

O autorze (13)

Przedmowa (15)

Część I Procesy adaptacyjne (21)

Rozdział 1. Fundamenty adaptacyjne technologii J2EE (23)

- Potrzeba błyskawicznego wytwarzania oprogramowania (24)
- Wyzwania na poziomie przedsiębiorstw (25)
- Platforma J2EE (26)
- Definiowanie fundamentu adaptacyjnego (27)
 - o Dlaczego fundament? (27)
 - o Dlaczego adaptacyjny? (27)
- Przygotowywanie fundamentów pod błyskawiczne wytwarzanie oprogramowania (28)
 - o Ludzie (29)
 - o Narzędzia (29)
 - o Szkielety (29)
 - o Praktyki (30)
 - o Standardy (30)
 - o Procesy i procedury (30)
 - o Szkolenia (30)
 - o Ustawiczne doskonalenie (31)
- Korzyści wynikające z inwestycji w fundamenty adaptacyjne (32)
- Kluczowe czynniki decydujące o sukcesie (33)
 - o Pozyskiwanie przychylności projektantów i programistów (34)
 - o Edukacja (35)
 - o Wsparcie zarządu (36)
- Podsumowanie (37)
 - o Informacje dodatkowe (38)

Rozdział 2. Błyskawiczne wytwarzanie aplikacji (39)

- Wspólne elementy koncepcji RAD (40)
 - o Metoda skrzynki czasowej (40)
 - o Języki właściwe dla określonych dziedzin (41)
 - o Wielokrotne wykorzystywanie oprogramowania (42)
 - o Narzędzia zapewniające produktywność (44)
 - o Błyskawiczne tworzenie prototypów (44)
- Praca z prototypami (45)
 - o Typy prototypów i techniki ich budowy (46)
 - o Wybór pomiędzy trybem odrzucania a trybem ewoluowania (48)
- Podsumowanie (49)
 - o Informacje dodatkowe (49)

Rozdział 3. Korzystanie z metod adaptacyjnych (51)

- Po co w ogóle stosować metodyki? (52)
 - o Metodyka RAD dla platformy J2EE (52)
 - o Metody adaptacyjne kontra metody predyktywne (53)
- Kaskadowy model cyklu życia projektu (54)
 - o Klasyczny model kaskadowy (54)
 - o Zalety i wady (55)

- o Studium przypadku (56)
- Wytwarzanie iteracyjne (59)
 - o Zalety podejścia iteracyjnego (60)
 - o Procesy iteracyjne (63)
- Wprowadzenie do procesu RUP (63)
 - o Proces oparty na przypadkach użycia (65)
 - o Iteracyjne wytwarzanie oprogramowania w ramach procesu RUP (68)
 - o Etapy procesu RUP (69)
 - o Dziedziny (70)
 - o Elementy procesu RUP (72)
 - o Planowanie (73)
 - o Wsparcie dla projektów na poziomie korporacyjnym (75)
 - o Wady procesu RUP (75)
- Metody zwinne (76)
- Wprowadzenie do metodyki XP (77)
 - o Praktyki procesu XP (77)
 - o Planowanie (80)
 - o Projektowanie (81)
 - o Kodowanie (82)
 - o Testowanie (84)
 - o Role w procesie XP (84)
 - o Stosowanie metodyki XP dla projektów korporacyjnych opartych na platformie J2EE (86)
- Podsumowanie (88)
 - o Informacje dodatkowe (89)

Część II Architektury zwinne (91)

Rozdział 4. Projektowanie zapewniające błyskawiczność wytwarzania (93)

- Cele architektury i projektów (94)
- Architektura i projekt w koncepcji RAD (96)
 - o Wykorzystanie mocnych stron zespołu projektowego (96)
 - o Wykorzystaj najlepsze dostępne szkielety (97)
 - o Myśl o przyszłości (100)
 - o Wystrzegaj się projektowania na rzecz wielokrotnego wykorzystywania kodu (102)
 - o Stosowanie projektów prostopadłych (103)
 - o Stosuj architektury wielowarstwowe (104)
- Różne podejścia do architektury J2EE (106)
 - o Architektury dwuwarstwowe kontra architektury wielowarstwowe (106)
 - o Enterprise JavaBeans (108)
 - o Perspektywy klientów zdalnych i lokalnych (109)
 - o Rozproszone komponenty (110)
 - o Wybór właściwego projektu (113)
 - o Architektura zorientowana na interfejs WWW (113)
 - o Architektura zorientowana na komponenty EJB (116)
- Podsumowanie (118)
 - o Informacje dodatkowe (119)

