

Spis treści

O autorze	13
Wstęp	15
Tematyka książki	15
Nowości w czwartym wydaniu	17
Adresaci książki	17
Metodyka	18
Konwencje	19
Kod źródłowy	19
Rozdział 1 Podstawowe pojęcia obiektowości	21
Podstawowe pojęcia	22
Obiekty a stare systemy	22
Programowanie obiektowe a proceduralne	24
Zamiana podejścia proceduralnego na obiektowe	27
Programowanie proceduralne	27
Programowanie obiektowe	28
Definicja obiektu	28
Dane obiektu	29
Zachowania obiektu	29
Definicja klasy	33
Tworzenie obiektów	34
Atrybuty	35
Metody	36
Komunikaty	36
Modelowanie klas przy użyciu diagramów UML	36
Hermetyzacja i ukrywanie danych	37
Interfejsy	37
Implementacje	38
Realistyczna ilustracja paradygmatu interfejsu i implementacji	39
Model paradygmatu interfejs – implementacja	39

Dziedziczenie	40
Nadklasy i podklasy	42
Abstrakcja	42
Związek typu „jest”	42
Polimorfizm	44
Kompozycja	47
Abstrakcja	47
Związek typu „ma”	47
Podsumowanie	48
Listingi	48
TestPerson	48
TestShape	49
Rozdział 2 Myślenie w kategoriach obiektowych	51
Różnica między interfejsem a implementacją	52
Interfejs	54
Implementacja	54
Przykład implementacji i interfejsu	55
Zastosowanie myślenia abstrakcyjnego w projektowaniu interfejsów	59
Minimalizowanie interfejsu	61
Określanie grupy docelowej	62
Zachowania obiektu	63
Ograniczenia środowiska	63
Identyfikowanie publicznych interfejsów	63
Identyfikowanie implementacji	64
Podsumowanie	65
Źródła	65
Rozdział 3 Zaawansowane pojęcia z zakresu obiektowości	67
Konstruktory	67
Kiedy wywoływany jest konstruktor	68
Zawartość konstruktora	68
Konstruktor domyślny	69
Zastosowanie wielu konstruktorów	70
Projektowanie konstruktorów	73
Obsługa błędów	74
Ignorowanie problemu	74
Szukanie błędów i kończenie działania programu	75
Szukanie błędów i próba ich naprawienia	75
Zgłaszanie wyjątków	76
Pojęcie zakresu	78
Atrybuty lokalne	78
Atrybuty obiektowe	79
Atrybuty klasowe	81

Przeciążanie operatorów	82
Wielokrotne dziedziczenie	83
Operacje obiektów	84
Podsumowanie	85
Źródła	85
Listingi	86
TestNumber	86
Rozdział 4 Anatomia klasy	87
Nazwa klasy	87
Komentarze	89
Atrybuty	89
Konstruktory	91
Metody dostępne	93
Metody interfejsu publicznego	95
Prywatne metody implementacyjne	95
Podsumowanie	96
Źródła	96
Listingi	96
TestCab	96
Rozdział 5 Wytyczne dotyczące projektowania klas	99
Modelowanie systemów świata rzeczywistego	99
Identyfikowanie interfejsów publicznych	100
Minimalizacja interfejsu publicznego	100
Ukrywanie implementacji	101
Projektowanie niezawodnych konstruktorów i destruktorów	102
Projektowanie mechanizmu obsługi błędów w klasie	103
Pisanie dokumentacji i stosowanie komentarzy	103
Tworzenie obiektów nadających się do kooperacji	104
Wielokrotne użycie kodu	104
Rozszerzalność	105
Tworzenie opisowych nazw	105
Wyodrębnianie nieprzenośnego kodu	106
Umożliwianie kopiowania i porównywania obiektów	107
Ograniczanie zakresu	107
Klasa powinna odpowiadać sama za siebie	108
Konservacja kodu	109
Iteracja	110
Testowanie interfejsu	110
Wykorzystanie trwałości obiektów	112
Serializacja i szeregowanie obiektów	113
Podsumowanie	113
Źródła	114

Listingi	114
TestMath	114
Rozdział 6 Projektowanie z wykorzystaniem obiektów	115
Wytyczne dotyczące projektowania	115
Wykonanie odpowiedniej analizy	119
Określanie zakresu planowanych prac	119
Gromadzenie wymagań	120
Opracowywanie prototypu interfejsu użytkownika	120
Identyfikowanie klas	120
Definiowanie wymagań wobec każdej z klas	121
Określenie warunków współpracy między klasami	121
Tworzenie modelu klas opisującego system	121
Tworzenie prototypu interfejsu użytkownika	121
Obiekty opakowujące	122
Kod strukturalny	122
Opakowywanie kodu strukturalnego	124
Opakowywanie nieprzenośnego kodu	125
Opakowywanie istniejących klas	126
Podsumowanie	127
Źródła	128
Rozdział 7 Dziedziczenie i kompozycja	129
Wielokrotne wykorzystywanie obiektów	129
Dziedziczenie	131
Generalizacja i specjalizacja	133
Decyzje projektowe	134
Kompozycja	136
Reprezentowanie kompozycji na diagramach UML	137
Czemu hermetyzacja jest podstawą technologii obiektowej	138
Jak dziedziczenie osłabia hermetyzację	139
Szczegółowy przykład wykorzystania polimorfizmu	141
Odpowiedzialność obiektów	141
Klasy abstrakcyjne, metody wirtualne i protokoły	145
Podsumowanie	146
Źródła	147
Listingi	147
TestShape	147
Rozdział 8 Wielokrotne wykorzystanie kodu	
— interfejsy i klasy abstrakcyjne	149
Wielokrotne wykorzystanie kodu	149
Infrastruktura programistyczna	150
Co to jest kontrakt	152
Klasy abstrakcyjne	153
Interfejsy	156

