, 256, 361, 417
<<=, >>= 46
=, == 46, 57, 58
>> 46, 104, 256, 415, 440
?: 71
[] 218, 222, 366
[][]
Matrixklasse 404
Zeigerdarstellung 246

139
%, %= 46
& Adress-Operator 121
&, &= Bit-Operatoren 46
&& logisches UND 58
&& Shell 640
&& R-Wert 470
\ 31, 55, 137
\0 223, 227–229
\", \a, \b 55
\f, \n, \r, \t, \v 55, 104
\x 55
\\ 55
~ 46
^ 46
|= 46
|| 58
! 46, 58
!= 57, 58, 453
\$<, \$^, \$@ 637
@D, @F 647
” 33, 223, 224

A

abgeleitete Klasse 298, 306, 310
und virtueller Destruktor 321
Abhängigkeit (make) 634
automatische Ermittlung 639

abort() 947
abs() 52, 776, 777, 945, 946
abstrakter Datentyp 169, 969
abstrakte Klasse 315, 969
accumulate() 724
acos() 777, 945
acosh() 777
Adapter, Iterator- 889
Additionsoperator 359
adjacent_difference() 728
adjacent_find() 759
adjustfield 426
Adresse 214
 symbolische 34
Adressoperator 214
advance() 887
Aggregat 89, 220, 852
Aggregation 699, 969
Aktualparameter 116
<algorithm> 701, 790, 895
Algorithmus 28, 449
 accumulate() 724
 adjacent_difference() 728
 adjacent_find() 759
 all_of, any_of 738
 binary_search() 761
 clamp() 798
 copy(), copy_backward() 787
 copy_if(), copy_n() 789
 count(), count_if() 737
 equal() 775
 equal_range() 763
 exclusive_scan() 728
 fill(), fill_n() 722
 find(), find_if(), find_if_not() 754
 find_end() 758
 find_first_of() 755
 for_each(), for_each_n() 786
 generate(), _n() 723
 includes() 764
 inclusive_scan() 727
 inner_product() 726
 inplace_merge() 753
 iota() 723
 is_heap(), is_heap_until() 772
 is_partitioned() 745
 is_permutation() 740
 is_sorted(), is_sorted_until() 747
 iter_swap() 790
 lexicographical_compare() 742
 lexicographical_compare_three_way() 743
 lower_bound() 762
 make_heap() 771
 make_pair() 824
 make_tupel() 825
 max(...) 796
 max_element() 730
 merge() 752
 mergesort() 753
 min(...) 796
 min_element() 730
 minmax() 797
 minmax_element() 730
 mismatch() 773
 move(), move_backward() 822
 next_permutation() 739
 none_of 738
 nth_element() 750
 partial_sort(), -_copy 749
 partial_sum() 727
 partition() 744
 partition_copy(), partition_point() 744
 pop_heap() 770
 prev_permutation() 739
 push_heap() 771
 remove(), -_if(), -_copy(),
 -_copy_if() 794
 replace(), -_if(), -_copy(),
 -_copy_if() 792
 reverse(), reverse_copy() 735
 rotate(), rotate_copy() 731
 sample() 736
 search() 756, 758
 search_n() 760
 set_difference() 767
 set_intersection() 766
 set_symmetric_difference() 768
 set_union() 765
 shuffle() 732, 733
 sort() 747

- sort_heap() 772
- split() 702
- stable_partition() 744
- stable_sort() 747
- swap(), swap_ranges() 790
- transform() 791
- transform_exclusive_scan() 728
- transform_inclusive_scan() 728
- unique(), unique_copy() 733
- upper_bound() 762
- Alias-Name 121, 215, 217
 - *this 240
- alignment_of, alignof() 511
- all_of 738
- allgemeiner Konstruktor 178
- allocator 818
- alternative Funktions-Syntax 281
- Anführungszeichen 33, 223, 224
- Anker (bei regulärem Ausdruck) 522
- anonymer Namespace 150
- Anweisung 64
- any() (Bitset) 830
- any_of 738
- app 437
- append()
 - Pfad 806
 - String 926
- apply() 827
- Äquivalenz 388, 762
 - klasse 656
- arg() 776, 777
- argc 239
- Argument *siehe* Parameter
- Argument Dependent Lookup (ADL) 970
- argv[] 239
- Arität (Template) 497
- Arithmetik
 - mit Iteratoren 452
 - mit Zeigern 220
- arithmetische Operatoren 46
- Array
 - char 225
 - von C-Strings 225
 - dynamisches 230, 246, 247, 364
 - Freigabe 234
 - als Funktionsparameter 236, 243
 - mehrdimensionales 242, 246, 247
 - vs. Zeiger 219
 - zweidimensionales
 - Matrixklasse 404, 406
- Array2d 249
- <array> 842, 851
- ASCII 905
 - Dateien 256
 - Tabelle 56, 951
- asctime() 950
- asin() 777, 945
- asinh() 777
- assert() 139
- assign() 848, 927
- assignable_from 518
- Assoziation (UML) 696
- Assoziativität von Operatoren 59, 956
- async() 559
- at() 91, 94, 849, 852, 857, 867, 876, 925
- atan() 777, 945
- atan2() 945
- atanh() 777
- ate 437
- atexit() 947
- atof(), atoi(), atol() 947
- Atom-Uhr 601
- atomic 556
- atomic_ref 558
- Attribut
 - Compilersteuerung 957
 - einer Klasse 171, 970
- Aufforderung 970
- Aufzählungstyp 85
- Ausdruck
 - Auswertung 59, 60
 - Definition 40
 - mathematischer 52, 128
- Ausgabe 104, 106, 417
 - benutzerdefinierter Typen 418
 - Datei- 107, 413
 - Formatierung
 - mit Flags 424
 - mit std::format() 419
 - mit Manipulatoren 427
 - Weite der 421, 424
- Ausgabeoperator 361, 417

- Ausnahme 345, 970
- Auswertungsreihenfolge 52, 60, 956
- auto 99, 458, 680
 - als Rückgabetyt 155, 176, 313
- auto[] 101, 825
- automatische Variable 146
 - make 637
- Autotools (GNU) 653

- B**
- back() 849, 852, 854, 857, 859, 925
- back_inserter(), -insert_iterator 891
- Backslash
 - Zeichenkonstante \ 56
 - Zeilenfortsetzung 137
- Backspace 56
- bad() 436
- bad_alloc 351
- badbit 435
- bad_cast 336, 351
- bad_function_call 351
- bad_typeid 339, 351
- bad_weak_ptr 351
- base() 889
- basefield 426
- basic_string 923
- basic_-Streamklassen 414
- Basisklasse 298
 - und virtueller Destruktor 321
 - Konstruktor 330
 - Subobjekt 333
 - virtuelle 332
- Bedingungsausdruck 67, 68
- Bedingungsoperator ?: 71
- beg 438
- begin()
 - Container 455, 844
 - Namespace std 222, 845, 888
 - string 273, 924
 - vector 237
- Belegungsgrad 873
- benutzerdefinierte
 - Datentypen 84, 87
 - Klassen 361
 - Literale 401
 - Typen (Ausgabe) 418
 - Typen (Eingabe) 440
- Benutzungszählung 938
- Bereichsnotation 846, 972
- Bereichsoperator :: 62, 174, 311
 - namespace 148
- Bibliothek
 - C 162, 943
 - C++ 816
- Bibliotheksmodul 142
 - dynamisch 650
 - statisch 648
- Bidirectional-Iterator 883
- Big Three (Regel) 254
- Big Three/Five/Zero (Regel) 483
- Binärdatei 258
- binäre Ein-/Ausgabe 255
- binärer Operator 359
 - optimiert 480
- binäres Prädikat 734, 774, 896
- binäre Zahlendarstellung 47
- Binärzahl 43
- binary 109, 437
- binary_search() 761
- bind 833
- Binden 142
 - dynamisches *siehe* dynamisches B.
 - statisches *siehe* statisches Binden
- Bit
 - feld 101
 - Operatoren 46, 47
 - Verschiebung 47
 - pro Zahl 41
- bitset 827
- bitweises ODER, UND, XOR 46
- Block 31, 33, 62, 66, 203
 - und dyn. Objekte 233
- bool 57
- boolalpha 58, 425, 428
- break 72, 82
- bsearch() 947
- Bstatic (Makefile) 649
- Bucket 872
- bucket() 875
- Byte 54
 - std::byte 87, 946
 - Reihenfolge 602

