

Spis treści

Przedmowa	23
Przedmowa do wydania trzeciego	27
Przedmowa do wydania drugiego	29
Przedmowa do wydania pierwszego	31
CZĘŚĆ I. WPROWADZENIE	33
Rozdział 1. Uwagi do czytelnika	35
1.1. Struktura książki	35
1.1.1. Wprowadzenie	36
1.1.2. Podstawowe narzędzia	36
1.1.3. Techniki abstrakcji	37
1.1.4. Biblioteka standardowa	39
1.1.5. Przykłady i odwołania	40
1.2. Projekt języka C++	41
1.2.1. Styl programowania	43
1.2.2. Kontrola typów	46
1.2.3. Zgodność z językiem C	47
1.2.4. Język, biblioteki i systemy	48
1.3. Nauka języka C++	50
1.3.1. Programowanie w języku C++	52
1.3.2. Rady dla programistów C++	53
1.3.3. Rady dla programistów C	53
1.3.4. Rady dla programistów języka Java	54
1.4. Historia	55
1.4.1. Oś czasu	56
1.4.2. Pierwsze lata	57

1.4.3. Standard z 1998 r.	59
1.4.4. Standard z 2011 r.	62
1.4.5. Do czego jest używany język C++	65
1.5. Rady	67
1.6. Literatura	68

Rozdział 2. Kurs języka C ++ . Podstawy **73**

2.1. Wprowadzenie	73
2.2. Podstawy	74
2.2.1. Witaj, świecie!	75
2.2.2. Typy, zmienne i arytmetyka	76
2.2.3. Stałe	78
2.2.4. Testy i pętle	79
2.2.5. Wskaźniki, tablice i pętle	80
2.3. Typy zdefiniowane przez użytkownika	82
2.3.1. Struktury	83
2.3.2. Klasy	84
2.3.3. Wyliczenia	86
2.4. Modułowość	87
2.4.1. Osobna kompilacja	88
2.4.2. Przestrzenie nazw	89
2.4.3. Obsługa błędów	90
2.5. Pośłowie	93
2.6. Rady	93

Rozdział 3. Kurs języka C++. Techniki abstrakcji **95**

3.1. Wprowadzenie	95
3.2. Klasy	96
3.2.1. Typy konkretne	96
3.2.2. Typy abstrakcyjne	101
3.2.3. Funkcje wirtualne	103
3.2.4. Hierarchie klas	104
3.3. Kopiowanie i przenoszenie	108
3.3.1. Kopiowanie kontenerów	108
3.3.2. Przenoszenie kontenerów	110
3.3.3. Zarządzanie zasobami	112
3.3.4. Tłumienie operacji	113

3.4. Szablony	113
3.4.1. Typy parametryzowane	114
3.4.2. Szablony funkcji	115
3.4.3. Obiekty funkcyjne	116
3.4.4. Zmienne szablony	118
3.4.5. Aliasy	119
3.5. Rady	120

Rozdział 4. Kurs języka C++ . Kontenery i algorytmy **121**

4.1. Biblioteki	121
4.1.1. Przegląd biblioteki standardowej	122
4.1.2. Nagłówki i przestrzeń nazw biblioteki standardowej	123
4.2. Łańcuchy	124
4.3. Strumienie wejścia i wyjścia	126
4.3.1. Wyjście	126
4.3.2. Wejście	127
4.3.3. Wejście i wyjście typów zdefiniowanych przez użytkownika	128
4.4. Kontenery	129
4.4.1. vector	130
4.4.2. list	133
4.4.3. map	134
4.4.4. unordered_map	135
4.4.5. Przegląd kontenerów	135
4.5. Algorytmy	137
4.5.1. Używanie iteratorów	138
4.5.2. Typy iteratorów	140
4.5.3. Iteratory strumieni	140
4.5.4. Predykaty	142
4.5.5. Przegląd algorytmów	143
4.5.6. Algorytmy kontenerowe	143
4.6. Rady	144

Rozdział 5. Kurs języka C ++. Współbieżność i narzędzia **145**

5.1. Wprowadzenie	145
5.2. Zarządzanie zasobami	146
5.2.1. unique_ptr i shared_ptr	146
5.3. Współbieżność	148

5.3.1. Zadania i wątki	149
5.3.2. Przekazywanie argumentów	150
5.3.3. Zwracanie wyników	150
5.3.4. Wspólne używanie danych	151
5.3.5. Komunikacja między zadaniami	154
5.4. Drobne, ale przydatne narzędzia	157
5.4.1. Czas	157
5.4.2. Funkcje typowe	158
5.4.3. pair i tuple	160
5.5. Wyrażenia regularne	161
5.6. Matematyka	162
5.6.1. Funkcje i algorytmy matematyczne	162
5.6.2. Liczby zespolone	163
5.6.3. Liczby losowe	163
5.6.4. Arytmetyka wektorów	165
5.6.5. Limity liczbowe	165
5.7. Rady	166