Rozdział 5. Oprogramowanie modelujące (121)

- Po co w ogóle modelować? (122)
 - o Komunikacja (122)
 - o Weryfikacja poprawności (124)
- Perspektywy architektury (125)
- Zunifikowany język modelowania (UML) (126)
 - o Diagramy przypadków użycia (127)
 - o Diagramy aktywności (129)
 - o Diagramy klas (130)
 - o Diagramy interakcji (132)
 - o Diagramy stanów (134)
 - o Diagramy wdrożenia i diagramy komponentów (135)
- Najczęstsze błędy (135)
 - o Oprogramowanie tworzone w warunkach kultu cargo (137)
- Narzędzia modelujące (137)
 - o Wybór narzędzia modelującego (138)
 - o Obsługa języka UML (139)
 - o Sprawdzanie poprawności modelu (140)
 - o Inżynieria normalna i inżynieria odwrotna (140)
 - o Obsługa wzorców projektowych (142)
- Dlaczego narzędzia projektowe nie zdają egzaminu? (145)
 - o Syndrom szczeniaka pod choinką (145)
- Metody skutecznego korzystania z narzędzi modelujących (148)
- Podsumowanie (150)
 - o Informacje dodatkowe (150)

Rozdział 6. Generowanie kodu (153)

- Czym właściwie jest generowanie kodu? (154)
- Pasywne generatory kodu (155)
 - o Generowanie kodu za pomocą narzędzia Apache Velocity (156)
 - o Zalety pasywnego generowania kodu (159)
- Aktywne generatory kodu (160)
 - o Zalety aktywnego generowania kodu (160)
 - o Aktywne generatory kodu i wzorce kodu (162)
- Programowanie atrybutowe (163)
 - o Czym są atrybuty? (163)
 - o Atrybuty kontra dyrektywy preprocesora (164)
 - o Przypisy platformy J2SE 5.0 kontra atrybuty (165)
- Wprowadzenie do narzędzia XDoclet (166)
 - o Instalacja narzędzia XDoclet (167)
 - o Przygotowywanie pliku kompilacji narzędzia Ant (167)
 - o Utworzenie komponentu sesyjnego (171)
- Praca z aktywnie generowanym kodem (177)
 - o Wskazówki dotyczące generowania kodu (177)
- Podsumowanie (181)
 - o Informacje dodatkowe (181)

Rozdział 7. Błyskawiczność a bazy danych (183)

- Problem baz danych (184)

- o Dane są cenną wartością korporacji (184)
 - o Problem odwzorowania obiektowo-relacyjnego (187)
- Możliwości w zakresie dostępu do danych (190)
 - o Interfejs Java Database Connectivity (JDBC) (191)
 - o Narzędzia do odwzorowywania obiektowo-relacyjnego (192)
 - o Komponenty encyjne (194)
 - o Obiekty danych Javy (196)
- Generowanie kodu i odwzorowywanie obiektowo-relacyjne (197)
 - o Wprowadzenie do narzędzia Hibernate (198)
 - o Wprowadzenie do narzędzia Middlegen (199)
- Przygotowanie bazy danych (200)
 - o Wprowadzenie do systemu MySQL (201)
 - o Utworzenie schematu bazy danych (201)
 - o Uruchamianie skryptu bazy danych (204)
- Generowanie warstwy utrwalania danych (204)
 - o Uruchamianie narzędzia Middlegen z poziomu programu Ant (206)
 - o Graficzny interfejs użytkownika programu Middlegen (207)
 - o Dokumenty odwzorowania obiektowo-relacyjnego narzędzia Hibernate (208)
 - o Od dokumentów odwzorowań do kodu źródłowego Javy (212)
 - o Dokończenie całego procesu (213)
- Podsumowanie (215)
 - o Informacje dodatkowe (216)

Rozdział 8. Architektura sterowana modelami (217)

- Założenia technologii MDA (217)
- Wyjaśnienie podstawowych założeń koncepcji architektury sterowanej modelami (219)
 - o Platforma (220)
 - o Modele (220)
 - o Odwzorowanie (223)
- Architektura MDA kontra tradycyjne techniki modelowania (224)
 - o Zalety (225)
 - o Wady (227)
- Narzędzia zgodne z architekturą MDA (229)
 - o Wprowadzenie do narzędzia AndroMDA (230)
- Architektura MDA i program AndroMDA (230)
 - o Wymiana modeli reprezentowanych w formacie XMI (232)
 - o Znaczniki modelu PIM (233)
 - o Wymienne wkłady MDA (236)
 - o Budowa wymiennego wkładu (238)
 - o Szablony wymiennych wkładów MDA (240)
 - o AndroMDA w praktyce (242)
 - o AndroMDA 3.0 (244)
- Podsumowanie (244)
 - o Informacje dodatkowe (245)

