

Spis treści

Słowo wstępne (13)

Przedmowa (14)

O autorach (20)

CZĘŚĆ I. ZWINNE WYTWARZANIE OPROGRAMOWANIA (21)

Rozdział 1. Praktyki agile (23)

Agile Alliance (24)

Manifest Agile Alliance (24)

Zasady (27)

Wniosek (29)

Bibliografia (29)

Rozdział 2. Przegląd informacji o programowaniu ekstremalnym (31)

Praktyki programowania ekstremalnego (31)

Klient jest członkiem zespołu (32)

Historyjki użytkowników (32)

Krótkie cykle (32)

Testy akceptacyjne (33)

Programowanie parami (33)

Programowanie sterowane testami (34)

Wspólna własność (34)

Ciągła integracja (34)

Równomierne tempo (35)

Otwarta przestrzeń robocza (35)

Gra w planowanie (35)

Prosty projekt (36)

Refaktoryzacja (37)

Metafora (37)

Wniosek (38)

Bibliografia (38)

Rozdział 3. Planowanie (39)

Początkowa eksploracja (40)

Tworzenie prototypów, dzielenie i szybkość (40)

Planowanie wersji dystrybucyjnych (41)

Planowanie iteracji (41)

Planowanie zadań (41)

Półmetek (42)

Przebieg iteracji (42)

Wniosek (43)

Bibliografia (43)

Rozdział 4. Testowanie (45)

Programowanie sterowane testami (45)

Przykład projektu w stylu "najpierw test" (46)

Izolacja testu (47)

Nieoczekiwane wyeliminowanie sprzężeń (48)

Testy akceptacyjne (49)

Przykład testów akceptacyjnych (50)

Architektura "przy okazji" (51)

Wniosek (51)

Bibliografia (52)

Rozdział 5. Refaktoryzacja (53)

Generowanie liczb pierwszych - prosty przykład refaktoryzacji (54)

Ostateczny przegląd (59)

Wniosek (62)

Bibliografia (63)

Rozdział 6. Epizod programowania (65)

Gra w kręgle (66)

Wniosek (98)

CZĘŚĆ II. PROJEKT AGILE (101)

Symptomy złego projektu (101)

Zasady (101)

Zapachy a zasady (102)

Bibliografia (102)

Rozdział 7. Co to jest projekt agile? (103)

Co złego dzieje się z oprogramowaniem? (103)

Zapachy projektu - woń psującego się oprogramowania (104)

Co stymuluje oprogramowanie do psucia się? (106)

Zespoły agile nie pozwalają psuć się oprogramowaniu (106)

Program Copy (106)

Przykład programu Copy wykonanego zgodnie z metodyką agile (109)

Skąd deweloperzy agile wiedzieli, co należy zrobić? (110)

Utrzymywanie projektu w jak najlepszej postaci (110)

Wniosek (111)

Bibliografia (111)

Rozdział 8. SRP - zasada pojedynczej odpowiedzialności (113)

SRP - zasada pojedynczej odpowiedzialności (113)

Czym jest odpowiedzialność? (115)

Rozdzielanie sprzężonych odpowiedzialności (115)

Trwałość (116)

Wniosek (116)

Bibliografia (116)

Rozdział 9. OCP - zasada otwarte-zamknięte (117)

OCP - zasada otwarte-zamknięte (117)

Opis (118)

Kluczem jest abstrakcja (118)

Aplikacja Shape (119)

Naruszenie zasady OCP (120)

Zachowanie zgodności z zasadą OCP (121)

Przyznaję się. Kłamałem (122)

Przewidywanie i "naturalna" struktura (122)

Umieszczanie "haczyków" (123)

Stosowanie abstrakcji w celu uzyskania jawnego domknięcia (124)

Zastosowanie podejścia "sterowania danymi" w celu uzyskania domknięcia (125)

Wniosek (126)

Bibliografia (126)

Rozdział 10. LSP - zasada podstawiania Liskov (127)

LSP - zasada podstawiania Liskov (127)

Prosty przykład naruszenia zasady LSP (128)

Kwadraty i prostokąty - bardziej subtelne naruszenie zasady LSP (129)

Prawdziwy problem (131)

Poprawność nie jest wrodzona (132)

Relacja IS-A dotyczy zachowania (132)

Projektowanie według kontraktu (132)

Specyfikowanie kontraktów w testach jednostkowych (133)

Realny przykład (133)

Motywacja (133)

Problem (135)

Rozwiązanie niezgodne z zasadą LSP (136)

Rozwiązanie zgodne z zasadą LSP (136)

Wydzielanie zamiast dziedziczenia (137)

Heurystyki i konwencje (139)