C

- C++-Schlüsselwörter 953
- C-Arrays 218
- C-Funktionen einbinden 163
- C-Header 943
- C-String 223
- call wrapper 541
- Callback-Funktion 260, 835
- canonical() (Filesystem) 806
- capacity() 850, 926
- capturing group 520
- case 73
 - lokale Variablen 74
- <cassert> 139, 943
- cast *siehe* Typumwandlung
- catch 346
- cbegin()
 - Container 844
 - Namespace std 845, 888
 - string 925
- <cctype> 944
- <cctype> 791
- cdecl 267
- ceil() 945
- cend()
 - Container 844
 - Namespace std 845, 888
 - string 925
- cerr 104, 414
- char 54, 221
- char* 223
- CHAR_BIT (Bits pro Byte) 603
- char* const vs. const* char 217
- char_traits 224, 279, 513
- char8_t, char16_t, char32_t 400, 906
- <chrono> 532
- cin 33, 104, 414, 436
- clamp() 798
- class 172, 304
- class (bei Template-Parametern) 156
- clear() 436, 848, 863, 928
- Client-Server
 - Beziehung 173
 - und callback 260
- clock(), clock_t 950
- clog 104, 414
- close() 108
- CMake 653
- <cmath> 945
- <cmath> 52, 817, 943
- co_await 567
- code bloat 295
- Code-Formatierung 112
- collate 910
- combine() 904
- compare() 910, 930
- <compare> 388
- Compilationsmodell 295
- Compiler 29, 31, 34, 142, 171, 173
 - befehle 954
 - direktiven 134
 - und Templates 159
 - Typumwandlung 196
- <complex> 776
- Computerarithmetik 52
- Concepts 514
- conditional 513
- configure 653
- conj() 777
- connect() 572
- const 53
 - correctness 175
 - Elementfunktionen 172, 174
 - globale Konstante 147
- const& *siehe* Referenz auf const
- const_cast<>() 337
- const char* vs. char* const 217
- consteval 154
- constexpr
 - Funktion 152
 - Konstante 53
 - Konstruktor / Methode 190
- constinit 290
- const_iterator 843
- const_local_iterator 875
- const_pointer 843
- const_reference 843
- const_reverse_iterator 845, 890
- constraint, Vererben von 326
- Container 448
 - implizite Datentypen 843
 - Methoden 844

Container-Adapter 858
container_type 858
contains() 864
Contiguous-Iterator 883
continue 82
convertible_to 518
copy semantics 472
copy() (Filesystem) 808
copy() 787, 925
copy_backward() 787
copy_if() 789
copy_n() 789
copy_constructible 518
copy_options 809
copy_result (Ranges) 721
copysign() 945
co_return 565
Coroutinen 564
cos(), cosh() 776, 945
count() 830, 864
 Algorithmus 737
count_if() 737
cout 33, 104, 414
co_yield 564
__cplusplus 164
crbegin()
 Container 845
 Namespace std 845, 888
 string 925
create_directory() 809
crend()
 Container 845
 Namespace std 845, 888
 string 925
critical section 544
CRLF 605
<cstddef> 946
<cstddef> 48, 216
<cstdlib> 127, 259
c_str() 925
<cstring> 947
<cstring> 223, 261, 271, 274
<ctime> 949
<ctime> 376
ctype 708, 710, 910, 911
cur 438

curr_symbol 915
current_path 806
<cwctype> 944
CXXFLAGS 636

D

dangling (Ranges) 722
dangling pointer 233
data race 544, 546
data() 849, 852, 925
Datagramm 600
date_order() 917
Datei
 ASCII 256
 binär 258
 Ein-/Ausgabe 107, 413
 kopieren 109, 808
 löschen 806
 Öffnungsarten 437
 öffnen 109
 Positionierung 438
 schließen 108
 umbenennen 810
 Zugriffsrechte 803
Daten
 als Attributwerte 971
 static-Element- 284
Datenbankanbindung 621
Datenkapselung 971
Datentypen 34, 40, 169
 abstrakte *siehe* abstrakter
 Datentyp
 benutzerdefinierte 84
 int und unsigned 61, 71
 logische 57
 parametrisierte 156, 291
 polymorphe 319
 strukturierte 87
 zusammengesetzte 84
Datum
 Klasse / Gültigkeit 374
 regulärer Ausdruck 778
daytime (Port 13) 595
Deadlock 563
dec 425, 428
decay, decay_t 512

- decimal_point() 913, 914
 - decltype 281
 - decltype(auto) 283, 509, 822
 - declval 284
 - default (in switch-Anweisung) 73
 - = default 186
 - default constructor 176
 - default_delete<X[]> 686
 - defaultfloat 425
 - Default-Parameter *siehe* vorgegebene P.
 - #define 134, 137
 - Definition 144, 971
 - von static-Elementdaten 284
 - von Objekten 178
 - Deklaration 33, 34, 144, 971
 - einer Funktion 115
 - Funktionszeiger 258
 - Lesen einer D. 264
 - in for-Schleifen 80
 - Deklarationsanweisung 65
 - Dekrementierung 46
 - Dekrementoperator 378
 - Delegation 342
 - delegierender Konstruktor 189
 - delete 230, 232, 287, 320, 321
 - überladen 393
 - delete [] 234
 - = delete 186, 682
 - Deleter 686, 687
 - deque 856
 - <deque> 842, 856
 - Dereferenzierung 214, 260
 - derived_from 518
 - Destruktor 203, 331, 483
 - implizite Deklaration 203
 - und exit() 205
 - virtueller 320
 - detach() 537
 - Dezimalpunkt 49, 913
 - Dialog 584
 - diamond problem 333
 - difference_type 843
 - Differenz (Menge) 767
 - difftime() 950
 - digits, digits10 796
 - distance() 887
 - Distribution 639, 971
 - div(), div_t 946
 - divides 832
 - Division durch 0 352
 - DNS 593
 - do while 77
 - domain_error 351
 - double 49, 220
 - nicht als Laufvariable 81
 - korrekter Vergleich 678
 - downcast 336
 - Drei (die großen Drei) 254, 483
 - Drei-Wege-Vergleich *siehe* Spaceship-Operator
 - drop() 460
 - Dubletten entfernen 733
 - Durchschnitt (Menge) 766
 - DYLD_LIBRARY_PATH 651
 - dynamic_cast<>() 336
 - dynamic_pointer_cast 940
 - dynamischer Typ 339
 - dynamisches
 - Array 364
 - Binden 258, 311, 972
 - Datenobjekt 229
- ## E
- e, E 49
 - Editor 29
 - effizienter binärer Operator 480
 - egrep 519
 - Ein- und Ausgabe 104
 - Einbinden von C-Funktionen 163
 - Eingabe 415
 - Datei- 107, 413
 - von Strings 105
 - benutzerdefinierter Typen 440
 - Einschränkung *siehe* constraint
 - Elementdaten, Zeiger auf 264
 - Elementfunktion 170
 - als Funktionsobjekt 836
 - Zeiger auf 263
 - Ellipse
 - in catch-Klausel 347
 - template parameter pack 498
 - else 66
 - #else, #elif 134

