

Spis treści

Przedmowa	13
Wstęp	15
Część I Programowanie niskiego poziomu w języku C	19
Wprowadzenie	21
1 Typy	23
1.1 Fundamentalne typy danych	23
1.2 Typy pochodne	25
1.3 Typy niekompletne	25
2 Stałe	27
2.1 Stałe całkowitoliczbowe	27
2.2 Stałe zmiennopozycyjne	28
2.3 Stałe znakowe	28
2.4 Znaki przestankowe	29
2.5 Stałe łańcuchowe	29
2.6 Stałe wyliczeniowe	30
3 Zmienne i ich nazwy	31
3.1 Kwalifikatory typu	32
3.2 Kwalifikatory miejsca	34
3.3 Kombinacje kwalifikatorów	36
3.4 Zakres ważności nazw zmiennych	36
4 Operatory i wyrażenia	38
4.1 Wyrażenia	38
4.2 Podstawowe operatory arytmetyczne	38
4.3 Operatory inkrementacji i dekrementacji	39
4.4 Operatory relacyjne	39
4.5 Operatory logiczne	40
4.6 Operator przecinka	41
4.7 Operatory bitowe	41

4.8	Operatory przypisania	43
4.9	Konwersje typów i operatory konwersji (rzutowania)	44
4.10	Wyrażenie warunkowe	46
4.11	Operator sizeof	47
4.12	Priorytety operatorów	47
4.13	Punkt sekwencji	48
5	Instrukcje sterujące przebiegiem programu	50
5.1	Instrukcje i bloki	50
5.2	Instrukcja warunkowa	50
5.3	Instrukcja wyboru	51
5.4	Pętle	52
5.5	Sterowanie pętlami — break i continue	54
5.6	Instrukcja skoku	54
6	Funkcje	55
6.1	Zwracanie wartości przez funkcję	56
6.2	Przesyłanie argumentów do funkcji przez wartość	57
6.3	Deklaracja zapowiadająca funkcji i pliki nagłówkowe	58
6.4	Rekurencja wywołań funkcji	59
6.5	Stary styl definicji funkcji	60
6.6	Funkcja o zmiennej liczbie argumentów wywołania	60
7	Wskaźniki i tablice	63
7.1	Definiowanie wskaźników	64
7.2	Wskaźniki jako argumenty funkcji	65
7.3	Tablice	66
7.4	Wskaźniki a tablice	69
7.5	Działania arytmetyczne na wskaźnikach	71
7.6	Wskaźniki nieokreślonego typu (void)	72
7.7	Wskaźniki do wskaźników	73
7.8	Wskaźniki do funkcji	75
7.9	Wskaźniki a kwalifikator const	76
8	Struktury i unie	78
8.1	Deklarowanie i definiowanie struktur	78
8.2	Praca ze strukturami	79
8.3	Przesyłanie struktur do i z funkcji	80
8.4	Tablice struktur	82
8.5	Struktury alokowane dynamicznie	84
8.6	Deklaracje typu	86
8.7	Unie	88
8.8	Pola bitowe	90

9	Preprocesor	91
9.1	Załączanie plików	92
9.2	Kompilacja warunkowa	92
9.3	Definiowanie symboli	93
9.4	Definiowanie makr	94
9.5	Makra predefiniowane	96
9.6	Inne dyrektywy	96
 Część II Programowanie obiektowe w języku C++		97
Wprowadzenie		99
10	Język C++ a język C	101
10.1	Stałe	102
10.2	Nowe typy danych	103
10.3	Referencje	104
10.4	Nazwy zastępcze	104
10.5	Priorytety operatorów języka C++	105
11	Funkcje w języku C++	107
11.1	Deklarowanie nazw funkcji	107
11.2	Funkcje typu „inline”	108
11.3	Argumenty domniemane funkcji	109
11.4	Nienazwany argument funkcji	110
11.5	Przekazywanie argumentów funkcji przez referencję	110
11.6	Przeładowanie nazw funkcji	112
11.7	Typy rozróżniane przy przeładowaniu	114
11.8	Etapy dopasowania przeładowanych funkcji	116
12	Klasy	120
12.1	Deklarowanie i definiowanie klas	121
12.2	Elementy składowe klasy	121
12.3	Enkapsulacja składników klasy	122
12.4	Deklaracja przyjaźni	124
12.5	Funkcje składowe	126
12.6	Funkcje składowe typu „inline”	127
12.7	Wskaźnik „this”	127
12.8	Przesłanianie nazw	128
12.9	Statyczny składnik klasy	129
12.10	Statyczna funkcja składowa klasy	131
12.11	Stały składnik klasy	132
12.12	Funkcja składowa typu <code>const</code>	133
12.13	Składnik dostrajalny (<code>mutable</code>)	134
12.14	Funkcja składowa typu <code>volatile</code>	135
12.15	Wskaźniki do składników klasy	135
12.16	Struktury i unie jako klasy	140