CZĘŚĆ II. PODSTAWOWE NARZĘDZIA **167**

Rozdział 6. Typy i deklaracje	169
6.1. Standard ISO języka C++	169
6.1.1. Implementacje	171
6.1.2. Podstawowy źródłowy zestaw znaków	171
6.2. Typy	172
6.2.1. Typy podstawowe	172
6.2.2. Typ logiczny	173
6.2.3. Typy znakowe	174
6.2.4. Typy całkowitoliczbowe	179
6.2.5. Typy zmiennoprzecinkowe	181
6.2.6. Przedrostki i przyrostki	182
6.2.7. void	183
6.2.8. Rozmiary	183
6.2.9. Wyrównanie	185
6.3. Deklaracje	186
6.3.1. Struktura deklaracji	188
6.3.2. Deklarowanie po kilka nazw	189

6.3.3. Nazwy	189
6.3.4. Zakres dostępności	191
6.3.5. Inicjacja	194
6.3.6. Dedukowanie typu: auto i decltype()	197
6.4. Obiekty i wartości	200
6.4.1. Wartości lewo- i prawostronne	200
6.4.2. Cykl istnienia obiektów	201
6.5. Aliasy typów	202
6.6. Rady	203

Rozdział 7. Wskaźniki, tablice i referencje **205**

7.1. Wprowadzenie	205
7.2. Wskaźniki	205
7.2.1. void*	206
7.2.2. nullptr	207
7.3. Tablice	208
7.3.1. Inicjatory tablic	209
7.3.2. Literały łańcuchowe	210
7.4. Wskaźniki do tablic	213
7.4.1. Przeglądanie tablic	214
7.4.2. Tablice wielowymiarowe	217
7.4.3. Przekazywanie tablic	217
7.5. Wskaźniki i const	220
7.6. Wskaźniki i własność	221
7.7. Referencje	222
7.7.1. Referencje lewostronne	224
7.7.2. Referencje prawostronne	227
7.7.3. Referencje do referencji	229
7.7.4. Wskaźniki i referencje	230
7.8. Rady	232

Rozdział 8. Struktury, unie i wyliczenia **233**

8.1. Wprowadzenie	233
8.2. Struktury	234
8.2.1. Układ struktur	235
8.2.2. Nazwy struktur	236
8.2.3. Struktury a klasy	237

8.2.4. Struktury a tablice	239
8.2.5. Ekwiwalencja typów	241
8.2.6. Stare zwykłe dane	241
8.2.7. Pola	244
8.3. Unie	244
8.3.1. Unie a klasy	246
8.3.2. Anonimowe unie	247
8.4. Wyliczenia	249
8.4.1. Klasy wyliczeniowe	250
8.4.2. Zwykłe wyliczenia	253
8.4.3. Wyliczenia anonimowe	254
8.5. Rady	255
Rozdział 9. Instrukcje	257
9.1. Wprowadzenie	257
9.2. Zestawienie instrukcji	258
9.3. Deklaracje jako instrukcje	259
9.4. Instrukcje wyboru	260
9.4.1. Instrukcje if	260
9.4.2. Instrukcje switch	261
9.4.3. Deklaracje w warunkach	264
9.5. Instrukcje iteracyjne	264
9.5.1. Zakresowe instrukcje for	265
9.5.2. Instrukcje for	266
9.5.3. Instrukcje while	267
9.5.4. Instrukcje do	267
9.5.5. Kończenie pętli	268
9.6. Instrukcje goto	269
9.7. Komentarze i wcięcia	269
9.8. Rady	271
Rozdział 10. Wyrażenia	273
10.1. Wprowadzenie	273
10.2. Kalkulator	273
10.2.1. Parser	274
10.2.2. Wejście	278
10.2.3. Wejście niskopoziomowe	282

10.2.4. Obsługa błędów	283
10.2.5. Sterownik	284
10.2.6. Nagłówki	284
10.2.7. Argumenty wiersza poleceń	285
10.2.8. Uwaga na temat stylu	286
10.3. Zestawienie operatorów	287
10.3.1. Wyniki	291
10.3.2. Kolejność wykonywania działań	292
10.3.3. Priorytety operatorów	292
10.3.4. Obiekty tymczasowe	293
10.4. Wyrażenia stałe	295
10.4.1. Stałe symboliczne	297
10.4.2. const w wyrażeniach stałych	297
10.4.3. Typy literałów	297
10.4.4. Argumenty referencyjne	298
10.4.5. Wyrażenia stałe adresowe	299
10.5. Niejawna konwersja typów	299
10.5.1. Promocje	300
10.5.2. Konwersje	300
10.5.3. Typowe konwersje arytmetyczne	303
10.6. Rady	304
Rozdział 11. Operacje wyboru	305
11.1. Różne operatory	305
11.1.1. Operatory logiczne	305
11.1.2. Bitowe operatory logiczne	306
11.1.3. Wyrażenia warunkowe	307
11.1.4. Inkrementacja i dekrementacja	307
11.2. Pamięć wolna	309
11.2.1. Zarządzanie pamięcią	311
11.2.2. Tablice	313
11.2.3. Sprawdzanie dostępności miejsca w pamięci	314
11.2.4. Przeciążanie operatora new	315
11.3. Listy	318
11.3.1. Model implementacji	318
11.3.2. Listy kwalifikowane	319
11.3.3. Listy niekwalifikowane	320