emplace() 847, 858, 860, 861
emplace_back() 850, 854, 857
emplace_front() 854, 857
empty() 844, 858, 859, 861, 925
 Namespace std 845
enable_if 512
end 438
end()
 Container 455, 844
 Namespace std 222, 845, 888
 string 273, 924
 vector 237
#endif 136
endl 55, 430
 oder '\n'? 107
ends 430
ends_with() 930
ENTER 31, 105
»enthält«-Beziehung 332
enum 85
Environment, env[] 239
EOF 416
eof() 347, 416, 417, 436, 450
eofbit 435
epsilon() 796
equal() 775
equal_range() 763, 864
equalsIgnoreCase() 710
equal_to 833
erase() 848, 863, 928
ereignisgesteuerte Programmierung 570
errno 344
error_code 804
Exception 970
 arithmetische Fehler 352
 und Destruktor 345
 Handling 345
 Hierarchie 349
 Speicherleck durch Exc. 684
exception 349
<exception> 351
Exception-Sicherheit 355
exclusive_scan() 728
ExecutionPolicy 800
exists() (Filesystem) 807
exit() 127, 947

 und Destruktor 205
Exklusiv-Oder (Menge) 768
exp() 777, 945
explicit 186
explizite Instanziierung von
 Templates 296
Exponent 49–51
export 165
extension() (Filesystem) 806
extent 511
extern 145–147
extern "C" 164
extern template 295
external linkage 150
extract() 864

F

f, F 49
Fünf (die großen Fünf) 483
fabs() 945
Facette 910
fail() 436, 437
failbit 435, 442
fakultaet() 114
[[fallthrough]] 74
Fallunterscheidung 72
false 58
false_type 506
falsename() 913
Fehlerbehandlung 343
 Ein- und Ausgabe 434
Fibonacci 496, 729
__FILE__ 141
filename() 806
Filesystem
 canonical() 806
 copy() 808
 create_directory() 809
 current_path() 806
 exists() 807
 extension() 806
 filename() 806
 is_directory() 807
 recursive_directory_iterator 812
 remove() 806
 remove_all() 807

stem() 806
 fill() 424, 722
 fill_n() 722
 Filter 460
 filter() 460
 final 322, 682
 find()
 Algorithmus 754
 assoziative Container 863
 string 929
 find_end() 758
 find_first_of() 755
 find...-Methoden (string) 929
 fixed 425, 426, 428
 Fixture 665
 flache Kopie 252
 flags() 425
 flip() 851
 Bitset 829
 float 49
 floatfield 426
 floating_point 518
 floor() 945
 flush() 107
 flush (Manipulator) 430
 fmod() 945
 fmtflags 425
 Fold-Expression 501
 for 79
 Kurzform 98
 foreach(), for_each_n() 786
 Formalparameter 116
 Formatierung 419
 forward() 822
 Forward-Iterator 882
 <forward_list> 842
 forwarding reference 479
 frac_digits() 915
 Fragmentierung (Speicher) 234
 Framework 571
 free store 230
 free() 397
 frexp() 945
 friend 277
 from_chars() 705
 front() 849, 852, 854, 857, 859, 925

front_inserter(), _insert_iterator 891
 fstream 414, 438
 <fstream> 107
 Füllzeichen 420, 424
 __func__ 141
 function 835
 <functional> 351, 832
 Funktion 114
 frei oder global 149
 mit Gedächtnis (static) 117
 mit initializer_list 846
 klassenspezifische 284
 mathematische 52, 945
 Parameterübergabe
 per Referenz 121
 per Wert 119
 per Zeiger 235
 vorgegebene Parameterwerte 124
 rein virtuelle 315
 mit Definition 316
 static 284
 alternative Syntax 281
 Überschreiben 310
 virtuelle *siehe* virtuelle
 Funktionen
 Funktionsobjekte 386, 431
 function 835
 mem_fn 836
 Funktions-Template 156
 Funktor *siehe* Funktionsobjekte
 future 562
 <future> 351

G

Ganzzahlen 41
 garbage collection 234
 gcd() 799
 gegenseitige Abhängigkeit von
 Klassen 210
 Genauigkeit 50
 Generalisierung 298
 generate(), generate_n() 723
 generische Programmierung 448
 GET (http) 605
 get() 104, 256, 415, 416
 getenv() 947

getline() für Strings 106, 931
 getline(char*,...) 416
 getloc() 915
 get_money() (Manipulator) 428
 get_monthname() 917
 get_time() (Manipulator) 428
 get_time(), get_weekday(), get_year()
 (locale) 917
 ggT 76
 std::gcd() 799
 schnell 198
 Gleichheitsoperator bei Vererbung 410
 Gleichverteilung 782
 Gleitkommazahl 53
 Syntax 49
 global 62
 Funktion 149
 Namensraum 162
 Variable 145, 146
 glvalue 469
 gmtime() 950
 GNU Autotools 653
 good() 436
 goodbit 435
 goto 972
 Grafische Benutzungsschnittstelle 569
 greater, greater_equal 833
 greedy (regex-Auswertung) 522
 Grenzwerte von Zahltypen 795
 Groß- und Kleinschreibung 31, 41
 größter gemeinsamer Teiler *siehe* ggT
 grouping() 913, 914
 guard (Threads) 545
 Gültigkeitsbereich 127
 Block 62
 Datei 150
 Funktion 116
 Klassen 172
 und new 233

H

Hängender Zeiger 233
 hardware_concurrency() 533
 has_facet() 904
 hash() 911
 hasher 874

Hash-Funktion 872, 874
 has_infinity 796
 __has_include 136
 has_sort_function 507
 has_value() (optional) 839
 »hat«-Beziehung 699
 Header 33, 162
 Datei 142
 Inhalt 145
 Http 605
 der Standardbibliothek 816
 Heap 768
 hex 425, 428
 Hexadezimalzahl 43
 hexfloat 425
 Host Byte Order 602

I

-I Compileroption 134
 iconv 909
 IDE 30, 36
 Identität von Objekten 169, 972
 identity 721
 IEC 60559, IEEE 754 50
 if 66
 if constexpr 71, 500
 #if, #ifdef, #ifndef 134
 ifstream 107, 414
 ignore() 416
 imag() 776, 777
 imbue() 709, 902
 Implementation 142
 -svererbung 340
 -sdatei, Inhalt 145
 implizite Deklaration
 Destruktor 203
 Konstruktor 176
 Zuweisungsoperator 369, 409
 import 165
 in 437
 #include 32, 134
 Include-Guard 135
 includes() 764
 inclusive_scan() 727
 Indexoperator 90, 218, 222, 246, 366
 index_sequence 830

