

## Spis treści

<b>1. Wstęp</b> .....	9
1.1. Inżynieria oprogramowania jako proces .....	10
1.1.1. Algorytm .....	11
1.2. Programowanie w językach wysokiego poziomu .....	11
1.3. Obiektowe podejście do programowania .....	12
1.3.1. Rozwój technik programowania .....	12
1.3.2. Cechy obiektowego podejścia do programowania .....	13
1.3.3. Języki realizujące wsparcie dla programowania obiektowego .....	14
<b>2. Projektowanie systemów informatycznych</b> .....	15
2.1. Architektura oparta na modelu (MDA) .....	15
2.1.1. Struktura MDA .....	16
2.1.2. Typowy przebieg procesu .....	17
2.2. Ogólna charakterystyka języka UML .....	18
2.3. Diagramy struktury .....	19
2.3.1. Diagramy klas .....	19
2.3.1.1. Klasy .....	19
2.3.1.2. Interfejsy .....	20
2.3.1.3. Zależności .....	21
2.3.1.4. Powiązania .....	21
2.3.1.5. Klasy powiązania .....	23
2.3.1.6. Uogólnienia .....	24
2.3.1.7. Klasy zagnieżdżone .....	24
2.4. Diagramy dynamiki .....	25
2.4.1. Diagramy przypadków użycia .....	25
2.4.2. Diagramy sekwencji .....	27
2.4.2.1. Jawne tworzenie i usuwanie obiektów .....	30
2.4.2.2. Specyfikowanie parametrów czasowych komunikatów .....	30
2.4.2.3. Fragmenty wyodrębnione .....	31
2.4.2.4. Rozgałęzienia .....	32
2.4.2.5. Iteracje .....	33
2.4.2.6. Zagnieżdżanie diagramów sekwencji .....	34

2.4.3. Diagram stanów .....	34
2.4.3.1. Stany .....	35
2.4.3.2. Przejścia .....	36
2.4.3.3. Zdarzenia .....	36
2.4.3.4. Stany złożone .....	37
2.4.3.5. Punkty wejścia i wyjścia .....	38
2.4.3.6. Punkty węzłowe i obszary współbieżne .....	39
2.4.3.7. Historia .....	40
2.4.4. Diagram czynności .....	41
2.4.4.1. Partycje .....	41
2.5. Metodyki projektowe .....	43
2.5.1. Projekty o małej skali .....	43
2.5.2. Modele liniowe .....	44
2.5.3. Modele iteracyjne .....	45
2.5.3.1. Programowanie ekstremalne .....	46
2.5.3.2. Metodyka <i>Scrum</i> .....	47
2.5.3.3. Model spiralny .....	48
2.5.3.4. <i>Rational Unified Process</i> .....	50
<b>3. Implementacja w językach C i C++ .....</b>	<b>54</b>
3.1. Struktura programu w języku C .....	54
3.1.1. Identyfikatory .....	56
3.1.2. Słowa kluczowe .....	56
3.1.3. Zmienne .....	56
3.1.4. Typy danych .....	57
3.1.4.1. Typy proste .....	57
3.1.4.2. Typy pochodne .....	58
3.1.5. Funkcje .....	70
3.1.5.1. Wskaźniki do funkcji .....	72
3.1.6. Instrukcje .....	73
3.1.6.1. Instrukcje wyrażeniowe .....	73
3.1.6.2. Instrukcja warunkowa .....	74
3.1.6.3. Instrukcja wyboru .....	76
3.1.6.4. Instrukcja złożona (blokowa) .....	76
3.1.6.5. Instrukcje pętli programowych .....	77
3.1.7. Wyrażenia .....	80
3.1.7.1. Operatory arytmetyczne .....	80
3.1.7.2. Operacje przypisania .....	80
3.1.7.3. Operacje porównania .....	80
3.1.7.4. Operacje logiczne .....	81
3.1.7.5. Operacje bitowe .....	81
3.1.7.6. Operacja warunkowa .....	81
3.1.8. Operacje wejścia-wyjścia .....	82
3.1.8.1. Funkcje wejścia-wyjścia języka C .....	82

