

Spis treści

Wstęp	7
Część pierwsza. Podstawy programowania.....	9
1. Podstawy programowania.....	11
1.1. Co to jest program komputerowy	11
1.2. Algorytmy	11
1.3. Języki programowania	13
1.4. Jak jest realizowany program komputerowy	16
1.5. Cykl powstawania programu	17
1.5.1. Utworzenie kodu źródłowego.....	18
1.5.2. Kompilacja kodu źródłowego.....	18
1.5.3. Konsolidacja	19
1.5.4. Uruchamianie i testowanie programu.....	20
1.5.5. Cykl tworzenia programu	20
2. Algorytmy	23
2.1. Sposób przedstawiania algorytmów	23
2.1.1. Opis słowny	24
2.1.2. Lista kroków	24
2.1.3. Schemat blokowy	24
2.1.4. Zapis w języku programowania.....	26
2.2. Elementy algorytmów.....	27
2.2.1. Sekwencja operacji	27
2.2.2. Rozgałęzienie.....	28
2.2.3. Pętla	28
2.3. Złożoność algorytmu	30
2.4. Iteracja i rekurencja	31
2.4.1. Iteracja	32
2.4.2. Rekurencja.....	33
Część druga. Pascal i C/C++, programowanie strukturalne	35
3. Środowisko programistyczne Free Pascal	37
4. Elementy języka Pascal.....	41
4.1. Struktura prostego programu w Pascalu	41
4.2. Nazwy, komentarze, estetyka i czytelność programu	45
4.2.1. Nazwy	45
4.2.2. Komentarze i akapitowanie	46

4.3.	Typy danych – do czego służą zmienne	47
4.3.1.	Typy podstawowe.....	49
4.3.2.	Typ porządkowy	49
4.3.3.	Typ BOOLEAN.....	50
4.3.4.	Typ STRING	50
4.3.5.	Typy definiowane przez użytkownika.....	51
4.4.	Instrukcje sterujące, zastosowanie do realizacji elementów algorytmów	57
4.4.1.	if...then lub if...then...else	57
4.4.2.	case	59
4.4.3.	repeat...until	61
4.4.4.	while...do	63
4.4.5.	for...to/downto...do	64
4.5.	Operatory arytmetyczne i logiczne	66
4.5.1.	Operatory arytmetyczne	67
4.5.2.	Operatory logiczne	69
4.6.	Procedury i funkcje	74
4.6.1.	Procedury	74
4.6.2.	Funkcje	78
4.6.3.	Zmienne globalne i lokalne. Zasięg nazw	79
4.7.	Projektowanie programu od ogółu do szczegółu (top-down) i od szczegółu do ogółu (bottom-up).....	81
4.8.	Uruchamianie programu, debugger – podstawy	83
4.8.1.	Techniki uruchamiania programów	83
4.8.2.	Debugowanie we Free Pascalu	84
4.9.	Obsługa wejścia/wyjścia.....	89
4.9.1.	Standardowe wejście/wyjście i moduł CRT	89
4.9.2.	Operacje na plikach	97
4.10.	Wskaźniki i dynamiczna rezerwacja pamięci	104
4.10.1.	Wskaźniki	104
4.10.2.	Dynamiczna rezerwacja pamięci	107
4.10.3.	Dynamiczne struktury danych.....	107
4.11.	Jak to zrobić?	110
4.11.1.	Sekwencja działań i instrukcja blokowa.....	111
4.11.2.	Moment decyzyjny	111
4.11.3.	Pętla, w której warunek sprawdzamy na początku.....	113
4.11.4.	Pętla, w której warunek sprawdzamy na końcu.....	114
4.11.5.	Wybór jednej z kilku dróg realizacji programu.....	115
5.	Elementy języka C/C++ – programowanie strukturalne	117
5.1.	Struktura programu w C/C++	118
5.1.1.	Kilka uwag o wprowadzaniu i wyprowadzaniu informacji w C++	118
5.1.2.	Struktura programu.....	119
5.1.3.	Nazwy (identyfikatory) w C/C++	124
5.2.	Środowisko C++ Builder 6	125
5.3.	Komentarze i czytelność programu	129

