

Spis treści

O autorze 11

Podziękowania 11

Wstęp 13

Rozdział 1. Podstawowe pojęcia obiektowości 17

Podstawowe pojęcia 17

Obiekty a stare systemy 18

Programowanie obiektowe a proceduralne 19

Zamiana podejścia proceduralnego na obiektowe 23

Programowanie proceduralne 23

Programowanie obiektowe 23

Definicja obiektu 24

Dane obiektu 24

Zachowania obiektu 24

Definicja klasy 28

Tworzenie obiektów 28

Atrybuty 30

Metody 30

Komunikaty 30

Modelowanie klas przy użyciu diagramów UML 31

Hermetyzacja i ukrywanie danych 31

Interfejsy 31

Implementacje 32

Realistyczna ilustracja paradygmatu interfejsu i implementacji 33

Model paradygmatu interfejs - implementacja 33

Dziedziczenie	34
Nadklasy i podklasy	35
Abstrakcja	36
Związek typu "jest"	37
Polimorfizm	38
Kompozycja	41
Abstrakcja	41
Związek typu "ma"	41
Podsumowanie	41
Rozdział 2. Myślenie w kategoriach obiektowych	43
Różnica między interfejsem a implementacją	44
Interfejs	46
Implementacja	46
Przykład implementacji i interfejsu	47
Zastosowanie myślenia abstrakcyjnego w projektowaniu interfejsów	51
Minimalizowanie interfejsu	52
Określanie grupy docelowej	53
Zachowania obiektu	54
Ograniczenia środowiska	54
Identyfikowanie publicznych interfejsów	54
Identyfikowanie implementacji	55
Podsumowanie	56
Źródła	56
Rozdział 3. Inne pojęcia z zakresu obiektowości	57
Konstruktory	57
Kiedy wywoływany jest konstruktor	58
Zawartość konstruktora	58

Konstruktor domyślny	58
Zastosowanie wielu konstruktorów	59
Projektowanie konstruktorów	63
Obsługa błędów	63
Ignorowanie problemu	64
Szukanie błędów i kończenie działania programu	64
Szukanie błędów i próba ich naprawienia	64
Zgłaszanie wyjątków	65
Pojęcie zakresu	67
Atrybuty lokalne	67
Atrybuty obiektowe	68
Atrybuty klasowe	70
Przeciążanie operatorów	71
Wielokrotne dziedziczenie	72
Operacje obiektów	73
Podsumowanie	74
Źródła	74
Rozdział 4. Anatomia klasy	75
Nazwa klasy	75
Komentarze	77
Atrybuty	77
Konstruktory	78
Metody dostępne	80
Metody interfejsu publicznego	83
Prywatne metody implementacyjne	83
Podsumowanie	84
Źródła	84

Rozdział 5. Wytyczne dotyczące projektowania klas 85

Modelowanie systemów świata rzeczywistego 85

Identyfikowanie interfejsów publicznych 86

Minimalizacja interfejsu publicznego 86

Ukrywanie implementacji 87

Projektowanie niezawodnych konstruktorów i destruktorów 88

Projektowanie mechanizmu obsługi błędów w klasie 89

Pisanie dokumentacji i stosowanie komentarzy 89

Tworzenie obiektów nadających się do kooperacji 90

Wielokrotne użycie kodu 90

Rozszerzalność 91

Tworzenie opisowych nazw 91

Wyodrębnianie nieprzenośnego kodu 92

Umożliwianie kopiowania i porównywania obiektów 92

Ograniczanie zakresu 93

Konserwacja kodu 94

Iteracja 94

Testowanie interfejsu 95

Wykorzystanie trwałości obiektów 96

Serializacja i szeregowanie obiektów 97

Podsumowanie 98

Źródła 98

Rozdział 6. Projektowanie z wykorzystaniem obiektów 99

Wytyczne dotyczące projektowania 99

Wykonanie odpowiedniej analizy 103

Określanie zakresu planowanych prac 103

Gromadzenie wymagań 103

Tworzenie prototypu systemu	104
Identyfikowanie klas	104
Definiowanie wymagań wobec każdej z klas	104
Określenie warunków współpracy między klasami	104
Tworzenie modelu klas opisującego system	104
Tworzenie prototypu interfejsu użytkownika za pomocą kodu	105
Obiekty opakowujące	105
Kod strukturalny	106
Opakowywanie kodu strukturalnego	107
Opakowywanie nieprzenośnego kodu	108
Opakowywanie istniejących klas	109
Podsumowanie	110
Źródła	110
Rozdział 7. Dziedziczenie i kompozycja	111
Wielokrotne wykorzystywanie obiektów	111
Dziedziczenie	112
Generalizacja i specjalizacja	115
Decyzje projektowe	116
Kompozycja	118
Reprezentowanie kompozycji na diagramach UML	118
Czemu hermetyzacja jest podstawą technologii obiektowej	120
Jak dziedziczenie osłabia hermetyzację	121
Szczegółowy przykład wykorzystania polimorfizmu	123
Odpowiedzialność obiektów	123
Klasy abstrakcyjne, metody wirtualne i protokoły	126
Podsumowanie	127
Źródła	128

Rozdział 8. Wielokrotne wykorzystanie kodu - interfejsy i klasy abstrakcyjne 129

Wielokrotne wykorzystanie kodu 129

Infrastruktura programistyczna 130

Co to jest kontrakt 132

Klasy abstrakcyjne 133

Interfejsy 135

Wnioski 137

Dowód kompilatora 139

Zawieranie kontraktu 140

Punkty dostępowe do systemu 142

Przykład biznesu elektronicznego 142

Biznes elektroniczny 142

Podjęcie niezakładające wielokrotnego wykorzystania kodu 143

Rozwiązanie dla aplikacji biznesu elektronicznego 145

Model obiektowy UML 145

Podsumowanie 148

Źródła 148

Rozdział 9. Tworzenie obiektów i projektowanie obiektowe 149

Relacje kompozycji 149

Podział procesu budowy na etapy 151

Rodzaje kompozycji 153

Agregacja 153

Asocjacja 153

Łączne wykorzystanie asocjacji i agregacji 155

Unikanie zależności 155

Liczność 156

Kilka asocjacji 158

Asocjacje opcjonalne 159

Praktyczny przykład 160

Podsumowanie 161

Źródła 161

Rozdział 10. Wzorce projektowe 163

Historia wzorców projektowych 164

Wzorzec MVC języka Smalltalk 165

Rodzaje wzorców projektowych 166

Wzorce konstrukcyjne 167

Wzorzec Metoda Fabryczna 168

Wzorce strukturalne 170

Wzorce czynnościowe 172

Antywzorce 174

Podsumowanie 175

Źródła 175

Rozdział 11. Jak uniknąć zależności i silnych powiązań między klasami 177

Kompozycja a dziedziczenie i wstrzykiwanie zależności 179

1. Dziedziczenie 179

2. Kompozycja 181

Wstrzykiwanie zależności 183

Wstrzykiwanie przez konstruktor 185

Wstrzykiwanie za pomocą metody ustawiającej 185

Podsumowanie 186

Źródła 186

Rozdział 12. Zasady SOLID projektowania obiektowego 187

Zasady SOLID projektowania obiektowego 188

1. Zasada jednej odpowiedzialności - SRP 188

2. Zasada "otwarty/zamknięty" - OCP 191

3. Zasada podstawiania Liskov - LSP 193

4. Zasada segregacji interfejsów - IPS 195

5. Zasada odwrócenia zależności - DIP 196

Podsumowanie 201

Źródła 202