- infinity() 796
- Initialisierung
 - array 852
 - direkte I. der Attribute 177
 - und virtuelle Basisklassen 333
 - C-Array 220, 243
 - einfacher Datentypen 44
 - mit Element-Initialisierungsliste 179
 - Konstante in Objekten 180, 285
 - globaler Konstanten 145, 147
 - Reihenfolge 290
 - mit {}-Liste 187, 846, 888
 - mit konstruktor-interner Liste 285
 - von Objekten 176
 - von Referenzen 961
 - Reihenfolge der Initialisierung
 - von Attributen 179
 - von Funktionsargumenten 133
 - in for-Schleife 79
 - von static-Elementdaten 284
 - struct 88
 - und Vererbung 302
 - und Zuweisung 45, 182
- initializer_list 188, 366, 797, 846
 - für zweidimensionales Array 846
- Inklusionsmodell 295
- Inkrementierung 46
- Inkrementoperator 374
- inline
 - Elementfunktion 175
 - Funktion 151
 - Konstante 147, 288
 - Variable 151, 288
- inner_product() 726
- innere Klasse 456, 623
- inplace_merge() 753
- Input-Iterator 882
- insert() 848, 863, 871, 927
- inserter() 892
- insert_iterator 892
- insert_or_assign 867, 876
- Instanz 168, 972
- Instanziierung von Templates 293
 - explizite 296
 - ökonomische (bei vielen Dateien) 295
- int 33, 41
- int-Parameter in Templates 293
- intX_t, int_fastX_t, int_leastX_t
 - (X = 8, 16, 32, 64), intmax_t 48
- integer_sequence 830
- integral 518
- integral_constant 506
- integral promotion 61
- Interface (UML) 694
- internal 425, 428
- internal linkage 150
- Internet-Anbindung 591
- Intervall (und Notation) 972
- invalid_argument 351
- <iomanip> 428, 430
- ios 414, 434, 436
 - failure 436
 - Flags zur Dateipositionierung 438
 - Methoden 425, 436
- <ios> 428
- ios_base 414
 - binary 109
 - Fehlerstatusbits 435
 - Flags 425, 426
 - Manipulatoren 428
- iostate 434
- <iostream> 33, 414, 415, 418, 428
- iota, iota_view 723
- IPv4, IPv6 593
- IPv4-Adresse (regulärer Ausdruck) 780
- is() 911
- is_abstract 511
- isalnum(), isalpha() 910, 944
- is_arithmetic 509, 510, 515
- is_array 510
- is_base_of 511
- isblank() 944
- is_bounded 796
- is_class 506, 510
- iscntrl() 910, 944
- is_const 511
- is_convertible 511
- isdigit() 132, 184, 910, 944
- is_directory() (Filesystem) 807
- is_enum 510

is_exact 796
is_final 511
is_floating_point 510
is_function 510
is_fundamental 511
isgraph() 910, 944
is_heap(), is_heap_until() 772
is_iec559, is_integer 796
is_integral 510
islower() 910, 944
is_lvalue_reference 510
is_modulo 796
is_null_pointer 510
ISO 10646 906
ISO 8859-1, ISO 8859-15 905
is_partitioned() 745
is_permutation() 740
is_pointer 510
is_polymorphic 511
isprint() 910, 944
ispunct() 910
is_reference 510
is_rvalue_reference 510
is_same 511
is_signed 511, 796
is_sorted(), is_sorted_until() 747
isspace() 910, 944
ist-ein-Beziehung 298, 307, 325
istream 414, 415
Istream-Iterator 892
istream::seekg(), tellg() 438
istream::ws 430
istreamstringstream 414, 439
is_unsigned 511
isupper() 910, 944
is_void 510
isxdigit() 910, 944
iter_swap() 790
Iterator 273, 449, 453, 881
 Adapter 889
 Bidirectional 883
 Contiguous 883
 Forward 882
 Input 882
 Insert 890
 Output 882

 Random Access 883
 Reverse 889
 Stream 892
 Tag 883
 Zustand 454
iterator 843
iterator_category 882
<iterator> 881

J

Jahr 949
join() 536, 537
jthread (Klasse) 538

K

Kardinalität 697
Kategorie (locale) 910
key_equal 874
key_type 863, 870
key_comp(), key_compare 864
Klammerregeln 59
Klasse 168, 171, 973
 abgeleitete *siehe* abgeleitete
 Klasse
 abstrakte 315
 Basis- *siehe* Basisklasse
 Deklaration 173
 innere 456, 623
 konkrete 315
 Ober- *siehe* Oberklasse
 für einen Ort 172
 Unter- *siehe* Unterklasse
 für rationale Zahlen 194
Klassenname (typeid) 339
klassenspezifische
 Daten und Funktionen 284
 Konstante 288
Klassen-Template 291
Klassifikation 298, 973
Kleinschreibung 31
kleinstes gemeinsames Vielfaches
 siehe lcm()
Kollisionsbehandlung 872
Kommandointerpreter 635
Kommandozeilenparameter 239
Kommaoperator 60, 82, 503

- Kommentar 31
 - komplexe Zahlen 776
 - Komplexität 978
 - Komposition 700
 - konkrete Klasse 315
 - Konsole 30
 - auf UTF-8 einstellen 905
 - Konstante 53
 - globale 145, 147
 - klassenspezifische 288
 - konstante Objekte 174
 - Konstruktor 173, 176
 - allgemeiner 178
 - implizite Deklaration 176
 - delegierender 189
 - erben 308
 - Kopier- *siehe* Kopierkonstruktor
 - vorgegebene Parameterwerte 178
 - Typumwandlungs- *siehe* Typumwandlungskonstruktor
 - Kontrollstrukturen 64
 - Konvertieren von Datentypen *siehe* Typumwandlung
 - Kopie, flache/tiefe 252
 - Kopieren
 - von Dateien 109
 - von Objekten 252
 - von Zeichenketten 228
 - Kopierkonstruktor 181, 367, 483
 - Auslassung durch Compiler 183
 - Kreuzreferenzliste 718
 - kritischer Bereich 544
 - Kurzform-Operatoren 45
- L**
- l, L 49
 - Länge eines Vektors 726
 - Lambda-Funktionen 485
 - LANG 902
 - late binding 258
 - Laufvariable 79, 80
 - Laufzeit 214
 - und Funktionszeiger 258
 - und new 230
 - und Polymorphie 311
 - Typinformation 338
 - lcm() 799
 - LD_LIBRARY_PATH 651
 - ldd 651
 - ldexp() 945
 - ldiv(), ldiv_t 946
 - left 425, 428
 - length() (C-String, constexpr) 224, 279, 513
 - length() (string) 925
 - length_error 351
 - less, less_equal 833
 - lexicographical_compare() 742
 - lexicographical_compare_three_way() 743
 - lexikografischer Vergleich 390, 973
 - <limits> 42, 49, 795
 - __LINE__ 141
 - Linken 146, 162
 - dynamisches 650, 973
 - internes, externes 150
 - statisches 648, 973
 - Linker 35
 - linksassoziativ 59, 242
 - list 854
 - <list> 819, 842
 - Liste
 - Initialisierungs- 179, 285
 - Initialisierungs- (bei C-Arrays) 243
 - Liste (Klasse) 455
 - Literal 224, 974
 - benutzerdefiniert 401
 - String 400
 - Zahl- 53
 - Zeichen- 54, 906
 - load_factor() 875
 - local_iterator 874, 875
 - <locale> 901
 - localtime() 376, 950
 - log(), log10() 776, 945
 - logic_error 350, 351
 - logical_and, !_not, !_or 833
 - logischer Datentyp 57
 - logische Negation 58
 - lokal (Block) 62
 - lokale Objekte 217
 - long 41