Zdegenerowane funkcje w klasach pochodnych (140)

Zgłaszanie wyjątków z klas pochodnych (140)

Wniosek (140)

Bibliografia (140)

Rozdział 11. DIP - zasada odwracania zależności (141)

DIP - zasada odwracania zależności (141)

Podział na warstwy (142)

Odwrócenie własności (142)

Zależność od abstrakcji (143)

Prosty przykład (144)

Wyszukiwanie potrzebnych abstrakcji (145)

Przykład programu Furnace (146)

Polimorfizm dynamiczny i statyczny (147)

Wniosek (148)

Bibliografia (148)

Rozdział 12. ISP - zasada segregacji interfejsów (149)

Zaśmiecanie interfejsów (149)

Odrębne klienty oznaczają odrębne interfejsy (150)

Siła oddziaływania klientów na interfejsy (151)

ISP - zasada segregacji interfejsów (151)

Interfejsy klas a interfejsy obiektów (152)

Separacja przez delegację (152)

Separacja przez wielokrotne dziedziczenie (153)

Przykład interfejsu użytkownika bankomatu (153)

Poliady i monady (158)

Wniosek (159)

Bibliografia (159)

CZĘŚĆ III. STUDIUM PRZYPADKU: SYSTEM PŁACOWY (161)

Szczątkowa specyfikacja systemu płacowego (162)

Ćwiczenie (162)

Przypadek użycia nr 1: dodawanie nowego pracownika (162)

Przypadek użycia nr 2: usuwanie pracownika (163)

Przypadek użycia nr 3: dostarczenie karty pracy (163)

Przypadek użycia nr 4: dostarczenie raportu sprzedaży (163)

Przypadek użycia nr 5: dostarczenie informacji o opłacie na rzecz związku zawodowego (164)

Przypadek użycia nr 6: zmiana danych pracownika (164)

Przypadek użycia nr 7: wygenerowanie listy płac na dzień (164)

Rozdział 13. Wzorce projektowe Polecenie i Aktywny obiekt (165)

Proste polecenia (166)

Transakcje (167)

Fizyczny i czasowy podział kodu (168)

Czasowy podział kodu (168)

Metoda Undo (169)

Aktywny obiekt (169)

Wniosek (173)

Bibliografia (173)

Rozdział 14. Metoda szablonowa i Strategia: dziedziczenie a delegacja (175)

Metoda szablonowa (176)

Nadużywanie wzorca (178)

Sortowanie bąbelkowe (179)

Strategia (181)

Sortowanie jeszcze raz (183)

Wniosek (185)

Bibliografia (185)

Rozdział 15. Wzorce projektowe Fasada i Mediator (187)

Fasada (187)

Mediator (188)

Wniosek (190)

Bibliografia (190)

Rozdział 16. Wzorce projektowe Singleton i Monostate (191)

Singleton (192)

Korzyści ze stosowania wzorca Singleton (193)

Koszty stosowania wzorca Singleton (193)

Wzorzec projektowy Singleton w praktyce (193)

Monostate (194)

Korzyści ze stosowania wzorca Monostate (196)

Koszty stosowania wzorca Monostate (196)

Wzorzec projektowy Monostate w praktyce (196)

Wniosek (200)

Bibliografia (200)

Rozdział 17. Wzorzec projektowy Obiekt Null (201)

Wniosek (204)

Bibliografia (204)

Rozdział 18. Studium przypadku: system płacowy. Pierwsza iteracja (205)

Wprowadzenie (205)

Specyfikacja (206)

Analiza według przypadków użycia (206)

Dodawanie pracowników (207)

Usuwanie pracowników (208)

Dostarczenie karty pracy (209)

Dostarczenie raportów sprzedaży (209)

Dostarczenie informacji o opłacie na rzecz związku zawodowego (210)

Zmiana danych pracownika (210)

Wypłaty (212)

Refleksja: czego się nauczyliśmy? (214)

Wyszukiwanie potrzebnych abstrakcji (214)

Abstrakcja harmonogramu (214)

Sposoby wypłaty (215)

Przynależność do związków zawodowych (216)

Wniosek (216)

Bibliografia (216)

Rozdział 19. Studium przypadku: system płacowy. Implementacja (217)

Dodawanie pracowników (218)

Baza danych systemu płacowego (219)

Zastosowanie wzorca Metoda szablonowa do dodawania pracowników (220)

Usuwanie pracowników (223)

Zmienne globalne (225)

Karty pracy, raporty sprzedaży i składki (225)

Zmiana danych pracowników (231)

Zmiana klasyfikacji (235)

Co ja paliłem? (240)

Realizacja wypłat (244)

