
Spis treści

<i>Przedmowa</i>	xv
Struktura książki	xvi
Zmiany wprowadzone do trzeciego wydania	xxi
Przyszłość języka C++	xxii
Podziękowania	xxii
Podziękowania do drugiego wydania	xxiii
Literatura	xxiv
 Część I Przegląd języka C++	1
<i>Rozdział 1 Zaczynamy</i>	4
1.1 Rozwiązywanie problemów	4
1.2 Program w C++	5
1.3 Dyrektywy preprocesora	13
1.4 Kilka uwag o komentarzach	17
1.5 Pierwszy rzut oka na wejście-wyjście	19
 <i>Rozdział 2 Poznajemy język C++</i>	23
2.1 Wbudowany typ tablicowy	23
2.2 Dynamiczne przydzielanie pamięci oraz wskaźniki	27
2.3 Projektowanie abstrakcyjnych typów danych	30
2.4 Projektowanie obiektowe	42
2.5 Projektowanie uogólnione	52
2.6 Projektowanie z uwzględnieniem sytuacji wyjątkowych	58
2.7 Tablica pod dowolną inną nazwą	63
2.8 Tablica standardowa jest wektorem	68
 Część II Język podstawowy	75
<i>Rozdział 3 Typy danych w C++</i>	77
3.1 Stałe	77
3.2 Zmienne	81
3.3 Typy wskaźnikowe	90
3.4 Napisy	95
3.5 Modyfikator const	104
3.6 Typy referencyjne	107
3.7 Typ logiczny	112
3.8 Typy wyliczeniowe	113

3.9	Typy tablicowe	116
3.10	Klasa <code>vector</code>	123
3.11	Typ liczb zespolonych <code>complex</code>	127
3.12	Definicje nazw typów	128
3.13	Obiekty ulotne	129
3.14	Typ <code>pair</code>	130
3.15	Klasy	131
<i>Rozdział 4</i>	<i>Wyrażenia</i>	142
4.1	Co to jest wyrażenie?	142
4.2	Operatory arytmetyczne	144
4.3	Operatory relacyjne oraz logiczne	147
4.4	Operatory przypisania	150
4.5	Operatory zwiększania i zmniejszania	155
4.6	Operacje na liczbach zespolonych	157
4.7	Operator warunkowy	160
4.8	Operator <code>sizeof</code>	161
4.9	Wyrażenia: <code>new</code> oraz <code>delete</code>	163
4.10	Operator przecinkowy	165
4.11	Operatory bitowe	165
4.12	Operacje na obiektach klasy <code>bitset</code>	169
4.13	Reguły pierwszeństwa	173
4.14	Przekształcenia typów	176
4.15	Przykład klasy <code>iStos</code>	185
<i>Rozdział 5</i>	<i>Instrukcje</i>	190
5.1	Instrukcje proste i złożone	190
5.2	Instrukcja deklaracji	192
5.3	Instrukcja warunkowa <code>if</code>	195
5.4	Instrukcja warunkowa <code>switch</code>	203
5.5	Instrukcja pętli <code>for</code>	210
5.6	Instrukcja <code>while</code>	214
5.7	Instrukcja <code>do while</code>	217
5.8	Instrukcja <code>break</code>	218
5.9	Instrukcja <code>continue</code>	220
5.10	Instrukcja <code>goto</code>	220
5.11	Przykład listy z jednokierunkowymi wiązaniami	222
<i>Rozdział 6</i>	<i>Kolekcje</i>	248
6.1	System zapytań dotyczących tekstów	249
6.2	Wektor czy lista?	253
6.3	Jak powiększa się wektor	255
6.4	Definiowanie kolekcji uporządkowanej	259
6.5	Iteratory	263