long double 49

lower_bound() 762, 865

lvalue 469

L-Wert 470

M

magic number 974

main() 31, 32, 127

MAKE 646

make 633

 automatische Ermittlung von

 Abhängigkeiten 639

 parallelisieren 647

 rekursiv 646

 Variable 636

Makefile 142, 634

make_from_tuple() 826

make_heap() 771

make_index_sequence 830

make_integer_sequence 830

make_pair() 824

make_shared 685, 939

make_tuple() 825

make_unique 268, 685, 937

Makro 137

malloc() 397

Manipulatoren 427

Mantisse 50

<map> 842, 865

mapped_type 867

match_results 526

mathematischer Ausdruck 52

mathematische Funktionen 945

Matrix

 C-Array 242

 C-Array, dynamisch 246

 Klasse 249

max() 796

max(initializer_list<T>) 797

max_bucket_count() 875

max_element() 730

max_exponent, max_exponent10 796

max_load_factor() 875

max_size() 844

[[maybe_unused]] 957

mehrdimensionales Array 242

Mehrfachvererbung 300, 328, 331

MeinString (Klasse) 271

mem_fn() 836

member function *siehe* Element-
funktion

memcpy() 366

memory leak 233, 684

<memory> 351, 819

Mengenoperationen auf sortierten
Strukturen 764

merge() 752, 855, 864

mergesort() 753

messages 910, 919

Metaprogrammierung 493

Methode 168, 170, 974

 Regeln zur Konstruktion

 von Prototypen 675

midpoint() 799

MIME 974

min() 796

min(initializer_list<T>) 797

min_element() 730

min_exponent, min_exponent10 796

MinGW-Konsole 35

minmax(...) 797

minmax_element() 730

minus 832

Minute 949

mischen 752

mismatch() 773

mkdir 641

mktime() 950

modf() 945

Modulare Programmgestaltung 141

Module 164

Modulo 46

modulus 832

Monat 949

monetary 910, 914

money_get 915

money_punct 914

money_put 916

Monitor-Konzept 554

move semantics 472

move() 476

- move(), move_backward()
 - Container-Bereich 822
 - move_constructible 518
 - moving constructor 475
 - mt19937 733, 781
 - multimap 869
 - multiplies 832
 - Multiplikationsoperator 372
 - Multiplizität (UML) 697
 - multiset 871
 - mutable
 - Attribut 174
 - Lambda-Funktion 490
 - mutex 544
- N**
- '\n' 33
 - oder endl? 107
 - und regex_replace 528
 - Nachbedingung 133, 974
 - Nachkommastellen 49
 - precision 427
 - Name 41
 - einer Klasse (typeid) 339
 - name() 339, 904
 - Namenskonflikte bei Mehrfach-
vererbung 331
 - Namenskonventionen 41
 - Namespace 63, 147
 - anonym 150
 - in Header-Dateien 150
 - Verzeichnisstruktur 643
 - namespace 32, 147
 - namespace std 63
 - NaN (not a number) 389
 - narrow() 912
 - naionale Sprachumgebung 902
 - NDEBUG 140
 - negate 832
 - Negation
 - bitweise 46, 47
 - logische 58
 - negative_sign() 915
 - neg_format() 915
 - Network Byte Order 602
 - Netzwerkprogrammierung 591
 - neue Zeile 56, 66
 - new 229, 232, 287
 - Fehlerbehandlung 354
 - Placement-Form 941
 - überladen 393
 - <new> 941
 - new_handler 354
 - <new> 351
 - next() 887
 - next_permutation() 739
 - noboolalpha 428
 - Nodehandle 864
 - [[nodiscard]] 957
 - noexcept 348
 - none() (Bitset) 830
 - none_of 738
 - norm() 776, 777
 - Normalverteilung 784
 - noshowbase, -point, -pos 428
 - noskipws 428
 - not_equal_to 833
 - nothrow 355
 - notify_one(), _all() 550, 691
 - nounitbuf, nouppercase 428
 - npos 924
 - nth_element() 750
 - NTP – Network Time Protocol 602
 - NULL 216, 946
 - nullopt 839
 - nullptr 216
 - Null-Zeiger und new 355
 - <numbers> 191
 - numeric 910, 912
 - numeric_limits 49, 795
 - numerische Auslöschung 51
 - numerische Umwandlung 703, 706, 931
 - num_get, num_put 912
 - NumeriertesObjekt (Klasse) 284
 - numpunct 913
- O**
- O-Notation 978
 - Oberklasse 298, 310, 975
 - Subjekt einer 302
 - Subtyp einer 306
 - Zugriffsrechte vererben 304

- Oberklassenkonstruktor 299, 302
 - object slicing 307, 315, 409
 - Objekt 28, 169, 173, 975
 - code 34
 - dynamisches 229
 - als Funktions- 386
 - hierarchie 332
 - Identität *siehe* Identität von Objekten
 - Initialisierung 176
 - konstantes 174
 - orientierung 167
 - Übergabe per Wert 182
 - verkettete Objekte 232
 - verwitwetes 233
 - vollständiges 333, 334, 978
 - Objektcode 35
 - oct 425, 428
 - ODER
 - bitweises 46
 - logisches 58
 - Öffnungsarten für Streams 437
 - offsetof 946
 - ofstream 107, 414
 - Oktalzahl 43
 - omanip 430
 - one definition rule 144
 - open() 107
 - Open Source 975
 - Operator
 - arithmetischer 46
 - binärer 359
 - Bit- 46
 - für char 57
 - als Funktion 358
 - Kurzform 47
 - für Literale 399
 - für logische Datentypen 58
 - Präzedenz 59, 955
 - relationale 46, 58
 - Syntax 358
 - Typumwandlungs- 378
 - unärer 359
 - für ganze Zahlen 45
 - operator delete() 393
 - operator new() 393
 - operator string() 379
 - operator>() 386
 - operator*() 373, 380, 453
 - operator*=() 372
 - operator++() 374, 377, 453
 - operator++(int) 680
 - operator+=() 361
 - operator->() 380
 - operator<=>() 388
 - operator<<() 361, 417
 - operator=() 370
 - operator==() 453, 830
 - bei Vererbung 410
 - operator>>() 415
 - operator[]() 406, 849, 852, 857, 883
 - operator!=() 453
 - Optimierung durch Vermeiden
 - temporärer Objekte 183
 - optional 839
 - Ort (Klasse) 172
 - ostream 361, 414, 417
 - ostream::endl, ends, flush() 430
 - Ostream-Iterator 892
 - ostream::seekp(), tellp() 438
 - ostreamstream 414, 439
 - osyncstream 552
 - out 437
 - out_of_range 351
 - Output-Iterator 882
 - overflow 44, 51
 - overflow_error 351
 - override 313, 681
- ## P
- packaged_task 562
 - pair 823
 - Parameter
 - expansion 499
 - einer Funktion 115
 - Pack 498
 - übergabe
 - per Referenz 121
 - per Wert 119
 - per Zeiger 235
 - parametrisierte Datentypen 156, 291
 - »part-of«-Beziehung 699

