

Spis treści

Wstęp (15)

CZĘŚĆ I WPROWADZENIE (19)

Rozdział 1. Przegląd technologii (21)

Najważniejsze cechy platformy Java EE 6 (22)

Model aplikacji Javy EE (23)

Rozproszone, wielowarstwowe aplikacje (23)

Bezpieczeństwo (25)

Komponenty Javy EE (25)

Klienty Javy EE (25)

Komponenty webowe (27)

Komponenty biznesowe (28)

Warstwa danych (29)

Kontenery Javy EE (29)

Usługi kontenera (30)

Typy kontenerów (30)

Obsługa usług sieciowych (31)

XML (32)

Protokół transportowy SOAP (32)

Standard formatu WSDL (33)

Budowanie i wdrażanie aplikacji Javy EE (33)

Pakowanie aplikacji (33)

Role w procesie wytwarzania aplikacji (35)

Dostawca oprogramowania Java EE (36)

- Dostawca narzędzi (36)
- Dostawca komponentów aplikacji (36)
- Budowniczy aplikacji (37)
- Wdrożeniowiec oraz administrator (37)
- API Javy EE 6 (37)
 - Technologia Enterprise JavaBeans (39)
 - Technologia Java Servlet (40)
 - Technologia JavaServer Faces (41)
 - Technologia JavaServer Pages (42)
 - Biblioteka JavaServer Pages Standard Tag Library (42)
 - Java Persistence API (42)
 - Java Transaction API (43)
 - API Javy dla usług sieciowych typu REST (43)
 - Managed Beans (43)
 - Contexts and Dependency Injection for the Java EE Platform (JSR 299) (44)
 - Dependency Injection for Java (JSR 330) (44)
 - Bean Validation (44)
 - Java Message Service API (44)
 - Architektura Java EE Connector (45)
 - JavaMail API (45)
 - Java Authorization Contract for Containers (45)
 - Java Authentication Service Provider Interface for Containers (46)
- API Javy EE 6 wchodzące w skład platformy Java Platform, Standard Edition 6 i 7 (46)
 - Java Database Connectivity API (46)
 - Java Naming and Directory Interface API (47)
 - JavaBeans Activation Framework (47)
 - Java API for XML Processing (48)

Java Architecture for XML Binding (48)

SOAP with Attachments API for Java (48)

Java API for XML Web Services (48)

Java Authentication And Authorization Service (49)

Narzędzia serwera GlassFish (49)

Rozdział 2. Używanie przykładowych aplikacji z samouczka (51)

Wymagane oprogramowanie (51)

Java Platform, Standard Edition (51)

Java EE 6 Software Development Kit (52)

Komponent samouczka Javy EE 6 (53)

NetBeans IDE (53)

Apache Ant (54)

Uruchamianie i zatrzymywanie serwera GlassFish (55)

Uruchamianie serwera GlassFish z poziomu środowiska NetBeans IDE (56)

Uruchamianie konsoli administracyjnej (56)

Uruchamianie konsoli administracyjnej w środowisku NetBeans IDE (56)

Uruchamianie i zatrzymywanie serwera Java DB (56)

Uruchamianie serwera bazy danych przy użyciu środowiska NetBeans IDE (57)

Budowanie przykładowych aplikacji (57)

Struktura katalogów z przykładami samouczka (57)

Pobieranie najnowszych aktualizacji samouczka (58)

Aktualizacja samouczka za pomocą centrum aktualizacji (58)

Debugowanie aplikacji Javy EE (59)

Używanie logów serwera (59)

Używanie debuggera (59)

CZĘŚĆ II WARSTWA WEBOWA (61)

Rozdział 3. Technologia JavaServer Faces - tematy zaawansowane (63)

Cykl życia aplikacji JavaServer Faces (63)

Omówienie cyklu życia JavaServer Faces (64)

Faza przywrócenia widoku (66)

Faza zastosowania wartości zapytania (67)

Faza przetwarzania walidacji (68)