11.4. Wyrażenia lambda	322
11.4.1. Model implementacji	322
11.4.2. Alternatywy dla lambda	323
11.4.3. Lista zmiennych	325
11.4.4. Wywoływanie i zwracanie wartości	329
11.4.5. Typ lambda	329
11.5. Jawną konwersja typów	330
11.5.1. Konstrukcja	331
11.5.2. Rzutowania nazwane	333
11.5.3. Rzutowanie w stylu języka C	334
11.5.4. Rzutowanie w stylu funkcyjnym	334
11.6. Rady	335
Rozdział 12. Funkcje	337
12.1. Deklarowanie funkcji	337
12.1.1. Dlaczego funkcje	338
12.1.2. Składniki deklaracji funkcji	338
12.1.3. Definiowanie funkcji	339
12.1.4. Zwracanie wartości	340
12.1.5. Funkcje inline	342
12.1.6. Funkcje constexpr	343
12.1.7. Funkcje [[noreturn]]	346
12.1.8. Zmienne lokalne	346
12.2. Przekazywanie argumentów	347
12.2.1. Argumenty referencyjne	348
12.2.2. Argumenty tablicowe	350
12.2.3. Argumenty listowe	351
12.2.4. Nieokreślona liczba argumentów	353
12.2.5. Argumenty domyślne	356
12.3. Przeciążanie funkcji	358
12.3.1. Automatyczne wybieranie przeciążonych funkcji	358
12.3.2. Przeciążanie a typ zwrotny	360
12.3.3. Przeciążanie a zakres	360
12.3.4. Wybieranie przeciążonych funkcji z wieloma argumentami	361
12.3.5. Ręczne wybieranie przeciążonej funkcji	362
12.4. Warunki wstępne i końcowe	362
12.5. Wskaźnik do funkcji	364

12.6. Makra	368
12.6.1. Kompilacja warunkowa	370
12.6.2. Makra predefiniowane	371
12.6.3. Pragmy	372
12.7. Rady	372

Rozdział 13. Obsługa wyjątków **373**

13.1. Obsługa błędów	373
13.1.1. Wyjątki	374
13.1.2. Tradycyjna obsługa błędów	376
13.1.3. Niedbała obsługa błędów	377
13.1.4. Alternatywne spojrzenie na wyjątki	378
13.1.5. Kiedy nie można używać wyjątków	379
13.1.6. Hierarchiczna obsługa błędów	380
13.1.7. Wyjątki a wydajność	381
13.2. Gwarancje wyjątków	383
13.3. Zarządzanie zasobami	385
13.3.1. Finalizacja	388
13.4. Egzekwowanie przestrzegania niezmienników	389
13.5. Zgłaszanie i przechwytywanie wyjątków	394
13.5.1. Zgłaszanie wyjątków	394
13.5.2. Przechwytywanie wyjątków	397
13.5.3. Wyjątki a wątki	404
13.6. Implementacja wektora	405
13.6.1. Prosty wektor	405
13.6.2. Jawna reprezentacja pamięci	409
13.6.3. Przypisywanie	411
13.6.4. Zmienianie rozmiaru	413
13.7. Rady	416

Rozdział 14. Przestrzenie nazw **419**

14.1. Kwestie dotyczące kompozycji	419
14.2. Przestrzenie nazw	420
14.2.1. Bezpośrednia kwalifikacja	422
14.2.2. Deklaracje using	423
14.2.3. Dyrektywy using	424
14.2.4. Wyszukiwanie wg argumentów	425

14.2.5. Przestrzenie nazw są otwarte	427
14.3. Modularyzacja i interfejsy	428
14.3.1. Przestrzenie nazw i moduły	430
14.3.2. Implementacje	431
14.3.3. Interfejsy i implementacje	433
14.4. Składanie przy użyciu przestrzeni nazw	435
14.4.1. Wygoda a bezpieczeństwo	435
14.4.2. Aliasy przestrzeni nazw	436
14.4.3. Składanie przestrzeni nazw	436
14.4.4. Składanie i wybieranie	438
14.4.5. Przestrzenie nazw a przeciążanie	439
14.4.6. Wersjonowanie	441
14.4.7. Zagnieżdżanie przestrzeni nazw	443
14.4.8. Anonimowe przestrzenie nazw	444
14.4.9. Nagłówki języka C	444
14.5. Rady	445
Rozdział 15. Pliki źródłowe i programy	447
15.1. Rozdzielna kompilacja	447
15.2. Konsolidacja	448
15.2.1. Nazwy lokalne w plikach	451
15.2.2. Pliki nagłówkowe	451
15.2.3. Reguła jednej definicji	453
15.2.4. Nagłówki z biblioteki standardowej	455
15.2.5. Konsolidacja z kodem w innym języku	456
15.2.6. Konsolidacja a wskaźniki do funkcji	458
15.3. Używanie plików nagłówkowych	459
15.3.1. Organizacja z jednym nagłówkiem	459
15.3.2. Organizacja z wieloma nagłówkami	463
15.3.3. Strażnicy dołączania	467
15.4. Programy	468
15.4.1. Inicjacja zmiennych nielokalnych	469
15.4.2. Inicjacja i współbieżność	470
15.4.3. Zamykanie programu	470
15.5. Rady	472
CZĘŚĆ III. TECHNIKI ABSTRAKCJI	473