- partial_ordering 389
 - partial_sort(), partial_sort_copy 749
 - partial_sum() 727
 - partielle Spezialisierung von
 - Templates 882
 - partition() 744
 - partition_copy(), partition_point() 744
 - path 803
 - patsubst 639
 - peek() 417
 - perfect forwarding 822
 - Performance 465
 - Permutationen 739
 - Pfeiloperator 217
 - PHONY 636
 - π , pi, pi_v 191
 - Pipe (ranges) 460
 - Placement new/delete 941
 - plus 832
 - pointer 843
 - Pointer, smarte 380
 - polar() 777
 - polymorpher Typ 319, 339
 - Polymorphismus 311, 975
 - pop() 858, 860, 861
 - pop_back() 850, 854, 857
 - pop_front() 854, 857
 - pop_heap() 770
 - portabel (Zeichensatz) 908
 - pos_format() 915
 - Positionierung innerhalb einer Datei 438
 - positive_sign() 915
 - POSIX 902
 - POST (http) 610
 - postcondition *siehe* Nachbedingung
 - Postfix-Operator 374
 - pos_type 438
 - pow() 777
 - Prädikat
 - Algorithmus mit P. 896
 - binäres 734, 774, 896
 - unäres 737
 - Präfix-Operator 374
 - Präprozessor 133, 639
 - Präzedenz von Operatoren 59, 955
 - precision() 426
 - precondition *siehe* Vorbedingung
 - prev() 887
 - prev_permutation() 739
 - PRINT (Makro) 139
 - printf() 497
 - Priority-Queue 860
 - private 172, 303
 - private Vererbung 340
 - Programm
 - ausführbares 35
 - Strukturierung 113
 - Programmierrichtlinien 112, 684
 - Projection (Ranges) 721
 - proj() 777
 - Projekt 37, 142
 - promise 562
 - protected 303
 - protected-Vererbung 342
 - Prototyp
 - Funktions- 114
 - einer Methode 172
 - Regeln zur Konstruktion 675
 - prvalue 470
 - ptrdiff_t 845, 946
 - public 172, 303
 - Pufferung (Ein-/Ausgabe) 104, 418
 - push() 858, 859, 861
 - push_back() 850, 854, 857
 - vector 94
 - push_front() 854, 857
 - push_heap() 771
 - put() 107, 256, 418
 - putback() 416
 - put_money(), _time() (Manipulator) 428
- ## Q
- qsort() 259, 947
 - Qt 571
 - QThread 587
 - Quantifizierer 522
 - Quellcode 34
 - Queue 859
 - <queue> 819, 842, 859
 - quoted() 428

R

- race condition 544, 563
- radix 796
- RAII 545, 665, 685, 976
- <random> 733
- random access 454
- Random-Access-Iterator 883
- random_access_iterator 518
- random_device 781
- range based for 98
- range_error 351
- Ranges (Bereiche), <ranges> 459
- rank 511
- <ratio> 837
- Rationale Zahl
 - Klasse 194
 - Template std::ratio 837
- rbegin()
 - Container 845, 889
 - Namespace std 845, 888
 - string 925
- rdstate() 436
- read() 255
- real() 776, 777
- Rechengenauigkeit 50, 81
- rechtsassoziativ 59, 242
- recursive_directory_iterator 812
- reelle Zahlen 49
- ref(), reference_wrapper 541, 838
- reference
 - bitset 828
 - Container 843
 - vector<bool> 850
- reference collapsing rules 478
- reference counting 938
- Referenz 121, 131
 - auf Basisklasse 314
 - auf const 122, 181
 - auf istream 415, 440
 - auf Oberklasse 306, 314
 - auf ostream 362, 417
 - Parameterübergabe per 121
 - Rückgabe per 367
 - auf R-Wert 472
 - semantik 253, 465
 - weiterleitende 479
- Regel der 0/3/5 *siehe* Rule of 0/3/5
- regex_iterator 526
- regex_match() 527
- regex_replace() 528, 713
- regex_search() 528, 712
- <regex> 351, 527
- regex_iterator 703
- reguläre Ausdrücke 519
- Reihenfolge
 - Auswertungs- 51, 60
 - der Initialisierung
 - von Attributen 179
 - von Funktionsargumenten 133
 - umdrehen 735
- rein virtuelle Funktion 315
 - mit Definition 316
- reinterpret_cast<>() 221, 255, 338
- Rekursion 120
 - Template-Metaprogrammierung 494, 499
- rekursiver Abstieg 128
- rekursiver Make-Aufruf 646
- relationale Operatoren 46, 58
- remove()
 - Algorithmus 794
 - Datei/Verzeichnis 806
 - Liste 855
- remove_all() Dateien/Verzeichnisse 807
- remove_const 511
- remove_if() 794, 855
- remove_reference 511, 822
- rename() Datei/Verzeichnis 810
- rend()
 - Container 845, 889
 - Namespace std 845, 888
 - string 925
- replace() 792, 928
- replace_copy(), _if(), _copy_if() 792
- requires 515
- reserve() 850, 926
- reset() (Bitset) 829
- resetiosflags() 428
- resize() 850, 855, 857, 926
- resume() 564
- return 33, 183
- Return Value Optimization (RVO) 183

reverse() (list) 855
 reverse(), reverse_copy() 735
 reverse (Ranges) 460
 reverse_iterator 845
 Reverse-Iterator 889
 Reversible Container 845
 right 425, 428
 »rohes« Stringliteral 400
 rotate(), rotate_copy() 731
 round_error(), round_style() 796
 RTTI 338
 Rückgabetyyp auto 155, 176, 313
 Rule of zero/three/five 483
 runtime_error 351
 rvalue 470
 R-Wert, Referenz auf R-Wert 470

S

safe_iterator_t 722
 safe_subrange_t 722
 same_as 517
 sample() 736
 scan_is(), scan_not() 911
 Schleifen 75

- und Container 97
- do while 77
- for 79
- und Strings 226
- Tabellensuche 92
- terminierung 77
- while 75

 Schlüsselwörter 953
 Schnittmenge 766
 Schnittstelle 142, 976

- einer Funktion 118
- Regeln zur Konstruktion 675

 scientific 425, 426, 428
 scope 62
 scoped locking 545
 scoped_lock 545
 search() 756, 758
 search_n() 760
 seekg(), seekp() 438
 Seiteneffekt 68, 115, 133, 227, 229

- im Makro 140

 Seitenvorschub 56

Sekunde 949
 Selektion 66
 sentinel 92, 221
 Sequenz

- konstruktor 187, 188, 366
- methoden (Container) 848

 Server-Client

- Beziehung 173
- und callback 260

 set() (Bitset) 829
 <set> 842, 869
 setbase() 428
 set_difference() 767
 setf() 425, 426
 setfill() 428
 set_intersection() 766
 setiosflags() 428
 setprecision() 428
 setstate() 436
 set_symmetric_difference() 768
 set_terminate() 352
 set_union() 765
 setw() 428
 SFINAE 507, 976
 shared_ptr 385, 684, 938