Faza aktualizacji wartości modelu (68)

Faza wywołania aplikacji (69)

Faza renderowania odpowiedzi (69)

Częściowe przetwarzanie i rendering (70)

Cykl życia aplikacji faceletów (70)

Model komponentów interfejsu użytkownika (71)

Klasy komponentów interfejsu użytkownika (71)

Model renderingu komponentów (73)

Model konwersji (74)

Model zdarzeń i procesów ich obsługi (75)

Model walidacji (77)

Model nawigacji (78)

Rozdział 4. Wykorzystanie technologii Ajax wraz z technologią JavaServer Faces (81)

Technologia Ajax - wprowadzenie (82)

Wykorzystanie technologii Ajax wraz z technologią JavaServer Faces (83)

Wykorzystanie technologii Ajax wraz z faceletami (83)

Użycie znacznika f:ajax (84)

Wysłanie żądania Ajax (86)

Użycie atrybutu event (86)

Użycie atrybutu execute (86)

Użycie atrybutu immediate (87)

Użycie atrybutu listener (87)

Monitorowanie zdarzeń po stronie klienta (87)

Obsługa błędów (88)

Otrzymywanie odpowiedzi Ajax (89)

Cykl życia żądania Ajax (90)

Grupowanie komponentów (90)

Wczytywanie kodu JavaScript jako zasobu (91)

Użycie API dla kodu JavaScript w aplikacji z faceletami (91)

Użycie adnotacji @ResourceDependency w klasie ziarna (92)

Przykładowa aplikacja ajaxguessnumber (93)

Pliki źródłowe (93)

Uruchomienie przykładu ajaxguessnumber (95)

Dodatkowe informacje na temat użycia technologii Ajax wraz z technologią JavaServer Faces (96)

Rozdział 5. Komponenty złożone - tematy zaawansowane i przykłady (97)

Atrybuty komponentu złożonego (97)

Wywoływanie zarządzanego ziarna (98)

Walidacja wartości komponentu złożonego (98)

Przykładowa aplikacja compositecomponentlogin (99)

Plik komponentu złożonego (99)

Strona wykorzystująca komponent (100)

Zarządzane ziarno (100)

Uruchomienie przykładu compositecomponentlogin (101)

Rozdział 6. Tworzenie własnych komponentów UI i innych obiektów (103)

Określanie, czy potrzebny jest własny komponent lub renderer (105)

Kiedy użyć własnego komponentu? (105)

Kiedy zastosować własny renderer? (107)

Kombinacje komponentów, rendererów i znaczników (107)

Analiza przykładu z mapą obrazu (108)

Dlaczego mam korzystać z technologii JavaServer Faces do implementacji mapy obrazu? (109)

Działanie zrenderowanego kodu HTML (109)

Omówienie strony faceletu (110)

Konfiguracja danych modelu (111)

Podsumowanie klas mapy obrazu (113)

Kroki niezbędne do utworzenia własnego komponentu (113)

Tworzenie własnych klas komponentów (114)

Określenie rodziny komponentu (117)

Przeprowadzenie kodowania (117)

Przeprowadzenie dekodowania (119)

Umożliwienie właściwościom komponentu przyjmowania wyrażeń (120)

Zapis i przywracanie stanu (121)

Przekazanie renderowania do renderera (122)

Tworzenie klasy renderera (123)

Określenie rodzaju renderera (124)

Implementacja klasy nasłuchiwanie zdarzeń (125)

Implementacja klasy nasłuchiwanie zdarzeń zmiany wartości (125)

Implementacja klas nasłuchujących akcji (126)

Obsługa zdarzeń dla samodzielnie wykonanych komponentów (127)

Definicja znacznika własnego komponentu w deskrypcji biblioteki znaczników (128)

Użycie własnego komponentu (129)

Utworzenie i użycie własnego konwertera (130)

Tworzenie własnego konwertera (131)

Użycie własnego konwertera (133)