Rozdział 16. Klasy	475
16.1. Wprowadzenie	475
16.2. Podstawowe wiadomości o klasach	476
16.2.1. Funkcje składowe	477
16.2.2. Kopiowanie domyślne	478
16.2.3. Kontrola dostępu	479
16.2.4. Klasy i struktury	480
16.2.5. Konstruktory	481
16.2.6. Konstruktory explicit	483
16.2.7. Inicjatory wewnątrzklasowe	485
16.2.8. Wewnątrzklasowe definicje funkcji	486
16.2.9. Zmienność	487
16.2.10. Słowo kluczowe this	490
16.2.11. Dostęp do składowych	491
16.2.12. Składowe statyczne	492
16.2.13. Typy składowe	494
16.3. Klasy konkretne	495
16.3.1. Funkcje składowe	498
16.3.2. Funkcje pomocnicze	500
16.3.3. Przeciążanie operatorów	502
16.3.4. Znaczenie klas konkretnych	503
16.4. Rady	504
 Rozdział 17. Tworzenie, kasowanie, kopiowanie i przenoszenie	 505
17.1. Wprowadzenie	505
17.2. Konstruktory i destruktory	507
17.2.1. Konstruktory i niezmienniki	508
17.2.2. Destruktry i zasoby	509
17.2.3. Destruktry klas bazowych i składowych klas	510
17.2.4. Wywoływanie konstruktorów i destruktorów	511
17.2.5. Destruktry wirtualne	512
17.3. Inicjacja obiektów klas	513
17.3.1. Inicjacja bez konstruktorów	513
17.3.2. Inicjacja przy użyciu konstruktorów	515
17.3.3. Konstruktory domyślne	517
17.3.4. Konstruktory z listą inicjacyjną	519

17.4. Inicjacja składowych i bazy	524
17.4.1. Inicjacja składowych	524
17.4.2. Inicjatory bazy	525
17.4.3. Delegowanie konstruktorów	526
17.4.4. Inicjatory wewnątrzklasowe	527
17.4.5. Inicjacja składowych statycznych	529
17.5. Kopiowanie i przenoszenie	530
17.5.1. Kopiowanie	530
17.5.2. Przenoszenie	537
17.6. Generowanie domyślnych operacji	541
17.6.1. Jawne operacje domyślne	541
17.6.2. Operacje domyślne	542
17.6.3. Używanie operacji domyślnych	543
17.6.4. Usuwanie funkcji	547
17.7. Rady	548
Rozdział 18. Przeciążanie operatorów	551
18.1. Wprowadzenie	551
18.2. Funkcje operatorowe	553
18.2.1. Operatory dwu- i jednoargumentowe	554
18.2.2. Predefiniowane znaczenie operatorów	555
18.2.3. Operatory i typy zdefiniowane przez użytkownika	555
18.2.4. Przekazywanie obiektów	556
18.2.5. Operatory w przestrzeniach nazw	557
18.3. Typ reprezentujący liczby zespolone	559
18.3.1. Operatory składowe i zewnętrzne	559
18.3.2. Arytmetyka mieszana	560
18.3.3. Konwersje	561
18.3.4. Literały	564
18.3.5. Funkcje dostępne	565
18.3.6. Funkcje pomocnicze	565
18.4. Konwersja typów	567
18.4.1. Operatory konwersji	567
18.4.2. Operatory konwersji explicit	569
18.4.3. Niejednoznaczności	569
18.5. Rady	571

Rozdział 19. Operatory specjalne	573
19.1. Wprowadzenie	573
19.2. Operatory specjalne	573
19.2.1. Indeksowanie	573
19.2.2. Wywoływanie funkcji	574
19.2.3. Dereferencja	576
19.2.4. Inkrementacja i dekrementacja	578
19.2.5. Alokacja i dezalokacja	580
19.2.6. Literały zdefiniowane przez użytkownika	581
19.3. Klasa String	584
19.3.1. Podstawowe operacje	585
19.3.2. Dostęp do znaków	585
19.3.3. Reprezentacja	586
19.3.4. Funkcje składowe	589
19.3.5. Funkcje pomocnicze	591
19.3.6. Sposoby użycia	593
19.4. Przyjaciele	594
19.4.1. Znajdowanie przyjaciół	596
19.4.2. Przyjaciele i składowe	597
19.5. Rady	598
 Rozdział 20. Derywacja klas	 599
20.1. Wprowadzenie	599
20.2. Klasy pochodne	600
20.2.1. Funkcje składowe	602
20.2.2. Konstruktory i destruktory	604
20.3. Hierarchie klas	604
20.3.1. Pola typów	605
20.3.2. Funkcje wirtualne	607
20.3.3. Bezpośrednia kwalifikacja	610
20.3.4. Kontrola przesłaniania	610
20.3.5. Używanie składowych klasy bazowej	614
20.3.6. Rozluźnienie zasady dotyczącej typów zwrotnych	617
20.4. Klasy abstrakcyjne	619
20.5. Kontrola dostępu	621
20.5.1. Składowe chronione	624
20.5.2. Dostęp do klas bazowych	625