Czy chcemy, aby deweloperzy podejmowali decyzje biznesowe? (246)

Realizacja wypłat dla pracowników ze stałą pensją (246)

Realizacja wypłat dla pracowników zatrudnionych w systemie godzinowym (248)

Okresy rozliczeniowe: problem projektowy (251)

Program główny (257)

Baza danych (257)

Podsumowanie projektu systemu płacowego (258)

Historia (259)

Zasoby (259)

Bibliografia (259)

CZĘŚĆ IV. PODZIAŁ SYSTEMU PŁACOWEGO NA PAKIETY (261)

Rozdział 20. Zasady projektowania pakietów (263)

Projektowanie z wykorzystaniem pakietów? (263)

Ziarnistość: zasady spójności pakietów (264)

Zasada równoważności wielokrotnego wykorzystania kodu i dystrybucji (REP) (264)

Zasada zbiorowego wielokrotnego użytku (CRP) (265)

Zasada zbiorowego zamykania (CCP) (266)

Podsumowanie tematyki spójności pakietów (266)

Stabilność: zasady sprzęgania pakietów (267)

Zasada acyklicznych zależności (ADP) (267)

Cotygodniowe kompilacje (267)

Eliminowanie cykli zależności (268)

Skutki istnienia cykli w grafie zależności między pakietami (269)

Przerywanie cykli (270)

Odchylenia (270)

Projekt góra-dół (271)

Zasada stabilnych zależności (SDP) (272)

Stabilność (272)

Metryki stabilności (273)

Nie wszystkie pakiety muszą być stabilne (274)

Gdzie powinna się znaleźć implementacja projektu wysokiego poziomu? (276)

Zasada stabilnych abstrakcji (SAP) (276)

Mierzenie abstrakcji (276)

Ciąg główny (277)

Odległość od ciągu głównego (278)

Wniosek (280)

Rozdział 21. Wzorzec projektowy Fabryka (281)

Cykl zależności (283)

Fabryki wymienne (284)

Wykorzystanie wzorca Fabryka do tworzenia zestawów testowych (284)

Znaczenie korzystania z fabryk (286)

Wniosek (287)

Bibliografia (287)

Rozdział 22. Studium przypadku: system płacowy (część 2.) (289)

Struktura pakietów i notacja (290)

Zastosowanie zasady zbiorowego domykania (CCP) (291)

Zastosowanie zasady równoważności wielokrotnego wykorzystania kodu i dystrybucji (REP) (292)

Sprzężenia i hermetyzacja (294)

Metryki (296)

Zastosowanie wskaźników do aplikacji płacowej (297)

Fabryki obiektów (300)

Fabryka obiektów dla pakietu TransactionImplementation (300)

Inicjowanie fabryk (301)

Przebudowa granic spójności (301)

Ostateczna struktura pakietów (302)

Wniosek (304)

Bibliografia (304)

CZĘŚĆ V. STUDIUM PRZYPADKU: STACJA POGODOWA (305)

Rozdział 23. Wzorzec projektowy Kompozyt (307)

Przykład: polecenia kompozytowe (308)

Wielokrotność czy brak wielokrotności (309)

Rozdział 24. Obserwator - ewolucja kodu do wzorca (311)

Zegar cyfrowy (311)

Wniosek (326)

Wykorzystanie diagramów w tym rozdziale (327)

Wzorzec projektowy Obserwator (327)

Zarządzanie zasadami projektu obiektowego dla wzorca projektowego Obserwator (328)

Bibliografia (329)

Rozdział 25. Wzorce projektowe Serwer abstrakcyjny i Most (331)

Wzorzec projektowy Serwer abstrakcyjny (332)

Kto jest właścicielem interfejsu? (333)

Wzorzec projektowy Adapter (333)

Wzorzec projektowy Adapter w formie klasy (334)

Problem modemu. Adaptery i zasada LSP (334)

Wzorzec projektowy Most (338)

Wniosek (339)

Bibliografia (340)

Rozdział 26. Wzorce projektowe Pełnomocnik i Schody do nieba - zarządzanie zewnętrznymi interfejsami API (341)

Wzorzec projektowy Pełnomocnik (342)

Implementacja wzorca projektowego Pełnomocnik w aplikacji koszyka na zakupy (345)

Podsumowanie wiadomości o wzorcu projektowym Pełnomocnik (356)

Obsługa baz danych, oprogramowania middleware oraz zewnętrznych interfejsów API (357)

Schody do nieba (359)

Przykład zastosowania wzorca Schody do nieba (360)

Wniosek (365)

Inne wzorce projektowe, które można wykorzystywać z bazami danych (365)

Wniosek (366)

Bibliografia (366)