- für Arrays 685

 short 41
 showbase, showpoint, showpos 425, 428
 showContainer() 508
 shrink_to_fit() 850, 857, 926
 shuffle() 732, 733
 Sichtbarkeit 62

- sbereich (namespace) 147

 sign() 945
 Signal 572, 575
 Signalton 56
 Signatur 125, 300, 311, 314, 976
 signed char 54
 signed_integral 518
 sin(), sinh() 776, 945
 single entry/ single exit 83
 size()

- Container 844
- Namespace std 90, 219, 845
- string 94
- vector 90

- size_t 48, 946
- size_type 843
- sizeof 220
 - nicht bei dynamischen Arrays 247
- sizeof... (variadische Templates) 499
- Skalarprodukt 726
- skipws 425, 428
- sleep_for(), sleep_until() 532, 537
- Slot 572, 575
- Smart Pointer 380
 - und Exceptions 684
- Socket 595
- Sommerzeit 949
- Sonderzeichen 56
- sort_heap() 772
- sortable 517
- Sortieren
 - mit qsort() 260
 - mit sort() 747
 - stabiles 747
 - durch Verschmelzen 753
- source_location 140
- Spaceship-Operator 388, 844
- span 237, 879
- 842
- Speicher
 - klasse 146
 - leck 233, 684
 - platzfreigabe 217
- Spezialisierung
 - von Klassen 298
 - von Templates 158
- splICE() 855
- split() 702
- Sprachumgebung 902
- SQL 621
- sqrt() 52, 777, 945
- sregex_iterator 703
- ssize() 98, 845
- <sstream> 440
- stable_partition() 744
- stable_sort() 747
- Stack 62
 - Klasse 291
- <stack> 819, 842, 858
- stack unwinding 345
- Standard
 - bibliothek
 - C 162, 943
 - C++ 816
 - header 163, 819
 - klassen 819
 - Typumwandlung 60, 262
 - Zeiger 262
- Standard Ein-/Ausgabe 104
- starts_with() 930
- static 146
 - Attribute und Methoden 284
 - in Funktion 117
 - Initialisierungsreihenfolge 290
- static_assert 140
- static_cast<>() 56, 335
- static und -Bstatic (Makefile) 649
- statisches Binden 311, 976
- Statusabfrage einer Datei 436
- std 32, 63, 149
- <stdexcept> 351
- stdio 425
- Stelligkeit (Template) 497
- stem() (Filesystem) 806
- Stichprobe 736
- STL 447
- stod() 704
- stoi() 703
 - und verwandte numerische Konversionsfunktionen 932
- stop_requested(), stop_token() 538
- strcat(), strchr(), strcmp() 948
- strncat(), strncmp() 949
- strpbrk(), strrchr(), strstr() 948
- strcpy() 229, 948
- strcspn() 948
- Stream
 - Öffnungsarten 437
 - Iterator 892
- stream 413
- streamsize 415
- strerror() 948
- Streuspeicherung 872
- strftime() 950
- String 94
 - in Zahl umwandeln 703

- Klasse MeinString 271
 - Länge 226, 227
 - Literal 400, 906
 - string 94, 923
 - append() 926
 - assign() 927
 - at() 94, 925
 - back() 925
 - begin() 924
 - capacity() 926
 - cbegin(), cend(), crbegin(), crend() 925
 - clear() 928
 - compare() 930
 - copy() 925
 - c_str() 925
 - data() 925
 - empty() 925
 - end() 924
 - erase() 928
 - find() 929
 - find_...-Methoden 929
 - front() 925
 - insert() 927
 - length() 94, 925
 - max_size() 925
 - operator+=() 926
 - rbegin(), rend() 925
 - replace() 928
 - reserve() 926
 - resize() 926
 - shrink_to_fit() 926
 - size() 94, 925
 - substr() 930
 - swap() 925
 - <string> 819
 - stringstream 705
 - string_view 280, 933
 - <string_view> 819
 - strlen() 224, 227, 948
 - strncpy() 229, 949
 - strong_ordering 389
 - strtod(), strtol(), strtoul() 947
 - strtok() 948
 - struct 87, 304
 - Strukturierte Bindung 101
 - Stunde 949
 - Subobjekt 299, 306, 330
 - in virtuellen Basisklassen 332
 - verschiedene 331
 - subrange 462
 - Substitutionsprinzip 325
 - substr() 930
 - Subtyp 300, 306, 325, 977
 - Suffix 49
 - suspend_always 566
 - swap() 370, 844, 925
 - Algorithmus 790
 - swap-Trick 254
 - swap_ranges() 790
 - switch 72
 - symmetrische Differenz (Menge) 768
 - Synchronisation 543
 - Syntax 54
 - Syntaxdiagramm
 - ?: Bedingungsoperator 71
 - do while-Schleife 77
 - enum-Deklaration 85
 - for-Schleife 79
 - Funktions-
 - aufruf 116
 - definition 115
 - prototyp 115
 - Template 156
 - if-Anweisung 67
 - mathematischer Ausdruck 129
 - operator-Deklaration 358
 - struct-Definition 87
 - switch-Anweisung 72
 - Typumwandlungsoperator 379
 - while-Schleife 75
 - system() 947
 - system_clock 532
 - system_error 351, 436
 - Szenario 206
- ## T
- Tabellensuche 92
 - Tabulator 56
 - Tag (des Monats) 949
 - Tag Dispatching (und Alternative) 884
 - tan(), tanh() 776, 945

- target (make) 634
- Taschenrechnersimulation 128
- Tastaturabfrage 105
- TCP 592
- »Teil-Ganzes«-Beziehung 699
- tellg(), tellp() 438
- Template
 - Alias 266
 - für Funktionen 156
 - Instanziierung von T. 293
 - explizite 296
 - ökonomische (bei vielen Dateien) 295
 - int-Parameter 293
 - für Klassen 291
 - Konstante 191
 - Metaprogrammierung 494
 - Method (Design-Muster) 324, 410
 - Spezialisierung 158
 - partielle 882
 - variable Parameterzahl 497
- temporäres Objekt (Vermeidung) 183
- Terminal 30
- terminate() 351
- terminate_handler 352
- test() (Bitset) 830
- Test Driven Development 657
- Test-Suite 658
- Textersetzung 137
- this 240
- this->, *this bei Zugriff auf Oberklassenelement 373
- this_thread 536
- thousands_sep() 913, 914
- Thread 531
- thread (Klasse) 534
- ThreadGroup 542
- Thread-Sicherheit 563
- throw 346
- tiefe Kopie 252
- time 910, 917
- time() 376, 950
- time_get 917
- time_put 918
- time_t 376, 949
- tm 376, 949
- to_array() 853
- tolower() 708, 910, 912, 944
- top() 858, 861
- to_string()
 - bitset 830
 - Zahlkonvertierung 932
- to_ulong() (Bitset) 829
- toupper() 708, 907, 910, 912, 944
- trailing return type 281
- traits 881
- traits::eof() 416
- transform() 791, 911
- transform() (Ranges) 460
- transform_exclusive_scan() 728
- transform_inclusive_scan() 728
- tree (Programm) 642, 812
- Trennung von Schnittstellen und Implementation 142
- true 58
- trunename() 913
- true_type 506
- trunc 437
- try 346
- try_emplace() 876
- Tupel, <tuple> 825
- Typ 977
 - polymorpher bzw. dynamischer Typ 339
- type cast *siehe* Typumwandlung
- typedef 265
- typeid() 338, 412
- type_info 338
- <typeinfo> 351
- typename (bei Template-Parametern) 156
- Type Traits 505
- Typinformation 320
 - zur Laufzeit 338
- Typumwandlung
 - cast 56, 216, 260, 335
 - durch Compiler 196
 - const_cast<>() 337
 - dynamic_cast<>() 336
 - mit explicit 186
 - implizite 61, 186
 - mit Informationsverlust 126
 - skonstruktor 184, 196