5.4.	Wbudowane typy danych i deklarowanie zmiennych	130
5.4.1.	Wbudowane typy danych	130
5.4.2.	Deklarowanie zmiennych	133
5.4.3.	Dyrektywa #define	133
5.5.	Operatory	134
5.5.1.	Operator przypisania (podstawienia)	135
5.5.2.	Operatory arytmetyczne dwuargumentowe	135
5.5.3.	Operatory arytmetyczne jednoargumentowe	137
5.5.4.	Pozostałe operatory przypisania	138
5.5.5.	Operatory logiczne	139
5.5.6.	Operator sizeof()	143
5.5.7.	Operator przecinek	145
5.5.8.	Konwersje typów danych. Operator rzutowania	145
5.6.	Priorytety operatorów	148
5.7.	Przykłady użycia operatorów	151
5.7.1.	Przykłady realizacji sekwencji działań arytmetycznych	151
5.7.2.	Przykłady tworzenia warunków	152
5.8.	Instrukcje sterujące	153
5.8.1.	Instrukcja if..else	154
5.8.2.	Instrukcja pętli for	156
5.8.3.	Instrukcja do..while	157
5.8.4.	Pętle zagnieżdżone	159
5.8.5.	Instrukcja while	159
5.8.6.	Instrukcja break	161
5.8.7.	Instrukcja switch	162
5.8.8.	Instrukcja continue	164
5.8.9.	Instrukcja goto	165
5.9.	Funkcje – podstawy	166
5.9.1.	Definiowanie funkcji	167
5.9.2.	Przekazywanie parametrów do funkcji – część pierwsza	176
5.9.3.	Funkcje typu inline	179
5.9.4.	Zmienne globalne i lokalne	180
5.10.	Typy danych definiowane przez użytkownika	183
5.10.1.	Tablice	184
5.10.2.	Tablice znakowe i przetwarzanie łańcuchów znaków	187
5.10.3.	Struktury	190
5.11.	Wskaźniki i referencje	192
5.11.1.	Wskaźniki	192
5.11.2.	Referencje	198
5.11.3.	Kilka słów o czytaniu deklaracji	199
5.12.	Przekazywanie parametrów do funkcji – część druga	201
5.12.1.	Przekazywanie parametrów do funkcji przez wartość, wskaźnik i referencję	201
5.12.2.	Przekazywanie do funkcji tablic i struktur	203
5.13.	Dynamiczna rezerwacja pamięci, przykłady zastosowań	208

5.13.1.	Operatory new i delete.....	209
5.13.2.	Funkcje rezerwujące pamięć w języku C	213
5.14.	Kilka przykładów funkcji bibliotecznych.....	215
6.	Podstawy obsługi wejścia/wyjścia.....	221
6.1.	Obsługa wejścia/wyjścia w języku C.....	221
6.2.	Obsługa wejścia/wyjścia w języku C++	229
6.2.1.	Klasy, strumienie i pliki nagłówkowe do obsługi wejścia/wyjścia w C++	229
6.2.2.	Operatory wprowadzania i wyprowadzania informacji dla strumieni	230
6.2.3.	Flagi stanu formatowania	231
6.2.4.	Domniemania w pracy strumieni predefiniowanych.....	232
6.2.5.	Sterowanie formatem wyświetlanej informacji.....	233
7.	Podstawy operacji na plikach	239
7.1.	Otwieranie i zamykanie pliku	239
7.2.	Pliki tekstowe i binarne.....	241
7.3.	Wejście/wyjście sformatowane i niesformatowane	241
7.3.1.	Wejście/wyjście sformatowane	242
7.3.2.	Wejście/wyjście niesformatowane. Dostęp sekwencyjny do pliku.....	247
7.3.3.	Dostęp swobodny do pliku	251
8.	Obsługa debuggera C++ Buildera	259
8.1.	Śledzenie realizacji programu – debugger.....	259
8.2.	Okno procesora – przykłady kompilacji instrukcji.....	264
	Podstawowe pliki nagłówkowe i biblioteki języka C	267
	Dodatek A. Najważniejsze komunikaty kompilatora o błędach	269
	Dodatek B. Przykłady zadań testowych.....	273
	Bibliografia.....	277
	Indeks.....	279