20.5.3. Deklaracje using i kontrola dostępu	627
20.6. Wskaźniki do składowych	627
20.6.1. Wskaźniki do funkcji składowych	628
20.6.2. Wskaźniki do danych składowych	630
20.6.3. Składowe bazy i klasy pochodnej	631
20.7. Rady	631
Rozdział 21. Hierarchie klas	633
21.1. Wprowadzenie	633
21.2. Projektowanie hierarchii klas	633
21.2.1. Dziedziczenie implementacji	634
21.2.2. Dziedziczenie interfejsu	637
21.2.3. Alternatywne implementacje	639
21.2.4. Lokalizowanie tworzenia obiektu	642
21.3. Wielodziedziczenie	644
21.3.1. Wiele interfejsów	644
21.3.2. Wiele klas implementacyjnych	644
21.3.3. Rozstrzyganie niejednoznaczności	646
21.3.4. Wielokrotne użycie klasy bazowej	649
21.3.5. Wirtualne klasy bazowe	651
21.3.6. Bazy wirtualne a replikowane	655
21.4. Rady	658
Rozdział 22. Informacje o typach w czasie działania programu	659
22.1. Wprowadzenie	659
22.2. Poruszanie się w obrębie hierarchii klas	660
22.2.1. Rzutowanie dynamiczne	661
22.2.2. Wielodziedziczenie	664
22.2.3. Rzutowanie statyczne i dynamiczne	665
22.2.4. Odzyskiwanie interfejsu	667
22.3. Podwójny polimorfizm i wizytatorzy	670
22.3.1. Podwójny polimorfizm	671
22.3.2. Wizytatorzy	673
22.4. Konstrukcja i destrukcja	675
22.5. Identyfikacja typów	675
22.5.1. Rozszerzone informacje o typie	677
22.6. Poprawne i niepoprawne używanie RTTI	678

22.7. Rady	680
------------	-----

Rozdział 23. Szablony	681
------------------------------	------------

23.1. Wprowadzenie i przegląd	681
23.2. Prosty szablon łańcucha	684
23.2.1. Definiowanie szablonu	685
23.2.2. Konkretyzacja szablonu	687
23.3. Kontrola typów	688
23.3.1. Ekwiwalencja typów	689
23.3.2. Wykrywanie błędów	690
23.4. Składowe szablonu klasy	691
23.4.1. Dane składowe	691
23.4.2. Funkcje składowe	692
23.4.3. Aliasy typów składowych	692
23.4.4. Składowe statyczne	692
23.4.5. Typy składowe	693
23.4.6. Szablony składowe	694
23.4.7. Przyjaciele	698
23.5. Szablony funkcji	699
23.5.1. Argumenty szablonu funkcji	701
23.5.2. Dedukcja argumentów szablonu funkcji	702
23.5.3. Przeciążanie szablonów funkcji	704
23.6. Aliasy szablonów	708
23.7. Organizacja kodu źródłowego	709
23.7.1. Konsolidacja	711
23.8. Rady	712

Rozdział 24. Programowanie ogólne	713
--	------------

24.1. Wprowadzenie	713
24.2. Algorytmy i uogólnianie	714
24.3. Koncepcje	718
24.3.1. Odkrywanie koncepcji	718
24.3.2. Koncepcje i ograniczenia	722
24.4. Konkretyzacja koncepcji	724
24.4.1. Aksjomaty	727
24.4.2. Koncepcje wieloargumentowe	728
24.4.3. Koncepcje wartości	729

24.4.4. Sprawdzanie ograniczeń	730
24.4.5. Sprawdzanie definicji szablonu	731
24.5. Rady	733

Rozdział 25. Specjalizacja **735**

25.1. Wprowadzenie	735
25.2. Argumenty i parametry szablonu	736
25.2.1. Typy jako argumenty	736
25.2.2. Wartości jako argumenty	738
25.2.3. Operacje jako argumenty	739
25.2.4. Szablony jako argumenty	742
25.2.5. Domyślne argumenty szablonów	742
25.3. Specjalizacja	744
25.3.1. Specjalizacja interfejsu	747
25.3.2. Szablon podstawowy	748
25.3.3. Porządek specjalizacji	750
25.3.4. Specjalizacja szablonu funkcji	750
25.4. Rady	753

Rozdział 26. Konkretyzacja **755**

26.1. Wprowadzenie	755
26.2. Konkretyzacja szablonu	756
26.2.1. Kiedy konkretyzacja jest potrzebna	757
26.2.2. Ręczne sterowanie konkretyzacją	758
26.3. Wiązanie nazw	759
26.3.1. Nazwy zależne	761
26.3.2. Wiązanie w miejscu definicji	762
26.3.3. Wiązanie w miejscu konkretyzacji	763
26.3.4. Wiele miejsc konkretyzacji	766
26.3.5. Szablony i przestrzenie nazw	767
26.3.6. Nadmiernie agresywne wyszukiwanie wg argumentów	768
26.3.7. Nazwy z klas bazowych	770
26.4. Rady	772