3.2. Obiektowe programowanie w języku C++ .....	92
3.2.1. Typ obiektowy (klasa) .....	92
3.2.1.1. Składnia deklaracji klasy .....	92
3.2.1.2. Reprezentacja obiektów w pamięci .....	93
3.2.1.3. Słowo kluczowe <i>this</i> .....	93
3.2.1.4. Typy referencyjne .....	94
3.2.1.5. Funkcje przeciążone .....	94
3.2.1.6. Konstruktor i destruktor .....	95
3.2.1.7. Przykład deklaracji klasy .....	96
3.2.1.8. Prosty przykład programu obiektowego .....	97
3.2.2. Argumenty domniemane funkcji .....	98
3.2.3. Funkcje operatorowe .....	99
3.2.3.1. Strumieniowe operatory << i >> .....	100
3.2.3.2. Statyczne składniki klasy .....	101
3.2.4. Zaprzyjaźnienia .....	102
3.2.5. Klasy zagnieżdżone .....	104
3.2.6. Szablony .....	106
3.2.6.1. Przykład szablonu funkcyjnego .....	106
3.2.6.2. Przykład szablonu klasy .....	107
3.2.6.3. Implementacja wektora o zmiennej liczbie elementów .....	108
3.2.7. Relacja całość-część .....	110
3.2.8. Dziedziczenie .....	112
3.2.8.1. Deklarowanie klas pochodnych .....	113
3.2.8.2. Klasy abstrakcyjne .....	116
3.2.9. Reakcja programu na sytuacje wyjątkowe .....	117
3.2.9.1. Obsługa wyjątków systemowych .....	118
3.2.9.2. Obsługa wyjątków programowych .....	119
3.2.9.3. Specyfikowanie wyjątków dla funkcji .....	120
3.2.9.4. Unifikacja obsługi wyjątków systemowych i programowych ...	121
3.2.9.5. Standardowe klasy wyjątków .....	122
3.2.10. Dynamiczne struktury danych .....	123
3.2.10.1. Stos ( <i>stack</i> ) .....	123
3.2.10.2. Kolejka ( <i>queue</i> ) .....	124
3.2.10.3. Lista dwukierunkowa .....	125
3.2.10.4. Drzewa .....	126
3.2.11. Biblioteka Standardowa C++ .....	127
3.2.11.1. Biblioteka strumieni wejścia-wyjścia .....	127
3.2.11.2. Zastosowanie STL do dynamicznych struktur danych .....	131
<b>4. Implementacja w języku Java .....</b>	<b>136</b>
4.1. Środowisko wykonania programów .....	136
4.2. Składnia .....	137
4.2.1. Pierwszy program .....	137
4.2.2. Klasa .....	138

4.2.2.1. Konstruktor .....	139
4.2.2.2. Inicjator klasy i obiektu .....	140
4.2.2.3. Destruktor .....	141
4.2.2.4. Słowo kluczowe <i>this</i> .....	141
4.2.3. Dziedziczenie .....	141
4.2.4. Klasy abstrakcyjne .....	143
4.2.5. Tworzenie i używanie obiektów .....	143
4.2.6. Kopiowanie obiektów .....	144
4.2.7. Tablice .....	145
4.2.8. Klasy wewnętrzne i lokalne .....	146
4.2.9. Interfejsy .....	148
4.2.10. Aplety .....	150
4.3. Obsługa zdarzeń .....	152
4.3.1. Rodzaje zdarzeń .....	154
4.4. Obsługa sytuacji wyjątkowych .....	155
4.4.1. Klasy obsługi wyjątków .....	156
4.4.2. Własne typy wyjątków .....	156
4.4.3. Generowanie wyjątków .....	157
4.5. Instrukcje wejścia-wyjścia .....	158
4.5.1. Bajtowo zorientowane wejście .....	158
4.5.2. Bajtowo zorientowane wyjście .....	159
4.5.3. Operacje dla danych typów wbudowanych .....	159
4.5.4. Wejście-wyjście dla obiektów .....	163
4.6. Programowanie wielowątkowe .....	166
4.6.1. Tworzenie wątków .....	166
4.6.2. Stany wątków .....	169
4.6.3. Priorytety wątków .....	169
4.6.4. Synchronizacja wątków .....	173
<b>5. Obiektowe wzorce projektowe .....</b>	<b>174</b>
5.1. Wzorce <i>Singleton</i> i <i>Multipleton</i> .....	174
5.1.1. Schemat wzorców <i>Singleton</i> i <i>Multipleton</i> .....	175
5.1.2. Implementacja wzorca <i>Singleton</i> i <i>Multipleton</i> .....	175
5.2. Wzorzec <i>Iterator</i> .....	177
5.2.1. Składniki wzorca <i>Iterator</i> .....	177
5.2.2. Konsekwencje zastosowania wzorca <i>Iterator</i> .....	177
5.2.3. Schemat wzorca <i>Iterator</i> .....	177
5.2.4. Implementacja wzorca <i>Iterator</i> .....	177
5.2.4.1. Implementacja klas abstrakcyjnych .....	177
5.2.4.2. Implementacja klasy kolekcji elementów .....	179
5.2.4.3. Porównanie implementacji konkretnych iteratorów .....	179
5.2.5. Przykładowy kod napisany z użyciem klas wzorca <i>Iterator</i> .....	180
5.3. Wzorzec <i>Obserwator</i> ( <i>Observer</i> ) .....	181