13 Konstruktory i destruktory	142
13.1 Deklarowanie i definiowanie konstruktora	142
13.2 Deklarowanie i definiowanie destruktora	144
13.3 Automatyczne wywołania konstruktora i destruktora	146
13.4 Przeladowanie konstruktora	147
13.5 Argumenty domniemane i konstruktor domniemany	148
13.6 Lista inicjalizacyjna konstruktora	149
13.7 Konstruktor kopiujący	150
13.8 Niepubliczny konstruktor	156
13.9 Singletony	157
14 Konwersja typów	161
14.1 Potrzeba konwersji	161
14.2 Konwersje automatyczne	162
14.3 Konstruktor jako konwerter	163
14.4 Jawne wywołanie konwersji	164
14.5 Rezygnacja z konwersji automatycznych	165
14.6 Operator konwersji	165
14.7 Jawne konwersje typów w języku C++	167
14.8 Konwersje a przeladowanie funkcji	169
15 Przeladowanie operatorów	174
15.1 Zasady przeladowania operatorów	175
15.2 Funkcja operatorowa jako składnik klasy	177
15.3 Globalna funkcja operatorowa	179
15.4 Operator przypisania	180
15.5 Operator przypisania a pseudoprzypisania	184
15.6 Operator pobrania adresu	185
15.7 Operator przecinka	186
15.8 Operatory <code>new</code> i <code>delete</code>	186
15.9 Operator tablicowy <code>[]</code>	197
15.10 Operator funkcyjny <code>()</code>	200
15.11 Operator odniesienia przez wskaźnik <code>-></code>	200
15.12 Operator odniesienia przez wskaźnik do składnika <code>->*</code>	205
15.13 Operatory inkrementacji i dekrementacji	206
16 Obsługa sytuacji wyjątkowych	208
16.1 Zgłaszanie wyjątków	209
16.2 Obsługa wyjątków	210
16.3 Rozróżnianie wyjątków	212
16.4 Wyjątki nigdzie nieobsłużone	213
16.5 Specyfikacje wyjątków	215
16.6 Różnice między wywołaniem funkcji a obsługą wyjątków	217
16.7 Wyjątki w konstruktorze	219
16.8 Wyjątki w destruktorze	223
16.9 Blok <code>try</code> na poziomie funkcji	225

16.10	Wyjątkowe bezpieczeństwo	226
16.11	Standardowe klasy wyjątków	228
17	Dziedziczenie i zawieranie klas	231
17.1	Zawieranie klas	231
17.2	Dziedziczenie	234
17.3	Reguła przesłaniania	236
17.4	Konstruktory klas podstawowych i składowych	237
17.5	Zagnieżdżona deklaracja klasy	239
17.6	Kolejność konstrukcji i destrukcji obiektów składowych	240
17.7	Sposoby dziedziczenia	241
17.8	Czego się nie dziedziczy?	245
17.9	Konwersje standardowe przy dziedziczeniu	247
17.10	Dziedziczenie wielokrotne	251
17.11	Wieloznaczość przy dziedziczeniu wielokrotnym	253
17.12	Dziedziczenie wirtualne	255
18	Funkcje wirtualne	259
18.1	Podstawy	259
18.2	Technikalia	262
18.3	Kiedy wystąpi polimorfizm?	263
18.4	Kiedy nie wystąpi polimorfizm?	265
18.5	Nietypowe funkcje wirtualne	266
18.6	Identyfikacja typu (RTTI)	268
18.7	Funkcje czysto wirtualne	272
18.8	Klasy abstrakcyjne	274
18.9	Wirtualne konstruktory i funkcje globalne	275
19	Szablony	280
19.1	Szablony funkcji	281
19.2	Szablony klas	282
19.3	Sposoby ukonkretnienia szablonu	284
19.4	Specjalizacja szablonu	285
19.5	Szablony z wieloma parametrami	286
19.6	Statyczny składnik w szablonie klasy	290
19.7	Szablony a przyjaźń	291
19.8	Inne uwagi dotyczące szablonów	295
20	Przestrzenie nazw	297
20.1	Deklarowanie przestrzeni nazw	297
20.2	Używanie przestrzeni nazw	298
20.3	Dyrektywa <code>using</code>	299
20.4	Instrukcja <code>using</code>	301
20.5	Anonimowe przestrzenie nazw	301