Rozdział 27. Hierarchie szablonów **773**

27.1. Wprowadzenie	773
27.2. Parametryzacja i hierarchia	774

27.2.1. Typy generowane	776
27.2.2. Konwersje szablonów	778
27.3. Hierarchie szablonów klas	779
27.3.1. Szablony jako interfejsy	780
27.4. Parametry szablonowe jako klasy bazowe	781
27.4.1. Składanie struktur danych	781
27.4.2. Linearyzacja hierarchii klas	785
27.5. Rady	790

Rozdział 28. Metaprogramowanie **791**

28.1. Wprowadzenie	791
28.2. Funkcje typowe	794
28.2.1. Aliasy typów	796
28.2.2. Predykaty typów	798
28.2.3. Wybieranie funkcji	799
28.2.4. Cechy	800
28.3. Struktury sterujące	802
28.3.1. Wybieranie	802
28.3.2. Iteracja i rekurencja	805
28.3.3. Kiedy stosować metaprogramowanie	806
28.4. Definicja warunkowa	807
28.4.1. Używanie Enable_if	809
28.4.2. Implementacja Enable_if	811
28.4.3. Enable if i koncepcje	811
28.4.4. Dodatkowe przykłady użycia Enable_if	812
28.5. Lista czasu kompilacji	814
28.5.1. Prosta funkcja wyjściowa	816
28.5.2. Dostęp do elementów	818
28.5.3. make_tuple	820
28.6. Szablony zmienne	821
28.6.1. Bezpieczna typowo funkcja printf()	821
28.6.2. Szczegóły techniczne	824
28.6.3. Przekazywanie	825
28.6.4. Typ tupie z biblioteki standardowej	827
28.7. Przykład z jednostkami układu SI	830
28.7.1. Jednostki	830
28.7.2. Wielkości	831

28.7.3. Literały jednostkowe	833
28.7.4. Funkcje pomocnicze	834
28.8. Rady	836

Rozdział 29. Projekt macierzy **837**

29.1. Wprowadzenie	837
29.1.1. Podstawowe zastosowania macierzy	838
29.1.2. Wymagania dotyczące macierzy	840
29.2. Szablon macierzy	841
29.2.1. Konstrukcja i przypisywanie	842
29.2.2. Indeksowanie i cięcie	843
29.3. Operacje arytmetyczne na macierzach	845
29.3.1. Operacje skalarne	846
29.3.2. Dodawanie	847
29.3.3. Mnożenie	848
29.4. Implementacja macierzy	850
29.4.1. Wycinki	850
29.4.2. Wycinki macierzy	850
29.4.3. Matrix_ref	852
29.4.4. Inicjacja listy macierzy	853
29.4.5. Dostęp do macierzy	855
29.4.6. Macierz zerowymiarowa	857
29.5. Rozwiązywanie równań liniowych	858
29.5.1. Klasyczna eliminacja Gaussa	859
29.5.2. Znajdowanie elementu centralnego	860
29.5.3. Testowanie	861
29.5.4. Połączone operacje	862
29.6. Rady	864

CZĘŚĆ IV. BIBLIOTEKA STANDARDOWA **865**

Rozdział 30. Przegląd zawartości biblioteki standardowej **867**

30.1. Wprowadzenie	867
30.1.1. Narzędzia biblioteki standardowej	868
30.1.2. Kryteria projektowe	869
30.1.3. Styl opisu	870
30.2. Nagłówki	871

30.3. Wsparcie dla języka	875
30.3.1. Wsparcie dla list inicjacyjnych	876
30.3.2. Wsparcie dla zakresowej pętli for	876
30.4. Obsługa błędów	877
30.4.1. Wyjątki	877
30.4.2. Asercje	882
30.4.3. system_error	882
30.5. Rady	892

Rozdział 31. Kontenery STL **893**

31.1. Wprowadzenie	893
31.2. Przegląd kontenerów	893
31.2.1. Reprezentacja kontenera	896
31.2.2. Wymagania dotyczące elementów	898
31.3. Przegląd operacji	901
31.3.1. Typy składowe	904
31.3.2. Konstruktory, destruktory i przypisania	904
31.3.3. Rozmiar i pojemność	906
31.3.4. Iteratory	907
31.3.5. Dostęp do elementów	908
31.3.6. Operacje stosowe	908
31.3.7. Operacje listowe	909
31.3.8. Inne operacje	910
31.4. Kontenery	910
31.4.1. vector	911
31.4.2. Listy	915
31.4.3. Kontenery asocjacyjne	917
31.5. Adaptacje kontenerów	929
31.5.1. Stos	929
31.5.2. Kolejka	931
31.5.3. Kolejka priorytetowa	931
31.6. Rady	932

