

Spis treści

Słowo wstępne.....	17
Przedmowa.....	21
Podziękowania.....	31
O autorach.....	33
Część I Wytwarzanie zwinne.....	35
Rozdział 1. Praktyki programowania zwinnego.....	37
Agile Alliance.....	38
Programiści i ich harmonijna współpraca jest ważniejsza od procesów i narzędzi.....	39
Działające oprogramowanie jest ważniejsze od wyczerpującej dokumentacji.....	40
Faktyczna współpraca z klientem jest ważniejsza od negocjacji zasad kontraktu.....	41
Reagowanie na zmiany jest ważniejsze od konsekwentnego realizowania planu.....	42
Podstawowe zasady.....	43
Konkluzja.....	46
Bibliografia.....	47
Rozdział 2. Przegląd technik programowania ekstremalnego.....	49
Praktyki programowania ekstremalnego.....	50
Cały zespół.....	50
Opowieści użytkownika.....	50

Krótkie cykle.....	51
Testy akceptacyjne.....	52
Programowanie w parach.....	53
Wytwarzanie sterowane testami (TDD).....	54
Wspólna własność.....	54
Ciągła integracja.....	55
Równe tempo.....	56
Otwarta przestrzeń pracy.....	56
Gra planistyczna.....	57
Prosty projekt.....	57
Refaktoryzacja.....	59
Metafora.....	59
Konkluzja.....	61
Bibliografia.....	61
Rozdział 3. Planowanie.....	63
Wstępne poznawanie wymagań.....	64
Dzielenie i scalanie opowieści użytkownika.....	65
Planowanie wydań.....	66
Planowanie iteracji.....	66
Definiowanie warunków zakończenia projektu.....	67
Planowanie zadań.....	67
Iteracje.....	69
Śledzenie postępu.....	69
Konkluzja.....	70
Bibliografia.....	71
Rozdział 4. Testowanie.....	73
Wytwarzanie sterowane testami.....	74
Przykład projektu poprzedzonego testami.....	75
Izolacja testów.....	76
Eliminowanie powiązań.....	78
Testy akceptacyjne.....	79
Wpływ testów akceptacyjnych	
na architekturę oprogramowania.....	81

Spis treści	5
Konkluzja	82
Bibliografia	82
Rozdział 5. Refaktoryzacja	83
Prosty przykład refaktoryzacji	
— generowanie liczb pierwszych	84
Testy jednostkowe	86
Refaktoryzacja	87
Ostatnie udoskonalenia	93
Konkluzja	97
Bibliografia	98
Rozdział 6. Epizod z życia programistów	99
Gra w kręgle	100
Konkluzja	146
Przegląd reguł gry w kręgle	147
Część II Projektowanie zwinne	149
Rozdział 7. Czym jest projektowanie zwinne?	153
Symptomy złego projektu	154
Symptomy złego projektu, czyli potencjalne źródła porażek	154
Sztywność	155
Wrażliwość	155
Nieelastyczność	156
Niedostosowanie do rzeczywistości	156
Nadmierna złożoność	156
Niepotrzebne powtórzenia	157
Nieprzejrzystość	157
Dlaczego oprogramowanie ulega degradacji	158
Program Copy	159
Przykład typowego scenariusza	159
Przykład budowy programu Copy w ramach projektu zwinnego	163
Konkluzja	166
Bibliografia	166

Rozdział 8. Zasada pojedynczej odpowiedzialności	167
Definiowanie odpowiedzialności	170
Oddzielanie wzajemnie powiązanych odpowiedzialności	171
Trwałość	171
Konkluzja	172
Bibliografia	172
Rozdział 9. Zasada otwarte-zamknięte	173
Omówienie zasady otwarte-zamknięte	174
Aplikacja Shape	177
Przykład naruszenia zasady OCP	177
Przykład pełnej zgodności z zasadą otwarte-zamknięte	180
Przewidywanie zmian i „naturalna” struktura	181
Przygotowywanie punktów zaczepienia	182
Stosowanie abstrakcji do jawnego zamykania oprogramowania dla zmian	184
Zapewnianie zamknięcia z wykorzystaniem techniki sterowania przez dane	185
Konkluzja	187
Bibliografia	187
Rozdział 10. Zasada podstawiania Liskov	189
Naruszenia zasady podstawiania Liskov	190
Prosty przykład	190
Przykład mniej jaskrawego naruszenia zasady LSP	192
Przykład zaczerpnięty z rzeczywistości	199
Wyodrębnianie zamiast dziedziczenia	205
Heurystyki i konwencje	208
Konkluzja	209
Bibliografia	209
Rozdział 11. Zasada odwracania zależności	211
Podział na warstwy	212
Odwracanie relacji własności	213
Zależność od abstrakcji	215