Utworzenie i użycie własnego walidatora (135)

Implementacja interfejsu Validator (136)

Określanie własnego znacznika (138)

Użycie własnego walidatora (138)

Wiązanie wartości i instancji komponentów z właściwościami zarządzanego ziarna (140)

Powiązanie wartości komponentu z właściwością (141)

Powiązanie wartości komponentu z niejawnym obiektem (142)

Powiązanie instancji komponentu z właściwością ziarna (144)

Wiązanie konwerterów, walidatorów i obsługi zdarzeń z właściwościami zarządzanego ziarna (145)

Rozdział 7. Konfiguracja aplikacji JavaServer Faces (147)

Wykorzystanie adnotacji do konfiguracji zarządzanych ziaren (148)

Korzystanie z zasięgów zarządzanych ziaren (148)

Plik zasobu konfiguracji aplikacji (149)

Kolejność plików zasobów konfiguracji aplikacji (151)

Konfiguracja zarządzanych ziaren (152)

Użycie elementu managed-bean (153)

Inicjalizacja właściwości przy użyciu elementu managed-property (155)

Inicjalizacja odwzorowań i list (160)

Rejestracja komunikatów aplikacji (160)

Użycie FacesMessage do utworzenia komunikatu (162)

Dostęp do komunikatów o błędach (162)

Korzystanie z walidatorów domyślnych (163)

Rejestracja własnego walidatora (164)

Rejestracja własnego konwertera (164)

Konfiguracja reguł nawigacyjnych (165)

 Konfiguracja reguły nawigacyjnej (167)

 Niejawne reguły nawigacyjne (168)

Rejestracja własnego renderera w zestawie rendererów (168)

Rejestracja własnego komponentu (170)

Podstawowe wymagania stawiane aplikacjom JavaServer Faces (171)

 Konfiguracja aplikacji przy użyciu deskryptora wdrożenia (172)

 Konfiguracja etapu projektu (175)

 Dołączanie klas, stron i innych zasobów (176)

Rozdział 8. Przesył plików do serwera w technologii Java Servlet (177)

 Adnotacja @MultipartConfig (177)

 Metody getParts i getPart (178)

 Przykładowa aplikacja fileupload (179)

 Architektura przykładowej aplikacji (179)

 Uruchomienie przykładu fileupload (182)

Rozdział 9. Umiędzynarodowienie i lokalizacja aplikacji webowych (185)

 Klasy umiędzynarodowienia platformy Javy (185)

 Lokalizacja komunikatów i etykiet (186)

 Określanie dostępnych języków i regionów (187)

 Określenie paczki zasobów (187)

 Pobranie komunikatów w odpowiednim języku (188)

 Formatowanie dat i czasu (189)

 Kodowanie i zestawy znaków (189)

 Zestawy znaków (189)

Kodowanie znaków (190)

CZĘŚĆ III USŁUGI SIECIOWE (193)

Rozdział 10. JAX-RS - tematy zaawansowane oraz przykład (195)

Adnotacje dla pól i właściwości ziarna w klasach zasobów (195)

Wydobycie parametrów ścieżki (196)

Wydobycie parametrów zapytania (197)

Wydobycie danych formularza (197)

Wydobycie typu Javy dla żądania lub odpowiedzi (198)

Podzasoby i dobór zasobów w trakcie działania usługi (198)

Metody podzasobów (199)

Lokalizator podzasobu (199)

Integracja JAX-RS z technologią EJB i CDI (200)

Warunkowe żądania HTTP (201)

Negocjacja treści w trakcie działania usługi (202)

Użycie JAX-RS z JAXB (204)

Wykorzystanie obiektów Javy do modelowania własnych danych (205)

Rozpoczynanie pracy od definicji schematu XML (207)

Użycie formatu JSON wraz z JAX-RS i JAXB (209)

Przykładowa aplikacja customer (210)

Omówienie elementów przykładowej aplikacji (210)

Klasy encji Customer i Address (210)