Wnioski	158
Dowód kompilatora	160
Zawieranie kontraktu	161
Punkty dostępowe do systemu	163
Przykład biznesu elektronicznego	163
Biznes elektroniczny	164
Podejście niezakładające wielokrotnego wykorzystania kodu	165
Rozwiązanie dla aplikacji biznesu elektronicznego	167
Model obiektowy UML	167
Podsumowanie	170
Źródła	170
Listingi	170
TestShop	171
Rozdział 9 Tworzenie obiektów	175
Relacje kompozycji	175
Podział procesu budowy na etapy	177
Rodzaje kompozycji	179
Agregacja	179
Asocjacja	180
Łączne wykorzystanie asocjacji i agregacji	181
Unikanie zależności	182
Liczność	183
Kilka asocjacji	184
Asocjacje opcjonalne	186
Praktyczny przykład	186
Podsumowanie	187
Źródła	188
Rozdział 10 Tworzenie modeli obiektowych	189
Co to jest UML	189
Struktura diagramu klasy	190
Atrybuty i metody	192
Atrybuty	192
Metody	192
Określanie dostępności	193
Dziedziczenie	194
Interfejsy	195
Kompozycja	196
Agregacja	196
Asocjacja	197
Liczność	199
Podsumowanie	200
Źródła	201

Rozdział 11 Obiekty i dane przenośne — XML	203
Przenośność danych	204
Rozszerzalny język znaczników — XML	205
XML a HTML	206
XML a języki obiektowe	207
Wymiana danych między firmami	208
Sprawdzanie poprawności dokumentu względem DTD	208
Integrowanie DTD z dokumentem XML	210
Kaskadowe arkusze stylów	216
Notacja obiektowa języka JavaScript (JSON)	217
Podsumowanie	222
Źródła	223
Rozdział 12 Obiekty trwałe — serializacja i relacyjne bazy danych	225
Podstawy trwałości obiektów	225
Zapisywanie obiektu w pliku płaskim	226
Serializacja pliku	227
Jeszcze raz o implementacji i interfejsach	229
Serializacja metod	231
Serializacja przy użyciu języka XML	231
Zapisywanie danych w relacyjnej bazie danych	233
Dostęp do relacyjnej bazy danych	235
Podsumowanie	237
Źródła	237
Listingi	238
Klasa Person	238
Rozdział 13 Obiekty w usługach sieciowych, aplikacjach mobilnych i aplikacjach hybrydowych	241
Ewolucja technik przetwarzania rozproszonego	241
Obiektowe skryptowe języki programowania	242
Weryfikacja danych za pomocą języka JavaScript	245
Obiekty na stronach internetowych	248
Obiekty JavaScript	248
Kontrolki na stronach internetowych	250
Odtwarzacze dźwięku	250
Odtwarzacze filmów	251
Animacje Flash	252
Obiekty rozproszone i systemy przedsiębiorstw	252
CORBA	254
Definicja usługi sieciowej	257
Kod usług sieciowych	261
Representational State Transfer (ReST)	263
Podsumowanie	264
Źródła	264

Rozdział 14 Obiekty w aplikacjach typu klient-serwer	265
Model klient-serwer	265
Rozwiązanie własnościowe	266
Kod obiektu do serializacji	266
Kod klienta	267
Kod serwera	269
Uruchamianie aplikacji	270
Technika z wykorzystaniem XML	271
Definicja obiektu	272
Kod klienta	273
Kod serwera	274
Uruchamianie programu	276
Podsumowanie	276
Źródła	276
Listingi	277
Rozdział 15 Wzorce projektowe	279
Historia wzorców projektowych	280
Wzorzec MVC języka Smalltalk	280
Rodzaje wzorców projektowych	283
Wzorce konstrukcyjne	283
Wzorce strukturalne	288
Wzorce czynnościowe	290
Antywzorce	291
Podsumowanie	292
Źródła	292
Listingi	293
Counter.cs	293
Singleton.cs	293
MailTool.cs	294
MailInterface.cs	294
MyMailTool.cs	295
Adapter.cs	295
Iterator.cs	296
Skorowidz	297