6.6	Operacje na elementach kolekcji uporządkowanych	268
6.7	Gromadzenie wierszy tekstu w pamięci	272
6.8	Znajdowanie fragmentów napisów	275
6.9	Obsługa znaków przestankowych	281
6.10	Różne odmiany tego samego napisu	284
6.11	Dodatkowe operacje na napisach	287
6.12	Tworzenie słownika dla tekstu	293
6.13	Tworzenie zbioru słów wykluczanych	303
6.14	Pełny program	307
6.15	Wielosłownik i wielozbiór	316
6.16	Stos	318
6.17	Kolejka i kolejka z priorytetami	321
6.18	Powrót do naszej klasy iStos	321
Część III Programowanie proceduralne		325
Rozdział 7	<i>Funkcje</i>	327
7.1	Przegląd	327
7.2	Prototyp funkcji	330
7.3	Przekazywanie argumentów funkcji	334
7.4	Wartość przekazywana przez funkcję	352
7.5	Rekurencja	357
7.6	Funkcje rozwijane w miejscu wywołania	359
7.7	Dyrektywa łączenia: extern "C"	361
7.8	Funkcja main(): obsługa opcji wiersza polecenia	363
7.9	Wskaźniki do funkcji	374
Rozdział 8	<i>Zasięg deklaracji i czas trwania obiektów</i>	385
8.1	Zasięg	385
8.2	Obiekty globalne i funkcje	391
8.3	Obiekty lokalne	398
8.4	Dynamiczne przydzielanie pamięci	402
8.5	Definicje przestrzeni nazw	416
8.6	Używanie składowych przestrzeni nazw	431
Rozdział 9	<i>Funkcje przeciążone</i>	440
9.1	Deklaracje funkcji przeciążonych	440
9.2	Trzy etapy rozstrzygania przeciążenia	453
9.3	Przekształcenia typów argumentów	455
9.4	Więcej o rozstrzyganiu przeciążenia funkcji	471
Rozdział 10	<i>Wzorce funkcji</i>	486
10.1	Definicje wzorców funkcji	486
10.2	Konkretyzowanie wzorca funkcji	494

10.3	Dedukowanie argumentów wzorca funkcji	497
10.4	Jawne argumenty wzorców funkcji	502
10.5	Modele kompilacji wzorców funkcji	505
10.6	Jawna specjalizacja wzorców funkcji	511
10.7	Przeciążanie wzorców funkcji	516
10.8	Rozstrzyganie przeciążenia i konkretyzowanie wzorców	519
10.9	Rozstrzyganie znaczenia nazw w definicjach wzorców funkcji ...	527
10.10	Przestrzenie nazw i wzorce funkcji	534
10.11	Przykład zastosowania wzorców funkcji	538
Rozdział 11	Obsługa sytuacji wyjątkowych	542
11.1	Zgłaszanie wyjątków	542
11.2	Blok try	546
11.3	Wychwytywanie wyjątków	550
11.4	Specyfikacje wyjątków	559
11.5	Wyjątki a projektowanie oprogramowania	563
Rozdział 12	Algorytmy uogólnione	566
12.1	Przegląd	566
12.2	Korzystanie z algorytmów uogólnionych	570
12.3	Obiekty funkcyjne	581
12.4	Iteratory	589
12.5	Algorytmy uogólnione	598
12.6	Kiedy nie należy używać algorytmów uogólnionych	602
Część IV	Programowanie w stylu obiektowym	607
Rozdział 13	Klasy	609
13.1	Definicja klasy	610
13.2	Obiekty klas	617
13.3	Metody	620
13.4	Niejawnie zdefiniowany wskaźnik this	632
13.5	Składowe statyczne	637
13.6	Wskaźnik do składowej	644
13.7	Oszczędność miejsca: unia	653
13.8	Oszczędność miejsca: pole bitowe	658
13.9	Zasięg klasy	660
13.10	Klasy zagnieżdżone	668
13.11	Klasy jako składowe przestrzeni nazw	679
13.12	Klasy lokalne	683
Rozdział 14	Inicjowanie, przypisywanie oraz niszczenie obiektów klasy ...	685
14.1	Inicjowanie klasy	685
14.2	Konstruktor klasy	687

14.3	Destruktor klasy	699
14.4	Tablice i wektory obiektów klasy	705
14.5	Lista inicjowania składowych	712
14.6	Inicjowanie składowa po składowej	719
14.7	Przypisywanie składowa po składowej	725
14.8	Rozważania dotyczące wydajności	728
Rozdział 15	<i>Operatory przeciążone i przekształcenia typów zdefiniowane przez użytkownika</i>	734
15.1	Przeciążanie operatorów	734
15.2	Operatory przeciążone zaprzyjaźnione z klasą	744
15.3	Operator przypisania =	748
15.4	Operator indeksowania []	750
15.5	Operator wywołania ()	751
15.6	Operator wyboru składowej ->	753
15.7	Operatory zwiększania ++ i zmniejszania --	755
15.8	Operatory new i delete	759
15.9	Przekształcenia typów zdefiniowane przez użytkownika	769
15.10	Wybieranie przekształcenia	778
15.11	Rozstrzygnięcie przeciążenia i metody klas	791
15.12	Rozstrzygnięcie przeciążenia i operatory	797
Rozdział 16	<i>Wzorce klas</i>	806
16.1	Definicja wzorca klasy	807
16.2	Konkretyzowanie wzorca klasy	815
16.3	Metody wzorców klas	824
16.4	Deklaracje zaprzyjaźnienia we wzorcach klas	827
16.5	Pola statyczne wzorców klas	833
16.6	Typy zagnieżdżone we wzorcach klas	835
16.7	Wzorce składowe	838
16.8	Wzorce klas a model kompilacji	842
16.9	Specjalizacje wzorców klas	849
16.10	Częściowa specjalizacja wzorców klas	853
16.11	Rozstrzygnięcie znaczenia nazw we wzorcach klas	855
16.12	Przestrzenie nazw a wzorce klas	858
16.13	Wzorzec klasy Tablica	860
Część V	Programowanie obiektowe	869
Rozdział 17	<i>Klasy pochodne i dziedziczenie</i>	871
17.1	Definiowanie hierarchii klas	874
17.2	Identyfikowanie składowych hierarchii	882
17.3	Dostęp do składowych klasy podstawowej	893
17.4	Budowa klasy podstawowej i pochodnej	900