Część III Języki błyskawicznego wytwarzania oprogramowania (247)

Rozdział 9. Skrypty (249)

- Po co w ogóle używać języków skryptowych? (249)
- Cechy języków skryptowych (251)
 - o Doświadczenie zespołu projektowego (251)
 - o Wieloplatformowość (252)
 - o Integracja z klasami Javy (252)
- Wprowadzenie do języka Jython (253)
 - o Instalacja Jythona (255)
 - o Uruchamianie skryptów (255)
 - o Język Jython (256)
 - o Integracja z językiem Java (258)
 - o Tworzenie prototypów interfejsu użytkownika (261)
 - o Tworzenie serwletów w języku Jython (262)
- Alternatywa - język Groovy (264)
- Podsumowanie (265)
 - o Informacje dodatkowe (265)

Rozdział 10. Praca z regułami (267)

- Reguły biznesowe (267)
 - o Czym jest reguła biznesowa? (268)
 - o Struktura reguły biznesowej (268)
 - o Dynamiczna natura reguł biznesowych (269)
- Reguły biznesowe w oprogramowaniu (269)
 - o Reguły definiowane z góry (270)
 - o Język definiowania reguł (270)
 - o Ścisłe wiązanie logiki systemowej z logiką biznesową (270)
 - o Powielanie reguł (271)
- Mechanizmy regułowe (271)
 - o Systemy regułowe (272)
 - o Mechanizmy regułowe w systemach korporacyjnych (273)
- Wprowadzenie do języka Jess (274)
 - o Instalacja Jessa (275)
 - o Przykład użycia języka Jess (275)
 - o Jess i Java (279)
- Interfejs API mechanizmu regułowego Javy (282)
- Mechanizmy regułowe stosowane na poziomie korporacyjnym (283)
 - o Cechy korporacyjnych mechanizmów regułowych (284)
 - o Kryteria oceny mechanizmów regułowych (286)
- Podsumowanie (287)
 - o Informacje dodatkowe (288)

Rozdział 11. Programowanie aspektowe (289)

- Dlaczego programowanie aspektowe? (290)
 - o Obszary przecinające (290)
 - o Mieszanie i rozpraszanie kodu (291)
 - o Tradycyjne podejścia do problemu obszarów przecinających (292)
- Wyjaśnienie koncepcji programowania aspektowego (295)
 - o Pojęcia i terminologia (295)
 - o Garbaci i smoki (297)

- o Metody tkania (297)
- Wprowadzenie do narzędzia AspectJ (298)
 - o AspectJ i Eclipse (299)
 - o Kompilator języka AspectJ (299)
 - o Przykład użycia języka AspectJ (300)
- Język kontra szkielet (306)
 - o Implementacje szkieletu programowania aspektowego (307)
- Wprowadzenie do szkieletu AspectWerkz (308)
 - o Definicja aspektu w formacie XML (309)
 - o Aspekty w postaci przypisów metadanych (310)
 - o Opcje tkania w szkielecie AspectWerkz (311)
 - o Aspektowe oprogramowanie pośredniczące (312)
- Wdrażanie aspektów (313)
 - o Aspekty wytwarzania (313)
 - o Aspekty produkcyjne (314)
 - o Programowanie aspektowe w zestawieniu z innymi koncepcjami wytwarzania oprogramowania (315)
- Podsumowanie (316)
 - o Informacje dodatkowe (317)

Część IV Środowiska dynamiczne (319)

Rozdział 12. Optymalne kompilowanie systemów (321)

- Czas i ruch (322)
 - o Linia produkcji oprogramowania (322)
 - o Czas i ruch w kontekście wytwarzania oprogramowania (323)
- Proces kompilacji (323)
 - o Projektowanie procesu kompilacji (324)
 - o Wymagania platformy J2EE w zakresie kompilacji (325)
 - o Czym jest wdrażanie "na gorąco"? (327)
- Wprowadzenie do narzędzia Ant (328)
- Kompilacje minimalne w programie Ant (330)
 - o Znaczenie zależności kompilacji (330)
 - o Definiowanie zależności kompilacji w plikach Anta (333)
- Praca z podprojektami (336)
- Przeglądanie zależności kompilacji (337)
- Standardowe zadania kompilacji (339)
- Organizacja projektu (341)
 - o Katalog źródłowy (source) (341)
 - o Katalog bibliotek (lib) (343)
 - o Katalog plików skompilowanych (build) (343)
 - o Katalog plików gotowych do wdrożenia (344)
- Integracja ze środowiskami IDE (345)
- Rozszerzanie Anta o obsługę języka Jython (347)
 - o Tworzenie nowego zadania Anta (347)
 - o Kompilowanie klas Jython (348)
 - o Testowanie nowego zadania (349)
- Podsumowanie (350)
 - o Informacje dodatkowe (350)