Klasa CustomerService (213)

Klasy CustomerClientXML i CustomerClientJSON (215)

Modyfikacja przykładu, by generował klasy encji z istniejącego schematu (217)

Uruchomienie przykładu customer (219)

CZĘŚĆ IV KOMPONENTY EJB (225)

Rozdział 11. Przykład ziarna sterowanego komunikatami (227)

Omówienie przykładu simplemessage (227)

Klient aplikacji simplemessage (228)

Klasa ziarna sterowanego komunikatami (228)

Metoda onMessage (229)

Uruchomienie przykładu simplemessage (231)

Obiekty administrowane związane z przykładem simplemessage (231)

Uruchomienie przykładu simplemessage w środowisku NetBeans IDE (231)

Uruchomienie przykładu simplemessage przy użyciu narzędzia Ant (232)

Usunięcie administrowanych obiektów z przykładu simplemessage (233)

Rozdział 12. Korzystanie z osadzonego kontenera komponentów EJB (235)

Omówienie osadzonego kontenera komponentów EJB (235)

Tworzenie aplikacji z EJB z możliwością osadzania (235)

Uruchamianie osadzonych aplikacji (236)

Tworzenie kontenera EJB (236)

Wyszukiwanie referencji do ziaren sesyjnych (237)

Wyłączanie kontenera EJB (238)

Przykładowa aplikacja standalone (238)

Uruchomienie przykładowej aplikacji standalone (239)

Rozdział 13. Wywoływanie metod asynchronicznych ziaren sesyjnych (241)

Wywoływanie metod asynchronicznych (241)

Tworzenie asynchronicznych metod biznesowych (242)

Wywoływanie metod asynchronicznych z poziomu klientów EJB (243)

Przykładowa aplikacja async (244)

Architektura przykładowej aplikacji async (244)

Uruchomienie przykładu async (245)

CZĘŚĆ V KONTEKSTY ORAZ WSTRZYKIWANIE ZALEŻNOŚCI NA PLATFORMIE JAVA EE (249)

Rozdział 14. Konteksty oraz wstrzykiwanie zależności na platformie Java EE - tematy zaawansowane (251)

Wykorzystanie alternatyw w aplikacjach CDI (251)

Specjalizacje (252)

Wykorzystanie w aplikacjach CDI metod produkujących, pól produkujących i metod usuwających (253)

Użycie metody produkującej (254)

Użycie pól produkujących do generowania zasobów (255)

Metody usuwające (255)

Użycie predefiniowanych ziaren w aplikacjach CDI (256)

Wykorzystanie zdarzeń w aplikacjach CDI (257)

Definiowanie zdarzeń (257)

Użycie metod obserwatorów do obsługi zdarzeń (257)

Zgłaszanie zdarzeń (258)

Użycie interceptorów w aplikacjach CDI (259)

Użycie dekoratorów w aplikacjach CDI (261)

Użycie stereotypów w aplikacjach CDI (262)

Rozdział 15. Uruchamianie zaawansowanych przykładów dotyczących kontekstów i wstrzykiwania zależności (265)

Przykład encoder - użycie alternatyw (265)

Interfejs Coder i jego implementacje (266)

Strona faceletu i zarządzane ziarno (266)

Uruchomienie przykładowej aplikacji encoder (268)

Przykład producermethods - użycie metody produkującej do wyboru implementacji ziarna (270)

Składniki przykładu producermethods (270)

Uruchomienie przykładowej aplikacji producermethods (272)

Przykład producerfields - użycie pól produkujących do generowania zasobów (273)

Pole produkujące w przykładzie producerfields (273)

Ziarno sesyjne i encja producerfields (274)

Zarządzane ziarno i strony faceletów (276)

Uruchomienie przykładowej aplikacji producerfields (277)

Przykład billpayment - zdarzenia i interceptory (279)

Klasa zdarzenia PaymentEvent (279)

Klasa nasłuchiwanie zdarzeń PaymentHandler (280)