5.3.1. Składniki wzorca <i>Obserwator</i> .....	181
5.3.2. Schemat wzorca <i>Obserwator</i> .....	181
5.3.3. Implementacja wzorca <i>Obserwator</i> .....	182
5.3.3.1. Implementacja klas abstrakcyjnych .....	182
5.3.3.2. Implementacja klas konkretnych .....	183
5.3.4. Przykładowy kod napisany z użyciem klas wzorca <i>Obserwator</i> .....	184
5.4. Wzorzec <i>Stan (State)</i> .....	184
5.4.1. Zastosowania wzorca <i>Stan</i> .....	184
5.4.2. Schemat wzorca <i>Stan</i> .....	185
5.4.3. Składniki wzorca <i>Stan</i> .....	185
5.4.4. Konsekwencje zastosowania wzorca <i>Stan</i> .....	185
5.4.5. Implementacja wzorca <i>Stan</i> .....	186
5.4.6. Przykładowy kod napisany z użyciem klas wzorca <i>Stan</i> .....	188
5.5. Wzorzec <i>Metoda Wytwórcza (Factory Method)</i> .....	188
5.5.1. Zastosowania wzorca <i>Metoda Wytwórcza</i> .....	189
5.5.2. Składniki wzorca <i>Metoda Wytwórcza</i> .....	189
5.5.3. Schemat wzorca <i>Metoda Wytwórcza</i> .....	189
5.5.4. Konsekwencje zastosowania wzorca <i>Metoda Wytwórcza</i> .....	189
5.5.5. Implementacja wzorca <i>Metoda Wytwórcza</i> .....	190
5.5.6. Przykładowy kod napisany z użyciem klas wzorca <i>Metoda Wytwórcza</i> ...	192
5.6. Wzorzec <i>Abstrakcyjna Fabryka (Abstract Factory)</i> .....	192
5.6.1. Zastosowania wzorca <i>Abstrakcyjna Fabryka</i> .....	192
5.6.2. Schemat wzorca <i>Abstrakcyjna Fabryka</i> .....	193
5.6.3. Składniki wzorca <i>Abstrakcyjna Fabryka</i> .....	193
5.6.4. Konsekwencje zastosowania wzorca <i>Abstrakcyjna Fabryka</i> .....	194
5.6.5. Implementacja wzorca <i>Abstrakcyjna Fabryka</i> .....	194
5.6.6. Przykładowy kod napisany z użyciem klas wzorca <i>Abstrakcyjna Fabryka</i> .....	196
5.7. Wzorzec <i>Kompozyt (Composite)</i> .....	197
5.7.1. Zastosowania wzorca <i>Kompozyt</i> .....	197
5.7.2. Składniki wzorca <i>Kompozyt</i> .....	197
5.7.3. Schemat wzorca <i>Kompozyt</i> .....	198
5.7.4. Konsekwencje zastosowania wzorca <i>Kompozyt</i> .....	198
5.7.5. Implementacja wzorca <i>Kompozyt</i> .....	199
5.7.6. Przykładowy kod napisany z użyciem klas wzorca <i>Kompozyt</i> .....	201
<b>6. Weryfikacja i walidacja</b> .....	<b>203</b>
6.1. Testowanie oprogramowania .....	203
6.1.1. Definicje .....	204
6.1.2. Metody testowania .....	205
6.1.2.1. Metody funkcjonalne .....	206
6.1.2.2. Metody strukturalne .....	206
6.1.3. Zakres testowania .....	206
6.1.3.1. Testy jednostek .....	206

6.1.3.2. Testy scalenia .....	207
6.1.3.3. Testy systemów użytkowych .....	208
6.1.4. Uruchamianie .....	208
6.1.4.1. Logowanie danych .....	208
6.1.4.2. Asercje .....	208
6.1.4.3. Interaktywne narzędzia uruchomieniowe .....	209
6.1.5. Profilowanie ( <i>profiling</i> ) .....	209
<b>7. Przegląd narzędzi .....</b>	<b>211</b>
7.1. Narzędzia do modelowania graficznego .....	212
7.1.1. <i>Rhapsody</i> firmy IBM/Telelogic .....	212
7.1.2. <i>Borland Together</i> .....	212
7.1.3. <i>IBM Rational System Architect</i> .....	213
7.1.4. <i>Enterprise Architect</i> firmy Sparx Systems .....	214
7.2. Zintegrowane środowiska rozwijania aplikacji .....	216
7.2.1. <i>NetBeans</i> .....	216
7.2.2. <i>Eclipse</i> .....	217
7.2.3. <i>Microsoft Visual Studio</i> .....	219
7.3. Środowiska do projektowania procesu .....	220
7.3.1. <i>Rational Method Composer</i> .....	220
7.3.2. <i>Eclipse Process Framework</i> .....	220
<b>8. Przykład realizacji systemu .....</b>	<b>222</b>
8.1. Opis wymagań .....	222
8.2. Analiza .....	223
8.2.1. Model przypadków użycia .....	223
8.2.1.1. Dokumentacja przypadków użycia .....	226
8.2.2. Model zachowania .....	227
8.2.2.1. Diagram sekwencji .....	227
8.2.2.2. Diagram maszyny stanowej .....	229
8.2.3. Model struktury .....	232
8.2.3.1. Pakiet <i>Main</i> .....	232
8.2.3.2. Pakiet <i>Shapes</i> .....	234
8.3. Implementacja .....	236
8.4. Uruchamianie .....	237
8.5. Testowanie .....	237
8.6. Wnioski .....	239
<b>Literatura .....</b>	<b>240</b>
<b>Indeks alfabetyczny .....</b>	<b>241</b>