Rozdział 27. Analiza przypadku: stacja pogodowa (367)

Firma Chmura (367)

Oprogramowanie WMS-LC (369)

Wybór języka (369)

Projekt oprogramowania systemu Nimbus-LC (369)

Historia 24-godzinna i utrwalanie (382)

Implementacja algorytmów HiLo (384)

Wniosek (391)

Bibliografia (391)

Przegląd wymagań dla oprogramowania Nimbus-LC (391)

Wymagania użytkowe (391)

Historia 24-godzinna (392)

Konfiguracja użytkownika (392)

Wymagania administracyjne (392)

Przypadki użycia systemu Nimbus-LC (393)

Aktorzy (393)

Przypadki użycia (393)

Historia pomiarów (393)

Konfiguracja (393)

Administracja (393)

Plan publikacji wersji dystrybucyjnych systemu Nimbus-LC (394)

Wprowadzenie (394)

Wydanie I (394)

Zagrożenia (394)

Produkty projektu (395)

Wydanie II (395)

Zaimplementowane przypadki użycia (395)

Zagrożenia (395)

Produkty projektu (395)

Wydanie III (396)

Zaimplementowane przypadki użycia (396)

Zagrożenia (396)

Produkty projektu (396)

CZĘŚĆ VI. STUDIUM PRZYPADKU: ETS (397)

Rozdział 28. Wzorzec projektowy Wizytator (399)

Rodzina wzorców projektowych Wizytator (400)

Wizytator (400)

Wzorzec projektowy Wizytator działa jak macierz (403)

Wzorzec projektowy Acykliczny wizytator (403)

Wzorzec projektowy Wizytator działa jak macierz rzadka (407)

Wykorzystanie wzorca projektowego Wizytator w generatorach raportów (407)

Inne zastosowania wzorca projektowego Wizytator (412)

Wzorzec projektowy Dekorator (413)

Wiele dekoratorów (416)

Wzorzec projektowy Obiekt rozszerzenia (418)

Wniosek (426)

Przypomnienie (426)

Bibliografia (426)

Rozdział 29. Wzorzec projektowy Stan (427)

Przegląd informacji o automatach stanów skończonych (427)

Techniki implementacji (429)

Zagnieżdżone instrukcje Switch/Case (429)

Interpretacja tabeli przejść (432)

Wzorzec projektowy Stan (433)

SMC - kompilator maszyny stanów (436)

Kiedy należy korzystać z maszyn stanów? (439)

Wysokopoziomowe strategie obsługi GUI (439)

Kontrolery interakcji z GUI (440)

Przetwarzanie rozproszone (441)

Wniosek (441)

Listingi (441)

Implementacja klasy Turnstile.java z wykorzystaniem interpretacji tabeli przejść (441)

Klasa Turnstile.java wygenerowana przez kompilator SMC oraz inne pliki pomocnicze (443)

Bibliografia (447)

Rozdział 30. Framework ETS (449)

Wprowadzenie (449)

Przegląd informacji o projekcie (449)

Wczesny okres 1993 - 1994 (451)

Framework? (451)

Framework (452)

Zespół z roku 1994 (452)

Termin (452)

Strategia (452)

Wyniki (453)

Projekt frameworka (454)

Wspólne wymagania dla aplikacji oceniających (454)

Projekt frameworka do wyznaczania ocen (456)

Przypadek zastosowania wzorca Metoda szablonowa (459)

Napisać pętlę raz (460)

Wspólne wymagania dla aplikacji zdawania (463)

Projekt frameworka do zdawania (463)

Architektura menedżera zadań (469)

Wniosek (472)

Bibliografia (472)

Dodatek A. Notacja UML. Część I: Przykład CGI (473)

System rejestrowania kursów: opis problemu (474)

Aktorzy (475)

Przypadki użycia (475)

Model dziedziny (478)

Architektura (482)

Klasy abstrakcyjne i interfejsy na diagramach sekwencji (492)

Podsumowanie (494)

Bibliografia (494)

Dodatek B. Notacja UML. Część II: STATMUX (495)

Definicja statystycznego multipleksera (495)

Środowisko oprogramowania (496)

Ograniczenia czasu rzeczywistego (496)

Procedury obsługi przerwań wejścia (497)

Procedury obsługi przerwań wyjścia (501)

Protokoły komunikacji (502)

Wniosek (512)

Bibliografia (512)

Dodatek C. Satyra na dwa przedsiębiorstwa (513)

Rufus! Inc. Project Kickoff (513)

Rupert Industries Projekt Alpha (513)

Dodatek D. Kod źródłowy jest projektem (525)

Czym jest projekt oprogramowania? (525)

Skorowidz (535)