Rozdział 13. Zintegrowane środowisko wytwarzania (353)

- Po co używać zintegrowanego środowiska wytwarzania? (354)
 - Podstawowe funkcje środowisk IDE (354)
- Wprowadzenie do platformy Eclipse (358)
 - Czym jest Eclipse? (358)
 - Instalacja i uruchamianie środowiska Eclipse (359)
 - Obszar roboczy platformy Eclipse (360)
 - Platforma Eclipse jako warsztat pracy programisty (360)
 - Rozszerzanie warsztatu pracy za pomocą dodatkowych modułów (362)
- Funkcjonalność środowisk IDE w kontekście wytwarzania oprogramowania korporacyjnego (364)
 - Kreatory kodu (365)
 - Obsługa edycji wielu typów plików (367)
 - Integracja z narzędziem Ant (369)
 - Praca z generatorami kodu (370)
 - Sterowanie pracą serwera i wdrażanie aplikacji (372)
 - Obsługa technik modelowania (374)
 - Dostęp do bazy danych (374)
- Diagnozowanie aplikacji J2EE w ramach platformy Eclipse (375)
 - Architektura programu uruchomieniowego platformy Javy (JPDA) (376)
 - Diagnozowanie aplikacji J2EE (378)
 - Wymiana "na gorąco" (380)
 - Diagnozowanie stron JSP (382)
 - Wskazówki dotyczące diagnozowania aplikacji (382)
- Podsumowanie (385)
 - Informacje dodatkowe (385)

Rozdział 14. Wytwarzanie sterowane testami (387)

- Testowanie jako część procesu wytwarzania (388)
 - Zalety wytwarzania sterowanego testami (389)
 - Koszt wytwarzania sterowanego testami (390)
- Wprowadzenie do narzędzia JUnit (391)
 - Wykonywanie testów szkieletu JUnit w środowisku Eclipse (395)
- Generowanie testów jednostkowych (397)
 - Generowanie testów jednostkowych w środowisku Eclipse (397)
 - Testy jednostkowe i architektura MDA (400)
 - Generowanie przypadków testowych (401)
- Testowanie "od podszewki" (403)
 - Czym jest obiekt pozorny? (404)
 - Praca z obiektami pozornymi (406)
 - Rodzaje obiektów pozornych (406)
 - Ratunek w postaci dynamicznych obiektów pozornych (407)
 - Wybór pomiędzy statycznymi a dynamicznymi obiektami pozornymi (411)
- Podsumowanie (412)
 - Informacje dodatkowe (413)

Rozdział 15. Efektywne zapewnianie jakości (415)

- Zapewnianie jakości (416)
 - o Środowisko projektowe (417)
 - o Proces testowania (418)
 - o Testowanie projektów RAD (419)
- Automatyzacja testów (420)
 - o Testowanie rozwiązań J2EE (422)
 - o Narzędzia automatyzujące testy (423)
- Testy funkcjonalne (425)
- Wprowadzenie do narzędzia HttpUnit (426)
 - o HttpUnit i JUnit (427)
 - o Pisanie testów z wykorzystaniem interfejsu HttpUnit API (427)
- Testy obciążeniowe i wytrzymałościowe (431)
 - o Problematyka wydajności (432)
- Wprowadzenie do narzędzia JMeter (432)
 - o Testowanie aplikacji MedRec za pomocą szkieletu JMeter (433)
 - o Tworzenie grupy wątków (435)
 - o Element konfiguracyjny (436)
 - o Kontrolery logiczne (437)
 - o Elementy próbkujące (439)
 - o Elementy nasłuchujące (442)
 - o Wykonywanie planu testu (443)
 - o Analiza wyników (443)
- Wskazówki dotyczące korzystania ze szkieletu JMeter (445)
- Podsumowanie (447)
 - o Informacje dodatkowe (448)

Dodatki (449)

Dodatek A Akronimy (451)

Dodatek B Bibliografia (455)

Skorowidz (459)