Strony faceletów i zarządzane ziarno przykładu billpayment (280)

Klasa interceptora LoggedInterceptor (283)

Uruchomienie przykładowej aplikacji billpayment (284)

Przykład decorators - dekorowanie ziarna (285)

Elementy przykładu decorators (285)

Uruchomienie przykładowej aplikacji decorators (286)

CZĘŚĆ VI JAVA PERSISTENCE API (289)

Rozdział 16. Tworzenie i używanie tekstowych zapytań z kryteriami (291)

Wprowadzenie do zapytań tekstowych w Criteria API (291)

Tworzenie zapytań tekstowych (292)

Wykonywanie zapytań tekstowych (293)

Rozdział 17. Sterowanie współbieżnym dostępem do danych encji przy użyciu blokad (295)

Omówienie blokowania encji i współbieżności (295)

Użycie blokad optymistycznych (296)

Tryby blokad (297)

Ustawienie trybu blokady (298)

Użycie blokad pesymistycznych (298)

Rozdział 18. Wykorzystanie pamięci cache drugiego poziomu w aplikacjach Java Persistence API (301)

Wprowadzenie do pamięci cache drugiego poziomu (301)

Określanie, czy encje można umieścić w cache (302)

Określanie ustawień trybu cache w celu poprawy wydajności (303)

Ustawienie trybu pobierania i zapisu w cache (303)

Sterowanie cache drugiego poziomu w sposób programowy (305)

CZĘŚĆ VII BEZPIECZEŃSTWO (307)

Rozdział 19. Bezpieczeństwo w Javie EE - tematy zaawansowane (309)

Korzystanie z certyfikatów cyfrowych (309)

Tworzenie certyfikatu serwera (310)

Dodanie użytkowników do dziedziny certyfikatu (313)

Użycie innego certyfikatu serwera w serwerze GlassFish (313)

Mechanizmy uwierzytelniania (314)

Uwierzytelnianie klienta (314)

Uwierzytelnianie wzajemne (314)

Logowanie przy użyciu formularza w aplikacjach JavaServer Faces (318)

Użycie `j_security_check` w formularzach JavaServer Faces (318)

Użycie zarządzanego ziarna do uwierzytelniania w aplikacjach JavaServer Faces (319)

Uwierzytelnianie za pomocą JDBC Realm (321)

Konfiguracja dziedziny uwierzytelniania JDBC (321)

Zabezpieczanie zasobów HTTP (325)

Zabezpieczenie klientów aplikacji (328)

Użycie modułów logowania (328)

Użycie logowania programowego (329)

Zabezpieczanie aplikacji typu EIS (329)

Logowanie zarządzane przez kontener (330)

Logowanie zarządzane przez komponent (330)

Konfiguracja bezpieczeństwa adapterów zasobów (331)

Odwzorowanie zarządcy aplikacji na zarządcę EIS (332)

Konfiguracja bezpieczeństwa przy użyciu deskryptorów wdrożenia (333)

Określanie w deskrytorze wdrożenia zasad bezpieczeństwa dla prostego uwierzytelniania (333)

Wskazanie w deskrytorze wdrożenia niedomyślnego odwzorowania zarządzającego na rolę (334)

Dalsze informacje na temat bezpieczeństwa (334)

CZĘŚĆ VIII TECHNOLOGIE WSPOMAGAJĄCE JAVĘ EE (335)

Rozdział 20. Zagadnienia technologii JMS (337)

Wprowadzenie do JMS API (337)

Czym jest przekazywanie komunikatów? (337)

Czym jest JMS API? (338)

Kiedy mogę użyć JMS API? (338)

W jaki sposób JMS API współpracuje z platformą Javy EE? (339)

Podstawowe koncepcje JMS API (340)

Architektura JMS API (340)