Prosty przykład praktycznego znaczenia zasady DIP	216
Odkrywanie niezbędnych abstrakcji.....	217
Przykład aplikacji Furnace	219
Konkluzja	221
Bibliografia	221
 Rozdział 12. Zasada segregacji interfejsów	223
Zanieczyszczanie interfejsów.....	223
Odrębne klasy klienckie oznaczają odrębne interfejsy.....	225
Interfejsy klas kontra interfejsy obiektów.....	227
Separacja przez delegację.....	227
Separacja przez wielokrotne dziedziczenie.....	229
Przykład interfejsu użytkownika bankomatu.....	230
Konkluzja.....	237
Bibliografia.....	237
 Rozdział 13. Przegląd języka UML	
dla programistów C#	239
Diagramy klas.....	243
Diagramy obiektów.....	244
Diagramy sekwencji.....	245
Diagramy współpracy.....	246
Diagramy stanów.....	246
Konkluzja.....	247
Bibliografia.....	247
 Rozdział 14. Praca z diagramami	249
Po co modelować oprogramowanie?	249
Po co budować modele oprogramowania ?.....	250
Czy powinniśmy pracować nad rozbudowanymi projektami przed przystąpieniem do kodowania?.....	251
Efektywne korzystanie z diagramów języka UML	251
Komunikacja z innymi programistami.....	252
Mapy drogowe.....	253
Dokumentacja wewnętrzna.....	255
Co powinniśmy zachowywać, a co można wyrzucać do kosza?.....	255

Iteracyjne udoskonalanie	257
Najpierw zachowania	257
Weryfikacja struktury	259
Wyobrażenie o kodzie	261
Ewolucja diagramów	262
Kiedy i jak rysować diagramy	264
Kiedy przystępować do tworzenia diagramów, a kiedy rezygnować z dalszego rysowania ich	264
Narzędzia CASE	265
A co z dokumentacją?	266
Konkluzja	267
 Rozdział 15. Diagramy stanów	269
Wprowadzenie	270
Zdarzenia specjalne	271
Superstany	272
Pseudostan początkowy i końcowy	274
Stosowanie diagramów skończonych maszyn stanów	274
Konkluzja	276
 Rozdział 16. Diagramy obiektów	277
Migawka	278
Obiekty aktywne	279
Konkluzja	283
 Rozdział 17. Przypadki użycia	285
Pisanie przypadków użycia	286
Przebiegi alternatywne	287
Co jeszcze?	288
Prezentowanie przypadków użycia na diagramach	288
Konkluzja	290
Bibliografia	290
 Rozdział 18. Diagramy sekwencji	291
Wprowadzenie	292
Obiekty, linie życia, komunikaty i inne konstrukcje	292
Tworzenie i niszczenie obiektów	293

Proste pętle.....	295
Przypadki i scenariusze.....	295
Pojęcia zaawansowane.....	298
Pętle i warunki.....	298
Komunikaty, których przesyłanie wymaga czasu.....	300
Komunikaty asynchroniczne.....	302
Wiele wątków.....	307
Obiekty aktywne.....	308
Wysyłanie komunikatów do interfejsów.....	309
Konkluzja.....	310
Rozdział 19. Diagramy klas.....	311
Wprowadzenie.....	312
Klasy.....	312
Asocjacje.....	313
Relacje dziedziczenia.....	314
Przykładowy diagram klas.....	315
Omówienie szczegółowe.....	318
Stereotypy klas.....	318
Klasy abstrakcyjne.....	319
Właściwości.....	320
Agregacja.....	321
Kompozycja.....	322
Liczność.....	323
Stereotypy asocjacji.....	324
Klasy zagnieżdżone.....	326
Klasy asocjacji.....	326
Kwalifikatory asocjacji.....	327
Konkluzja.....	328
Bibliografia.....	328
Rozdział 20. Heurystyki i kawa.....	329
Ekspres do kawy Mark IV Special.....	330
Specyfikacja.....	330
Popularne, ale niewłaściwe rozwiązanie.....	333
Nieprzemyślana abstrakcja.....	336
Poprawione rozwiązanie.....	337

