

Program kursu Java Developer PRO

WPROWADZENIE / ŚRODOWISKO PROGRAMISTYCZNE

- › Praca z systemem kontroli wersji Git
- › Zarządzanie zależnościami i automatyzacja zadań na poziomie projektu z Maven i/lub Gradle
- › Konteneryzacja z wykorzystaniem Docker/Podman
- › CI/CD z użyciem Jenkins

WYBRANE ZAGADNIENIA Z ZAKRESU PROGRAMOWANIA W JAVIE

- › Typy generyczne
- › Przegląd nowości z ostatnich wydań Javy
- › Wprowadzenie do wielowątkowości
- › Klasyczna wielowątkowość (java.util.concurrent)
- › Nowoczesne podejście do wielowątkowości (Project Loom)

CZYSTY KOD I REFAKTORYZACJA

- › Czym jest czysty kod?
- › Zasady tworzenia jakościowego i testowalnego kodu (SOLID, GRASP, SRP ...)
- › Przegląd wzorców GoF (wyjaśnienie, diagramy, przykłady w kodzie)
- › Czym jest refaktoryzacja i jak ją bezpiecznie przeprowadzić?
- › Techniki refaktoryzacji
- › Refaktoryzacja do wzorców
- › Praca z kodem legacy

ARCHITEKTURA

- › Czym jest architektura i jakie ma znaczenie?
- › Spójność, niskie sprzężenie oraz programowanie z użyciem kontraktów
- › Wprowadzenie do Domain Driven Design
- › odział aplikacji na moduły i ich wzajemna separacja
- › Architektura warstwowa w praktyce
- › Czysta architektura w praktyce

TESTOWANIE

- › Dlaczego testowanie jest ważne?
- › Poziomy testowania i piramida testów
- › Metryki pokrycia testami
- › Testy jednostkowe - definicja i zakres odpowiedzialności
- › Cechy dobrych testów jednostkowych
- › Tworzenie testów jednostkowych z wykorzystaniem narzędzia JUnit
- › Izolacja zależności z użyciem obiektów zastępczych oraz Mockito
- › Test Driven Development
- › Testy integracyjne - definicja i zakres odpowiedzialności
- › Cechy dobrych testów integracyjnych
- › Behavior-driven development / Specification by example
- › Testy funkcjonalne i obciążeniowe na przykładzie JMeter i/lub Gatling

STANDARD JPA I HIBERNATE

- › Model obiektowy vs. relacyjny
- › Wykorzystanie relacyjnych baz danych w języku Java
- › Standard Java Persistence API i jego implementacje
- › Obiekty trwałe/encje - tworzenie, cykl życia
- › Kontekst utrwalania i menadżer encji
- › Mapowanie encji
- › Zarządzanie relacjami i propagacja kaskadowa
- › Klucze główne i strategie ich generowania
- › Wykorzystanie kolekcji i map
- › Obiekty osadzone
- › Mapowanie typów niestandardowych
- › Odwzorowywanie dziedziczenia

PRACA Z OBIEKTAMI TRWAŁYMI

- › Wykonywanie podstawowych operacji z użyciem EntityManager
- › Strategie ładowania danych (Lazy/Eager, EntityGraphs)
- › Wywołania zwrotne i klasy nasłuchujące
- › Zarządzanie transakcjami
- › Bezpieczeństwo w środowisku wieloużytkownikowym - blokady optymistyczne i pesymistyczne
- › Walidacja encji z użyciem BeansValidation

- › Wykonywanie zapytań
- › Tworzenie i uruchamianie zapytań
- › Język JPA QL/HQL (składnia, operatory, klauzule, wyrażenia funkcjonalne, złączenia)
- › Criteria API
- › Zapytania natywne
- › Wykorzystanie pamięci cache drugiego poziomu

QUARKUS FRAMEWORK

- › Charakterystyka frameworku
- › Przygotowanie środowiska deweloperskiego i tworzenie projektu
- › Wstrzykiwanie zależności i separacja odpowiedzialności z użyciem CDI
- › Konfigurowanie połączenia do bazy danych
- › Integracja z JPA / Hibernate
- › Budowanie usług REST opartych o JAX-RS / RestEasy
- › Publikowanie dokumentacji z OpenAPI
- › Bezpieczeństwo - konfiguracja uwierzytelniania i autoryzacji dostępu
- › Testowanie

WPROWADZENIE DO SPRING

- › Charakterystyka frameworku
- › Najważniejsze wzorce i praktyki wykorzystywane w Spring
- › Tworzenie i konfigurowanie projektu
- › Praca z dokumentacją
- › Idea inwersji kontroli (IoC) oraz wstrzykiwanie zależności
- › Działanie i odpowiedzialność kontenera
- › Konfiguracja z użyciem adnotacji oraz JavaConfig
- › Komponenty zarządzane – tworzenie, definiowanie zasięgu oraz zależności, cykl życia
- › Podstawy programowania aspektowego
- › Tworzenie, konfigurowanie i użycie aspektów
- › Event bus i programowanie przez zdarzenia
- › Konfigurowanie połączenia do bazy danych
- › Menadżer transakcji - zasada działania, dostępne implementacje
- › Parametry transakcji - propagacja, reguły wycofywania, poziom izolacji, czas ważności
- › Konfigurowanie mechanizmu transakcyjnego
- › Integracja z JPA i frameworkiem Hibernate
- › Tworzenie warstwy utrwalania z użyciem Spring Data

TWORZENIE APLIKACJI OPARTYCH O SPRING

- Wprowadzenie do Spring MVC
- Tworzenie i konfigurowanie kontrolerów
- Obsługa wyjątków
- Obiekty transferowe i mapowanie między warstwami
- Dokumentacja API z wykorzystaniem SpringDoc/OpenAPI
- Thymeleaf - tworzenie widoków, obsługa formularzy, walidacja, internacjonalizacja
- Spring Boot - konfiguracja i wdrażanie
- Wprowadzenie do Spring Security
- Uwierzytelnianie i autoryzacja dostępu na poziomie web i komponentów
- Bezpieczeństwo usług REST - tokeny JWT, OAuth2, Keycloak
- Testy jednostkowe na poziomie repozytoriów, usług i kontrolerów
- Testy integracyjne