

Comarch Bootcamp -Junior Java Developer

Zaawansowane elementy Javy

Cele szkolenia

Kurs ma na celu kompleksowe przygotowanie do roli Młodsze Programisty Java oraz podjęcia pracy w charakterze programisty. Podczas kursu uczestnik buduje w pełni funkcjonalne aplikacje, korzysta w stopniu zaawansowanym z narzędzi niezbędnych w codziennej pracy programisty, wybiera zaawansowane funkcje języka Java usprawniające produkowanie programów oraz korzysta z wielu technologii pozwalających na tworzenie profesjonalnych projektów komercyjnych dbając przy tym o wysoką jakość kodu.

Efektem kursu będzie przygotowanie samodzielne projektu o charakterze zaawansowanym.

Jest to kontynuacja kursu Comarch Bootcamp- Junior Java Developer I.

Aby wziąć udział w kursie należy posiadać wiedzę i umiejętności z zakresu Comarch Bootcamp- Junior Java Developer I.

Umiejętności

Dzięki szkoleniu uczestnik będzie:

- Tworzyć aplikacje webowe przy użyciu Spring
- Wykorzystywać wzorzec odwrócenia zależności (IoC)
- Tworzyć i posługiwać się relacyjnymi bazami danych
- Używać Hibernate oraz JDBC do operacji bazodanowych z poziomu Java
- Tworzyć testy automatyczne aplikacji Java
- Zarządzać swoją pracą w metodyce Agile
- Tworzyć i konsumować Web Services REST
- Tworzyć podstawowe diagramy UML
- Projektować nowoczesne aplikacje zgodne z aktualnymi standardami
- Znał podstawy programowania funkcyjnego i strumieniowego przetwarzania danych
- Potrafił wytwarzać aplikacje wielowątkowe

- Pracować w metodyce Scrum

Profil uczestników

Ta część kursu przeznaczona jest dla osób znających język Java, które chcą poznać bardziej zaawansowane narzędzia i techniki programowania. W szczególności Bootcamp dedykujemy wszystkim, którzy chcą podjąć pracę jako programista JAVA.

Przygotowanie uczestników

Aby wziąć udział w kursie należy posiadać wiedzę i umiejętności z zakresu Comarch Bootcamp- Junior Java Developer I.

Szczegółowy program szkolenia

Java Zaawansowana:

Zapoznamy się ze sposobami wielowątkowego i równoległego przetwarzania tych danych. Poza działającą aplikacją wymiernym efektem pracy programisty jest jakość kodu, który wytwarza -poznamy również mechanizmy pozwalające upraszczać, skracać i porządkować kod który piszemy.

- wielowątkowość
 - thread
 - runnable
 - executor'y wątków
 - pule wątków
 - synchronizacja wątków
- strumienie oraz wyrażenia lambda
- wartości opcjonalne w Java
- zarządzanie pamięcią –GC
- debugowanie aplikacji
-

Wykorzystanie AI w programowaniu:

Sztuczna inteligencja zmienia sposób tworzenia oprogramowania automatyzując i przyspieszając pracę programistów. Na przykładzie Copilot - pionierskiego i najpopularniejszego asystenta kodowania - nauczymy się wykorzystywać AI w codziennej pracy programisty.

- inteligentna generacja kodu
 - szybkie odpowiedzi w czasie rzeczywistym
 - generacja pełnych funkcjonalności i komponentów
- wykorzystanie interaktywnego asystenta AI
- generowanie dokumentacji kodu
- Code Explanation
- planowanie wytwarzania kodu i realizacja planu przy pomocy AI
- zarządzanie kontekstem
- wykorzystanie różnych modeli językowych
- techniki promptowania, tworzenie precyzyjnych promptów
- code review przy pomocy AI

Bazy danych:

Oprócz danych przechowywanych w pamięci aplikacji, musimy również przechowywać wiele danych "na stałe". Do tego wykorzystujemy bazy danych. Musimy zapoznać się z działaniem baz danych oraz ich obsługą. Nauczymy się również sposobów łączenia aplikacji napisanych w języku Java z bazami danych wykorzystując nowoczesne biblioteki i API języka Java.

- relacyjne bazy danych
 - uruchamianie serwera baz danych
 - zasada działania
 - tworzenie baz danych
 - relacje
- język SQL
 - wstawianie rekordów do bazy
 - zapytania
 - modyfikacja danych
 - usuwanie rekordów
 - JOIN
 - operacje na zbiorach danych
 - grupowanie wyników
 - sortowanie
 - zapytania zagnieżdżone
 - klucze prywatne
 - auto inkrementacja
 - wartości domyślne
- JDBC
 - połączenie Java z bazą danych
 - operacje na bazie danych przy użyciu Java

Technologie i narzędzia:

Będziemy również potrzebować narzędzi do utrzymywania i zarządzania projektami oraz kontroli wersji oprogramowania. Kiedy nasze aplikacje staną się bardziej skomplikowane sam język Java przestanie nam wystarczać. Nauczymy się wykorzystywać dodatkowe narzędzia zewnętrzne, nie wbudowane w język Java. Poznamy nowe idee i koncepcje tworzenia komercyjnych aplikacji. Aby zobaczyć efekty naszej pracy wprowadzimy interfejs użytkownika do aplikacji Java i napiszemy testy automatyczne sprawdzające poprawność ich działania.