Implementacja modelu abstrakcyjnego.....	343
Zalety projektu w tej formie.....	358
Implementacja projektu obiektowego.....	366
Bibliografia.....	366
 Część III Studium przypadku listy płac.....	 367
Uproszczona specyfikacja systemu listy płac.....	368
Ćwiczenie.....	369
Przypadek użycia nr 1 — dodanie danych nowego pracownika.....	369
Przypadek użycia nr 2 — usunięcie danych pracownika.....	370
Przypadek użycia nr 3 — wysłanie karty czasu pracy.....	370
Przypadek użycia nr 4 — wysłanie raportu o sprzedaży.....	370
Przypadek użycia nr 5 — wysłanie informacji o opłacie na rzecz związku zawodowego.....	371
Przypadek użycia nr 6 — zmiana szczegółowych danych pracownika	371
Przypadek użycia nr 7 — wygenerowanie listy płatności na dany dzień ..	372
 Rozdział 21. Wzorce projektowe Command	
i Active Object — uniwersalność	
i wielozadaniowość.....	373
Proste polecenia.....	374
Transakcje.....	377
Fizyczny podział kodu.....	378
Czasowy ⁷ podział kodu.....	379
Metoda Undo.....	379
Wzorzec projektowy Active Object.....	380
Konkluzja.....	386
Bibliografia.....	386
 Rozdział 22. Wzorce projektowe Template Method	
i Strategy — dziedziczenie	
kontra delegacja.....	387
Wzorzec projektowy Template Method.....	388
Błędne zastosowanie wzorca Template Method.....	392
Sortowanie bąbelkowe.....	392

Wzorzec projektowy Strategy.....	396
Konkluzja.....	402
Bibliografia.....	402
Rozdział 23. Wzorce projektowe Facade i Mediator.....	403
Wzorzec projektowy Facade.....	404
Wzorzec projektowy Mediator.....	405
Konkluzja.....	407
Bibliografia.....	408
Rozdział 24. Wzorce projektowe	
Singleton i Monostate.....	409
Wzorzec projektowy Singleton.....	410
Zalety.....	412
Wady.....	412
Wzorzec Singleton w praktyce.....	413
Wzorzec projektowy Monostate.....	415
Zalety.....	417
Wady.....	417
Wzorzec Monostate w praktyce.....	417
Konkluzja.....	423
Bibliografia.....	423
Rozdział 25. Wzorzec projektowy Null Object.....	425
Omówienie.....	425
Konkluzja.....	429
Bibliografia.....	429
Rozdział 26. Przypadek użycia listy płac	
— pierwsza iteracja.....	431
Uproszczona specyfikacja.....	432
Analiza przez omówienie przypadku użycia.....	433
Dodanie danych nowego pracownika.....	434
Usunięcie danych pracownika.....	436
Wysłanie karty czasu pracy.....	436
Wysłanie raportu o sprzedaży.....	437

Wysłanie informacji o opłacie na rzecz związku zawodowego.....	438
Zmiana szczegółowych danych pracownika.....	439
Wygenerowanie listy płac na dany dzień.....	441
Refleksja — identyfikacja abstrakcji.....	443
Wynagrodzenia wypłacane pracownikom.....	444
Harmonogram wypłat.....	444
Formy wypłat.....	446
Przynależność związkowa.....	446
Konkluzja.....	447
Bibliografia.....	447
 Rozdział 27. Przypadek użycia listy płac	
— implementacja.....	449
Transakcje.....	450
Dodawanie danych pracowników.....	450
Usuwanie danych pracowników.....	456
Karty czasu pracy, raporty o sprzedaży	
i składki na związki zawodowe.....	458
Zmiana danych pracowników.....	466
Co ja najlepszego zrobiłem?.....	477
Wynagradzanie pracowników.....	480
Wynagradzanie pracowników etatowych.....	483
Wynagradzanie pracowników	
zatrudnionych w systemie godzinowym.....	486
Program główny.....	498
Baza danych.....	499
Konkluzja.....	500
O tym rozdziale.....	501
Bibliografia.....	502
 Część IV Pakowanie systemu płacowego.....	503
 Rozdział 28. Zasady projektowania pakietów	
i komponentów.....	505
Pakiety i komponenty.....	506
Zasady spójności komponentów — ziarnistość.....	507