Dziedziny komunikatów (341)
Konsumpcja komunikatów (343)
Model programistyczny JMS API (344)
Obiekty administracyjne JMS (344)
Połączenia JMS (347)
Sesje JMS (347)
Producenci komunikatów JMS (348)
Konsumenty komunikatów JMS (349)
Komunikaty JMS (351)
Przeglądarki kolejek JMS (353)
Obsługa wyjątków JMS (353)
Tworzenie wydajnych aplikacji JMS (354)
Użycie prostych mechanizmów trwałości (355)
Użycie zaawansowanych mechanizmów niezawodności (359)
Wykorzystanie JMS API w aplikacjach Javy EE (363)
Użycie adnotacji @Resource w komponentach webowych i EJB (364)
Użycie ziaren sesyjnych do produkcji i synchronicznego otrzymywania komunikatów (364)
Wykorzystanie ziaren sterowanych komunikatami do asynchronicznego otrzymywania komunikatów (365)
Zarządzanie transakcjami rozproszonymi (368)
Korzystanie z JMS API w klientach aplikacji i komponentach webowych (370)
Dodatkowe informacje na temat JMS (370)
 Rozdział 21. Przykłady wykorzystania JMS (371)
Tworzenie prostych aplikacji JMS (371)
Prosty przykład synchronicznego otrzymywania komunikatów (372)
Prosty przykład asynchronicznego otrzymywania komunikatów (381)
Prosty przykład przeglądania komunikatów w kolejce (386)

- Uruchomienie klientów JMS na wielu systemach (391)
- Usunięcie wdrożenia i wyczyszczenie systemu po przykładach JMS (397)
- Tworzenie rozbudowanych aplikacji JMS (397)
 - Przykład potwierdzania komunikatu (397)
 - Przykład trwałej subskrypcji (400)
 - Przykład transakcji lokalnej (402)
- Aplikacja stosująca JMS API w połączeniu z ziarnem sesyjnym (407)
 - Tworzenie komponentów aplikacji dla przykładu clientsessionmdb (407)
 - Tworzenie zasobów dla przykładu clientsessionmdb (410)
 - Uruchomienie przykładu clientsessionmdb (410)
- Aplikacja stosująca JMS API w połączeniu z encją (411)
 - Opis przykładowej aplikacji clientmdbentity (412)
 - Tworzenie komponentów przykładowej aplikacji clientmdbentity (413)
 - Tworzenie zasobów dla przykładu clientmdbentity (415)
 - Uruchomienie przykładu clientmdbentity (416)
- Przykład aplikacji konsumującej komunikaty z zewnętrznego serwera (419)
 - Opis modułów przykładu consumeremote (419)
 - Tworzenie komponentów modułu dla przykładu consumeremote (420)
 - Tworzenie zasobów dla przykładu consumeremote (421)
 - Użycie dwóch serwerów aplikacji dla przykładu consumeremote (421)
 - Uruchomienie przykładu consumeremote (421)
- Przykład aplikacji wdrażającej ziarno sterowane komunikatami na dwóch serwerach (425)
 - Opis modułów przykładu sendremote (425)
 - Tworzenie komponentów modułów dla przykładu sendremote (426)
 - Tworzenie zasobów dla przykładu sendremote (428)
 - Uruchomienie wdrożenia na zdalnym serwerze (428)
 - Użycie dwóch serwerów aplikacji dla przykładu sendremote (429)

Uruchomienie przykładu sendremote (429)

Uruchomienie przykładu sendremote w środowisku NetBeans IDE (430)

Uruchomienie przykładu sendremote przy użyciu narzędzia Ant (432)

Wyłączenie wdrażania na zdalnym systemie (434)

Rozdział 22. Mechanizm Bean Validation - tematy zaawansowane (435)

Tworzenie własnych ograniczeń (435)

Użycie wbudowanych ograniczeń do tworzenia własnych ograniczeń (435)

Dostosowywanie komunikatów walidatorów (436)

Paczka zasobów ValidationMessages (436)

Grupowanie ograniczeń (437)

Dostosowanie kolejności walidacji grup (437)