- soperator 378
 - ios 436
 - reinterpret_cast<>() 338
 - Standard- 60
 - Zeiger 262
 - static_cast<>() 335
- U**
- u8 906
 - u8string, u16string, u32string 400
 - UCS 906
 - UDP 592, 600
 - Überladen
 - von Funktionen 125
 - von Operatoren *siehe* operator
 - Überlauf 44, 51
 - Überschreiben
 - von Funktionen 310
 - Übersetzung 142
 - seinheit 144
 - uintX_t, uint_fastX_t, uint_leastX_t
(X = 8, 16, 32, 64), uintmax_t 48
 - Umgebungsvariable 239
 - UML 693
 - Umleitung der Ausgabe auf Strings 439
 - unärer Operator 359
 - unäres Prädikat 737
 - UND
 - bitweises 46, 47
 - logisches 58
 - #undef 136
 - undefined behaviour 234
 - underflow 51
 - underflow_error 351
 - unsigned_integral 518
 - Unicode 413, 905
 - uniform_int_distribution 782
 - uniform_real_distribution 783
 - union 102
 - unique() Sequenzen 855
 - unique(), -_copy() Algorithmen 733
 - unique_lock 545
 - unique_ptr 267, 385, 684, 935
 - für Arrays 687
 - Unit-Test 655
 - unitbuf 418, 425, 428
 - unordered_map 875
 - unordered_multimap 878
 - unordered_multiset 878
 - unordered_set 878
 - unsigned 42
 - unsigned char 54
 - Unterklasse 298, 977
 - upper_bound() 762, 865
 - uppercase 425, 428
 - URI, URL 592, 713
 - URL-Codierung 605
 - use case 205
 - use_facet() 708, 710, 904
 - using
 - enum 87
 - Deklaration 305, 341
 - Namespace
 - Deklaration 148
 - Direktive 148
 - statt typedef 265
 - UTC 950
 - UTF 400, 977
 - UTF-8 905
 - <utility> 284, 476, 790, 822, 825
- V**
- valarray 249
 - valgrind 399
 - value(), value_or() (optional) 839
 - value_comp(), value_compare 865
 - value_type 867
 - value_type 843, 863
 - Variable 34
 - automatische 146
 - globale 145, 146
 - make 636
 - Name 41
 - variadic templates 497
 - variant 831
 - vector 849
 - at() 91
 - push_back() 94
 - size() 90
 - vector<bool> 850
 - <vector> 819, 842, 849
 - Vektor 89

- Klasse 363
- Länge (geom.) 726
- Verbundanweisung 66
- verdecken (Methode) 300, 305
- Vereinigung (Menge) 765
- Vererbung 297, 977
 - der abstrakt-Eigenschaft 316
 - von constraints 326
 - der Implementierung 341
 - Mehrfach- 328
 - private 340
 - protected 342
 - von Zugriffsrechten 304
 - und Zuweisungsoperator 409
- Vergleich
 - von double-Werten 678
 - bei Vererbung 410
- Verschiebung (*move*) 472
- verschmelzen (*merge*) 752
- Vertrag 326, 978
- verwitwetes Objekt 233
- Verzeichnis
 - anlegen 809
 - anzeigen 811
 - kopieren 808
 - löschen 806
 - umbenennen 810
- Verzeichnisbaum
 - anzeigen 812
 - make 643
- Verzweigung 66
- View 460
- views 460
- virtual 311, 313, 321
- virtuelle Basisklasse 332
- virtueller Destruktor 320
- virtuelle Funktionen 311, 314
 - private 324
 - rein- 315
- void 216
 - als Funktionstyp 115
- void* 260
 - Typumwandlung nach 216
- volatile 978
- vollständiges Objekt 333, 334, 978
- Vorbedingung 133, 978

- vorgegebene Parameterwerte
 - in Funktionen 124
 - in Konstruktoren 178
- Vorkommastellen 49
- Vorrangregeln 59, 955
- Vorwärtsdeklaration 210

W

- Wächter (Tabellenende) 92, 221
- Wahrheitswert 58
 - zufällig erzeugen 785
- wait() 549
 - mit Lambda-Funktion 554
- Warteschlange 859
- Wartung 111
- wchar_t 55, 906, 946
- weak_ordering 390
- weak_ptr 940
- Webserver 611
- Weite der Ausgabe 421, 424
- weiterleitende Referenz 479
- Wert
 - eines Attributs 970
 - Parameterübergabe per 119
- Wertebereich ganzer Zahlen 43
- Wertsemantik 253, 449, 465
 - Performanceproblem 467
- what() 350
- while 75, 226, 228
- white space *siehe* Zwischenraum-
zeichen
- wide character 55
- widen() 912
- Widget 575
- width() 424
- Wiederverwendung
 - durch Delegation 342
- wildcard 639
- Winterzeit 949
- Wochentag 949
- wofstream 909
- Wrapperklasse für Iterator 889
- write() 255, 418
- ws 430
- wstring 400, 909, 923

X

XOR, bitweises 46
xvalue 469

Y

yield() 536

Z

Zahl in String umwandeln 706
Zahlenbereich 42, 50
Zeichen 54
Zeichenkette 33, *siehe auch* String
 C-String 223
 Kopieren einer 227
Zeichenklasse (Regex) 522
Zeichenliteral 906
Zeichensatz 904
Zeiger 213, 230
 Arithmetik 220
 vs. Array 219
 auf Basisklasse 314, 320
 Darstellung
 von [] 222
 von [][] 246
 auf Elementdaten 264
 auf Elementfunktionen 263
 auf Funktionen 258
 hängender 233
 intelligente *siehe* Smart Pointer
 Null-Zeiger 216
 auf Oberklasse 306, 314
 auf Objekt (Mehrfachvererbung) 331
 auf lokale Objekte 217
 Parameterübergabe per Z. 235
Zeile
 einlesen *siehe* getline()
 neue 56
Zeit-Server 603
Zeitkomplexität 978
Ziel (make) 634
Ziffernzeichen 54
Zufallszahlen 781
Zugriffsspezifizierer und -rechte 303
zusammengesetzte Datentypen 84
Zusicherung 139
Zustand 978

eines Iterators 454
Zuweisung 33, 65, 70, 978
 und Initialisierung 182
 und Vererbung 306, 409
Zuweisungsoperator 182, 369, 483
 implizite Deklaration 369, 409
 und Vererbung 409
zweidimensionale Matrix 403
Zweierkomplement 42
Zwischenraumzeichen 104, 224, 415