21 Elementy biblioteki standardowej	302
21.1 Iteratory	303
21.2 Klasy-pojemniki (kontenery)	308
21.3 Obiekty funkcyjne	318
21.4 Standardowe algorytmy	320
21.5 Klasa string	331
21.6 Standardowe wejście i wyjście	339
 Część III Programowanie symulacji Simulinka	 361
Wprowadzenie	363
 22 Podstawy działania s-funkcji	 365
22.1 Matematyczny opis bloku	365
22.2 Fazy symulacji	366
22.3 Bezpośrednie przejście sygnału przez blok	368
22.4 Dynamiczny rozmiar wejść i wyjść	369
22.5 Czas próbkowania dla bloku	369
 23 S-funkcje w języku Matlaba	 371
23.1 Format s-funkcji w języku Matlaba	371
23.2 Cechy s-funkcji	373
23.3 Przykład s-funkcji — wzmacniacz	373
23.4 Przykład s-funkcji — dynamika ciągła	375
23.5 Przykład s-funkcji — dynamika dyskretna	377
23.6 Przykład s-funkcji — układ hybrydowy	378
23.7 Wersja 2 interfejsu s-funkcji w języku Matlaba	381
 24 S-funkcje w języku C	 382
24.1 Kolejność wywoływania metod	383
24.2 Struktura SimStruct	387
24.3 Metody	388
24.4 Szablon s-funkcji w języku C	395
24.5 Kompilacja s-funkcji i osadzenie w modelu	399
24.6 Kreator s-funkcji	402
 25 Definiowanie właściwości bloku s-funkcji	 406
25.1 Zmienne s-funkcji	406
25.2 Sygnały wejściowe bloku	407
25.3 Sygnały wyjściowe bloku	415
25.4 Stany bloku	417
25.5 Parametry bloku s-funkcji	419
25.6 Zmienne robocze bloku	426
25.7 Definiowanie czasów próbkowania	431
25.8 Detekcja przejść przez zero	437
25.9 Definiowanie własnych typów danych	448

25.10	Obsługa błędów	450
25.11	Opcje symulacji	453
26	S-funkcje w języku C++	455
26.1	Szablon s-funkcji w języku C++	455
26.2	Przechowywanie obiektów w pamięci	459
26.3	Kompilacja s-funkcji w języku C++	461
 Część IV Symulowanie układów regulacji		463
Wprowadzenie		465
27	Dokumentacja w projektach programistycznych	468
27.1	Doxygen — wstęp	469
27.2	Dokumentowanie kodu	473
27.3	Formatowanie dokumentacji	477
28	Dyskretny obiekt symulacji	489
28.1	Model matematyczny	489
28.2	Programowanie symulacji modelu ARMAX	492
28.3	Sprawdzanie poprawności implementacji	494
29	Dyskretna pętla regulacji i interfejs dla klas regulatorów	496
29.1	Dyskretna pętla regulacji	496
29.2	Interfejs klasy regulatora	497
29.3	Prosty regulator	500
29.4	Programowanie dyskretnej pętli regulacji	500
29.5	Sprawdzanie poprawności implementacji	502
30	Regulacja PID	504
30.1	Ciągły regulator PID	504
30.2	Dyskretna postać regulatora PID	509
30.3	Implementacja regulatora PID	510
30.4	Regulator PID samonastrajalny	511
31	Regulacja predykcyjna	514
31.1	Pojęcia związane z regulacją predykcyjną	515
31.2	Algorytm regulacji predykcyjnej uogólnionej	517
31.3	Implementacja algorytmu GPC	519
31.4	Sprawdzanie poprawności działania	520
32	Identyfikacja parametryczna	522
32.1	Rekurencyjna ważona metoda najmniejszych kwadratów	522
32.2	Modyfikacje RMNK	524
32.3	Rozszerzona RMNK	527
32.4	Szybki algorytm RMNK	528
32.5	Implementacja algorytmu identyfikacji	529

33 Graficzny interfejs użytkownika	530
33.1 Trójstopniowa struktura programów w środowisku KDE	532
33.2 Sygnały i gniazda	541
33.3 Sygnały i gniazda w akcji	545
33.4 Grafika w KDE — tworzenie wykresu	552
33.5 Wykres w aplikacji	560
Bibliografia	566
Skorowidz	568