Zasada równoważności wielokrotnego użycia i wydawania (REP)...	507
Zasada zbiorowego wielokrotnego stosowania (CRP).....	509
Zasada zbiorowego zamykania (CCP).....	510
Podsumowanie problemu spójności komponentów.....	511
Zasady spójności komponentów — stabilność	511
Zasada acyklicznych zależności (ADP).....	511
Zasada stabilnych zależności (SDP).....	519
Zasada stabilnych abstrakcji (SAP).....	525
Konkluzja	530
Rozdział 29. Wzorzec projektowy Factory	531
Problem zależności.....	534
Statyczna kontra dynamiczna kontrola typów.....	536
Fabryki wymienne.....	537
Wykorzystywanie fabryk do celów testowych.....	538
Znaczenie fabryk.....	540
Konkluzja	540
Bibliografia.....	540
Rozdział 30. Studium przypadku systemu płacowego	
— analiza podziału na pakiety.....	541
Notacja i struktura komponentów.....	542
Stosowanie zasady zbiorowego zamykania (CCP).....	544
Stosowanie zasady równoważności wielokrotnego użycia i wydawania (REP).....	546
Wiązanie komponentów i hermetyzacja.....	549
Mierniki.....	551
Stosowanie mierników dla aplikacji płacowej.....	553
Fabryki obiektów.....	556
Przebudowa granic spójności	558
Ostateczna struktura pakietów.....	559
Konkluzja	561
Bibliografia.....	561

Rozdział 31. Wzorzec projektowy Composite	563
Polecenia kompozytowe	565
Liczność albo brak licznosci	566
Konkluzja	566
Rozdział 32. Wzorzec projektowy Observer	
— ewolucja kodu w kierunku wzorca	567
Zegar cyfrowy	568
Wzorzec projektowy Observer	589
Modele	590
Zarządzanie zasadami projektowania obiektowego	591
Konkluzja	592
Bibliografia	593
Rozdział 33. Wzorce projektowe Abstract Server,	
Adapter i Bridge	595
Wzorzec projektowy Abstract Server	596
Wzorzec projektowy Adapter	598
Forma klasowa wzorca Adapter	599
Problem modemu — adaptery i zasada LSP	599
Wzorzec projektowy Bridge	604
Konkluzja	607
Bibliografia	607
Rozdział 34. Wzorce projektowe Proxy i Gateway	
— zarządzanie cudzymi interfejsami API	609
Wzorzec projektowy Proxy	610
Implementacja wzorca Proxy	615
Podsumowanie	630
Bazy danych, oprogramowanie pośredniczące	
i inne gotowe interfejsy	631
Wzorzec projektowy Table Data Gateway	634
Testowanie konstrukcji TDG w pamięci	642
Test bram DB	643

Stosowanie pozostałych wzorców projektowych łącznie	
z bazami danych	646
Konkluzja	648
Bibliografia	648
Rozdział 35. Wzorzec projektowy Visitor	649
Wzorzec projektowy Visitor	650
Wzorzec projektowy Acyclic Visitor	654
Zastosowania wzorca Visitor.....	660
Wzorzec projektowy Decorator	668
Wzorzec projektowy Extension Object	674
Konkluzja	686
Bibliografia	686
Rozdział 36. Wzorzec projektowy State	687
Zagnieżdżone wyrażenia switch-case	688
Wewnętrzny zasięg zmiennej stanu.....	691
Testowanie akcji.....	692
Zalety i wady.....	692
Tabele przejść	693
Interpretacja tabeli przejść.....	694
Zalety i wady.....	695
Wzorzec projektowy State	696
Wzorzec State kontra wzorzec Strategy.....	699
Zalety i wady.....	699
Kompilator maszyny stanów (SMC).....	700
Plik <u>Turnstile.es</u> wygenerowany przez kompilator SMC	
i pozostałe pliki pomocnicze.....	703
Zastosowania skończonej maszyny stanów	709
Wysokopoziomowa polityka działania	
graficznych interfejsów użytkownika (GUI).....	709
Sterowanie interakcją z interfejsem GUI.....	711
Przetwarzanie rozproszone.....	712
Konkluzja	713
Bibliografia	714

Rozdział 37. Studium przypadku systemu płacowego	
— baza danych.....	715
Budowa bazy danych.....	716
Słaby punkt dotychczasowego projektu.....	716
Dodawanie danych nowych pracowników.....	719
Transakcje.....	732
Odczytywanie danych o pracownikach.....	738
Co jeszcze zostało do zrobienia?.....	753
Rozdział 38. Interfejs użytkownika systemu płacowego	
— wzorzec Model View Presenter.....	755
Interfejs.....	758
Implementacja.....	759
Budowa okna.....	770
Okno główne systemu płacowego.....	778
Pierwsza odsłona.....	791
Konkluzja.....	792
Bibliografia.....	792
Dodatek A Satyra na dwa przedsiębiorstwa.....	793
Rufus Inc. — Project Kickoff.....	793
Rupert Industries — Project Alpha.....	793
Dodatek B Czym jest oprogramowanie?.....	811
Skorowidz.....	827