- Maven
 - zarządzanie zależnościami
 - budowanie aplikacji
 - wtyczki do Maven
 - etapy budowania projektu
 - zdalne repozytorium Maven
- GIT
 - tworzenie repozytorium
 - inicjalizacja projektu git
 - dodawanie zdalnego repozytorium do projektu
 - tworzenie commit'ów
 - wrzucanie zmian do zdalnego repozytorium

- ☐ pobieranie repozytorium
 - ☐ uaktualnianie lokalnych źródeł
 - ☐ gałęzie
- Spring Boot
 - ☐ uruchamianie kontenera HTTP
 - ☐ kontenery wbudowane w Spring
- Spring IoC
 - ☐ beany
 - ☐ konteksty
 - ☐ wstrzykiwanie zależności (Dependency Injection)
- Metody konfiguracji Spring
 - ☐ pliki XML
 - ☐ Klasy konfiguracyjne
 - ☐ Adnotacje
- Spring MVC
 - ☐ wzorzec MVC
 - ☐ kontrolery
 - ☐ szablony widoków
 - ☐ mapowanie zapytań
 - ☐ metody zapytań HTTP
 - ☐ przekazywanie modelu do widoków
- Thymeleaf
- Hibernate
 - ☐ konfiguracja hibernate
 - ☐ ORM
 - ☐ zapytanie w hibernate
 - ☐ HQL
 - ☐ zapytania przy pomocy kryteriów
 - ☐ sesje
 - ☐ transakcje
- XML i JSON
- Podstawy front-end'u
 - ☐ JS
 - ☐ HTML
 - ☐ CSS
 - ☐ JQuery
 - ☐ AJAX
- Testowanie aplikacji w Java
 - ☐ JUnit
 - ☐ Mockito
 - ☐ TDD

WebServices:

Kiedy aplikacje stają się ogromne chcemy zmienić podejście do ich wytwarzania. Korzystamy w metody “dziel i zwyciężaj”, czyli dzielimy nasz wielki projekt na kilka mniejszych programów. Żeby te mniejsze “moduły” naszego

systemu mogli ze sobą współpracować musimy nauczyć się sposobów na zaimplementowanie komunikacji między nimi.

- REST
 - definicja API przy pomocy JSONSchema
 - generacja API przy pomocy Maven
 - klient REST API
 - provider REST API

Metodyki wytwarzania oprogramowania i dokumentacje:

Oprócz wytwarzania oprogramowania programista musi również dokumentować wytwarzany kod oraz projekty aplikacji. Nauczymy się, jakie dokumenty funkcjonują w świecie IT i dowiemy się jak przekładać nasz kod na formę papierową w postaci diagramów UML. Poznamy również strategię pracy podczas wytwarzania oprogramowania.

- Scrum
- Dokumentacje w IT
 - LLD
 - HLD
 - CR
 - PID
 - UM
 - DoR
 - DoD
- UML
 - diagramy przypadków użycia
 - diagramy klas
 - diagramy sekwencji
 - diagramy aktywności

Metoda realizacji szkolenia

Podczas zajęć kursanci **tworzą aplikacje od podstaw**. Poznają specyficzne technologie, zapoznają się z metodami iteracyjnego rozwijania oprogramowania i realizują scenariusze, z którymi zetkną się w pracy na stanowisku programisty.

Aplikacje utworzone podczas tych modułów będą stanowić portfolio uczestnika, potwierdzające umiejętność korzystania ze zdobytej wiedzy, a mentoring trenerski przez doświadczoną osobę w programowaniu Java pozwala na odkrywanie i rozwijanie potencjału ucznia.

W ramach naszego intensywnego Bootcampu Java do zrealizowania będą wyznaczane przez prowadzących **zadania domowe** oraz **jeden większy projekt** (ok 50 roboczogodzin) wykorzystujący umiejętności zdobyte na całym szkoleniu.

Prace domowe:

- pozwolą na obycie się z kodem, środowiskiem i zdobyciem podstawowego doświadczenia w pisaniu i projektowaniu oprogramowania.
- pomogą w przyswojeniu dobrych praktyk programowania i zrozumieniu problemów wynikających z niestarannie pisanego kodu oraz nieprzemyślanego projektu.
- pozwalają na utrwalenie wiedzy zdobytej na zajęciach, lepsze zrozumienie poznanych technologii, rozwijają zdolności algorytmiczne oraz projektowe.
- część zadań będzie polegała na samodzielnym zgłębieniu konkretnego tematu i przedyskutowaniu go na zajęciach.

Projekt końcowy:

to w pełni działająca aplikacja tworzona w metodyce agile. Wymagane jest przygotowanie dokumentacji technicznej aplikacji oraz wykorzystanie jak największej ilości technologii.

Projekt musi spełniać trzy główne założenia – **gromadzenie danych, przetwarzanie danych oraz możliwość manipulacji danymi**. Dzięki takiemu podejściu należy wykazać się umiejętnościami pisania przemyślanego kodu back-end’owego, front-end’owego oraz projektowania i implementowania prostych algorytmów.

Skończony projekt jest również dobrym **elementem składowym do portfolio przyszłego programisty**. Temat projektu można wybrać z puli przygotowanej przez trenerów lub po konsultacji i zaakceptowaniu napisać wymyśloną samodzielnie aplikację.

Podczas wykonywania zadań oraz projektu kursanci mogą liczyć na pomoc mentorów i konsultować napotkane problemy.

Liczba dni, liczba godzin szkoleniowych

14 dni szkoleniowych

+ 4h konsultacji HR