Rozdział 23. Wykorzystanie interceptorów Javy EE (439)

Wprowadzenie do interceptorów (439)

Klasy interceptorowe (440)

Cykl życia interceptorów (440)

Interceptorzy i CDI (441)

Użycie interceptorów (441)

Wywołania metod przechwytyjących (441)

Przechwytywanie zdarzeń wywołań zwrotnych cyklu życia (443)

Przechwytywanie zdarzeń upływu czasu (444)

Przykładowa aplikacja interceptor (445)

Uruchomienie przykładu interceptor (446)

Rozdział 24. Przykład z adapterem zasobów (449)

Adapter zasobów (449)

Ziarno sterowane komunikatami (450)

Aplikacja webowa (450)

Uruchomienie przykładu mailconnector (450)

Przygotowania przed wdrożeniem przykładu mailconnector (450)

Zbudowanie, spakowanie i wdrożenie przykładu mailconnector w środowisku NetBeans IDE (451)

Zbudowanie, spakowanie i wdrożenie przykładu mailconnector przy użyciu narzędzia Ant (452)

Uruchomienie przykładu mailconnector (452)

CZĘŚĆ IX PRZYKŁADOWE SCENARIUSZE UŻYCIA (453)

Rozdział 25. Przykładowy scenariusz użycia - księgarnia Duke'a (455)

Projekt i architektura księgarni Duke'a (455)

Interfejs aplikacji (456)

Encja Book korzystająca z Java Persistence API (456)

Komponenty EJB wykorzystywane w przykładzie (457)

Strony faceletów i zarządzane ziarna używane w przykładowej aplikacji (457)

Własne komponenty i inne własne obiekty wykorzystywane w księgarni Duke'a (459)

Pliki właściwości używane przez aplikację (459)

Deskryptory wdrożeń użyte w przykładowej aplikacji (460)

Uruchomienie aplikacji księgarnia Duke'a (461)

Budowanie i wdrażanie aplikacji w środowisku NetBeans IDE (461)

Budowanie i wdrażanie aplikacji przy użyciu narzędzia Ant (461)

Uruchomienie aplikacji (462)

Rozdział 26. Przykładowy scenariusz użycia - ćwiczenia Duke'a (463)

Projekt i architektura aplikacji (463)

Interfejs główny (465)

Encje Java Persistence API wykorzystywane przez interfejs główny (465)

Komponenty EJB wykorzystywane w interfejsie głównym (466)

Pliki faceletów wykorzystywane w interfejsie głównym (466)

Klasy pomocnicze używane w interfejsie głównym (467)

Pliki właściwości (468)

Deskryptory wdrożenia wykorzystywane przez aplikację (469)

Interfejs administracyjny (469)

Komponenty EJB wykorzystywane przez interfejs administracyjny (469)

Pliki faceletów wykorzystywane przez interfejs administracyjny (470)

Uruchomienie przykładowej aplikacji ćwiczenia Duke'a (470)

Konfiguracja serwera GlassFish (470)

Uruchomienie aplikacji ćwiczenia Duke'a (471)

Rozdział 27. Przykładowy scenariusz użycia - las Duke'a (475)

Projekt i architektura aplikacji las Duke'a (476)

Projekt events (478)

Projekt entities (479)

Projekt dukes-payment (481)

Projekt dukes-resources (481)

Projekt sklepu Duke'a (481)

Projekt wysyłka Duke'a (486)

Budowanie i wdrażanie aplikacji las Duke'a (488)

Zadanie przygotowawcze (489)

Zbudowanie i wdrożenie aplikacji las Duke'a w środowisku NetBeans IDE (489)

Zbudowanie i wdrożenie aplikacji las Duke'a przy użyciu narzędzia Ant (490)

Uruchamianie aplikacji las Duke'a (491)

Rejestracja jako klient sklepu (491)

Zakup produktów (491)

Zatwierdzenie wysyłki produktu (492)

Utworzenie nowego produktu (492)

Skorowidz (495)