Rozdział 32. Algorytmy STL **935**

32.1. Wprowadzenie	935
32.2. Algorytmy	935
32.2.1. Sekwencje	936

32.3. Argumenty zasad	938
32.3.1. Złożoność	939
32.4. Algorytmy nie modyfikujące sekwencji	940
32.4.1. for_each()	940
32.4.2. Predykaty sekwencji	940
32.4.3. count()	940
32.4.4. find()	941
32.4.5. equal() i mismatch()	942
32.4.6. search()	942
32.5. Algorytmy modyfikujące sekwencje	943
32.5.1. copy()	944
32.5.2. unique()	945
32.5.3. remove() i replace()	946
32.5.4. rotate(), random_shuffle() oraz partition()	947
32.5.5. Permutacje	948
32.5.6. fill()	948
32.5.7. swap()	949
32.6. Sortowanie i wyszukiwanie	950
32.6.1. Wyszukiwanie binarne	952
32.6.2. merge()	954
32.6.3. Algorytmy działające na zbiorach	954
32.6.4. Sterty	955
32.6.5. lexicographical_compare()	956
32.7. Element minimalny i maksymalny	957
32.8. Rady	958

Rozdział 33. Iteratory STL	959
33.1. Wprowadzenie	959
33.1.1. Model iteratorów	959
33.1.2. Kategorie iteratorów	961
33.1.3. Cechy iteratorów	962
33.1.4. Operacje iteratorów	964
33.2. Adaptacje iteratorów	965
33.2.1. Iterator odwrotny	966
33.2.2. Iteratory wstawiające	968
33.2.3. Iteratory przenoszące	969
33.3. Zakresowe funkcje dostępne	970

33.4. Obiekty funkcyjne	971
33.5. Adaptacje funkcji	972
33.5.1. bind()	972
33.5.2. mem_fn()	974
33.5.3. function	974
33.6. Rady	976

Rozdział 34. Pamięć i zasoby **977**

34.1. Wprowadzenie	977
34.2. „Prawie kontenery”	977
34.2.1. array	978
34.2.2. bitset	981
34.2.3. vector<bool>	985
34.2.4. Krotki	986
34.3. Wskaźniki do zarządzania pamięcią	990
34.3.1. unique_ptr	990
34.3.2. shared_ptr	993
34.3.3. weak_ptr	996
34.4. Alokatory	998
34.4.1. Alokator domyślny	1000
34.4.2. Cechy alokatorów	1001
34.4.3. Cechy wskaźników	1002
34.4.4. Alokatory zakresowe	1003
34.5. Interfejs odśmiecacza	1004
34.6. Pamięć niezainicjowana	1007
34.6.1. Bufory tymczasowe	1007
34.6.2. raw_storage_iterator	1008
34.7. Rady	1009

Rozdział 35. Narzędzia pomocnicze **1011**

35.1. Wprowadzenie	1011
35.2. Czas	1011
35.2.1. duration	1012
35.2.2. time_point	1015
35.2.3. Zegary	1017
35.2.4. Cechy czasu	1018
35.3. Działania arytmetyczne na liczbach wymiernych w czasie kompilacji	1019

35.4. Funkcje typowe	1020
35.4.1. Cechy typów	1020
35.4.2. Generatory typów	1025
35.5. Drobne narzędzia	1030
35.5.1. move() i forward()	1030
35.5.2. swap()	1031
35.5.3. Operatory relacyjne	1031
35.5.4. Porównywanie i mieszanie type_info	1032
35.6. Rady	1033

Rozdział 36. Łącuchy **1035**

36.1. Wprowadzenie	1035
36.2. Klasyfikacja znaków	1035
36.2.1. Funkcje klasyfikacji	1035
36.2.2. Cechy znaków	1036
36.3. Łącuchy	1038
36.3.1. Typ string a łańcuchy w stylu C	1039
36.3.2. Konstruktory	1040
36.3.3. Operacje podstawowe	1042
36.3.4. Łącuchowe wejście i wyjście	1044
36.3.5. Konwersje numeryczne	1044
36.3.6. Operacje w stylu biblioteki STL	1046
36.3.7. Rodzina funkcji find	1048
36.3.8. Podłańcuchy	1049
36.4. Rady	1050

Rozdział 37. Wyrażenia regularne **1053**

37.1. Wyrażenia regularne	1053
37.1.1. Notacja wyrażeń regularnych	1054
37.2. regex	1059
37.2.1. Wyniki dopasowywania	1061
37.2.2. Formatowanie	1063
37.3. Funkcje wyrażeń regularnych	1064
37.3.1. regex_match()	1064
37.3.2. regex_search()	1066
37.3.3. regex_replace()	1067
37.4. Iteratory wyrażeń regularnych	1068

37.4.1. regex_iterator	1068
37.4.2. regex_token_iterator	1070
37.5. regex_traits	1072
37.6. Rady	1073

Rozdział 38. Strumienie wejścia i wyjścia **1075**

38.1. Wprowadzenie	1075
38.2. Hierarchia strumieni wejścia i wyjścia	1077
38.2.1. Strumienie plikowe	1078
38.2.2. Strumienie łańcuchowe	1079
38.3. Obsługa błędów	1081
38.4. Operacje wejścia i wyjścia	1082
38.4.1. Operacje wejściowe	1083
38.4.2. Operacje wyjściowe	1086
38.4.3. Manipulatory	1088
38.4.4. Stan strumienia	1089
38.4.5. Formatowanie	1094
38.5. Iteratory strumieniowe	1101
38.6. Buforowanie	1102
38.6.1. Strumienie wyjściowe i bufory	1105
38.6.2. Strumienie wejściowe i bufory	1106
38.6.3. Iteratory buforów	1107
38.7. Rady	1109