17.5	Funkcje wirtualne klasy podstawowej i pochodnej	911
17.6	Inicjowanie i przypisywanie składowa po składowej	936
17.7	Klasa zarządzania zapytaniami: ZapytanieUżytkownika	941
17.8	Koniec wieńczy dzieło	950
<i>Rozdział 18</i>	<i>Wielodziedziczenie i dziedziczenie wirtualne</i>	<i>957</i>
18.1	Wprowadzenie	957
18.2	Wielodziedziczenie	962
18.3	Dziedziczenie publiczne, prywatne i chronione	969
18.4	Zasięg klasy wobec dziedziczenia	978
18.5	Dziedziczenie wirtualne	986
18.6	Przykład zastosowania wielodziedziczenia wirtualnego	999
<i>Rozdział 19</i>	<i>Zastosowania dziedziczenia w języku C++</i>	<i>1013</i>
19.1	Identyfikacja typu podczas wykonywania programu	1013
19.2	Sytuacje wyjątkowe a dziedziczenie	1025
19.3	Rozstrzyganie przeciążenia a dziedziczenie	1043
<i>Rozdział 20</i>	<i>Biblioteka wejścia-wyjścia</i>	<i>1054</i>
20.1	Wyjście	1058
20.2	Wejście	1063
20.3	Pozostałe operatory wejścia-wyjścia	1074
20.4	Przeciążanie operatora <<	1081
20.5	Przeciążanie operatora >>	1086
20.6	Pliki wejściowe i wyjściowe	1088
20.7	Stan strumienia	1098
20.8	Strumień wewnętrzny	1100
20.9	Stan dotyczący formatowania	1104
20.10	Biblioteka wejścia-wyjścia a przestrzeganie zgodności typów	1110
<i>Dodatek</i>	<i>Alfabetyczne zestawienie algorytmów uogólnionych</i>	<i>1113</i>
	accumulate()	1115
	adjacent_difference()	1116
	adjacent_find()	1117
	binary_search()	1118
	copy()	1119
	copy_backward()	1120
	count()	1121
	count_if()	1123
	equal()	1124
	equal_range()	1126
	fill()	1128
	fill_n()	1129
	find()	1130

find_if()	1131
find_end()	1132
find_first_of()	1134
for_each()	1135
generate()	1136
generate_n()	1137
includes()	1138
inner_product()	1139
inplace_merge()	1140
iter_swap()	1142
lexicographical_compare()	1143
lower_bound()	1144
max()	1146
max_element()	1146
min()	1146
min_element()	1147
merge()	1148
mismatch()	1149
next_permutation()	1151
nth_element()	1152
partial_sort()	1153
partial_sort_copy()	1154
partial_sum()	1155
partition()	1157
prev_permutation()	1158
random_shuffle()	1158
remove()	1159
remove_copy()	1160
remove_if()	1161
remove_copy_if()	1161
replace()	1162
replace_copy()	1163
replace_if()	1163
replace_copy_if()	1164
reverse()	1165
reverse_copy()	1165
rotate()	1166
rotate_copy()	1167
search()	1168
search_n()	1169
set_difference()	1170
set_intersection()	1171
set_symmetric_difference()	1171
set_union()	1172

sort()	1174
stable_partition()	1174
stable_sort()	1175
swap()	1176
swap_ranges()	1177
transform()	1178
unique()	1180
unique_copy()	1181
upper_bound()	1182
Algorytmy stogowe.	1183
make_heap()	1184
pop_heap()	1184
push_heap()	1185
sort_heap()	1185
<i>Skorowidz.</i>	1187