Rozdział 39. Lokalizacje **1111**

39.1. Różnice kulturowe	1111
39.2. Klasa locale	1114
39.2.1. Lokalizacje nazwane	1116
39.2.2. Porównywanie łańcuchów	1120
39.3. Klasa facet	1120
39.3.1. Dostęp do faset w lokalizacji	1121
39.3.2. Definiowanie prostej fasety	1122
39.3.3. Zastosowania lokalizacji i faset	1125
39.4. Standardowe fasety	1125
39.4.1. Porównywanie łańcuchów	1127
39.4.2. Formatowanie liczb	1131
39.4.3. Formatowanie kwot pieniężnych	1136

39.4.4. Formatowanie daty i godziny	1141
39.4.5. Klasyfikacja znaków	1144
39.4.6. Konwersja kodów znaków	1147
39.4.7. Wiadomości	1151
39.5. Interfejsy pomocnicze	1155
39.5.1. Klasyfikacja znaków	1155
39.5.2. Konwersje znaków	1156
39.5.3. Konwersje łańcuchów	1156
39.5.4. Buforowanie konwersji	1157
39.6. Rady	1158
Rozdział 40. Liczby	1159
40.1. Wprowadzenie	1159
40.2. Granice liczbowe	1160
40.2.1. Makra ograniczające	1162
40.3. Standardowe funkcje matematyczne	1163
40.4. Liczby zespolone	1164
40.5. Tablica numeryczna valarray	1166
40.5.1. Konstruktory i przypisania	1166
40.5.2. Indeksowanie	1168
40.5.3. Operacje	1169
40.5.4. Wycinki	1172
40.5.5. slice_array	1174
40.5.6. Uogólnione wycinki	1175
40.6. Uogólnione algorytmy numeryczne	1176
40.6.1. Algorytm accumulate()	1177
40.6.2. Algorytm inner_product()	1177
40.6.3. Algorytmy partial_sum() i adjacent_difference()	1178
40.6.4. Algorytm iota()	1179
40.7. Liczby losowe	1180
40.7.1. Mechanizmy	1182
40.7.2. Urządzenie losowe	1184
40.7.3. Rozkłady	1185
40.7.4. Losowanie liczb w stylu C	1189
40.8. Rady	1189
41. Współbieżność	1191

41.1. Wprowadzenie	1191
41.2. Model pamięci	1193
41.2.1. Lokalizacje pamięci	1194
41.2.2. Zmienianie kolejności instrukcji	1195
41.2.3. Porządek pamięci	1196
41.2.4. Wyścigi do danych	1197
41.3. Konstrukcje atomowe	1198
41.3.1. Typy atomowe	1201
41.3.2. Flagi i bariery	1205
41.4. Słowo kluczowe volatile	1207
41.5. Rady	1207
Rozdział 42. Wątki i zadania	1209
42.1. Wprowadzenie	1209
42.2. Wątki	1210
42.2.1. Tożsamość	1211
42.2.2. Konstrukcja	1212
42.2.3. Destrukcja	1213
42.2.4. Funkcja join()	1214
42.2.5. Funkcja detach()	1215
42.2.6. Przestrzeń nazw this_thread	1217
42.2.7. Likwidowanie wątku	1218
42.2.8. Dane lokalne wątku	1218
42.3. Unikanie wyścigów do danych	1220
42.3.1. Muteksy	1220
42.3.2. Wiele blokad	1228
42.3.3. Funkcja call_once()	1230
42.3.4. Zmienne warunkowe	1231
42.4. Współbieżność zadaniowa	1235
42.4.1. Typy future i promise	1236
42.4.2. Typ promise	1237
42.4.3. Typ packaged_task	1238
42.4.4. Typ future	1241
42.4.5. Typ shared_future	1244
42.4.6. Funkcja async()	1245
42.4.7. Przykład równoległej funkcji fmd()	1247
42.5. Rady	1251

Rozdział 43. Biblioteka standardowa C	1253
43.1. Wprowadzenie	1253
43.2. Pliki	1253
43.3. Rodzina printf()	1254
43.4. Łańcuchy w stylu C	1259
43.5. Pamięć	1260
43.6. Data i godzina	1261
43.7. Itd.	1264
43.8. Rady	1266
 Rozdział 44. Zgodność	 1267
44.1. Wprowadzenie	1267
44.2. Rozszerzenia C++11	1268
44.2.1. Narzędzia językowe	1268
44.2.2. Składniki biblioteki standardowej	1269
44.2.3. Elementy wycofywane	1270
44.2.4. Praca ze starszymi implementacjami C++	1271
44.3. Zgodność C i C++	1271
44.3.1. C i C++ to rodzeństwo	1271
44.3.2. „Ciche” różnice	1273
44.3.3. Kod C nie będący kodem C++	1274
44.3.4. Kod C++ nie będący kodem C	1277
44.4. Rady	1279
 Skorowidz	 1281