

Kod szkolenia: **JAVA/TOOLS**

Tytuł szkolenia: **Narzędzia Programisty Java**

Tools for Java Developer

Dni: 3

Opis:

Cel szkolenia

- Nabycie wiedzy odnośnie ogólnie stosowanych narzędzi i bibliotek.
- Umiejętność praktycznego ich zastosowania.
- Zwiększenie produktywności i efektywności pracy programisty.
- Zrozumienie wpływu omówionych narzędzi i bibliotek na jakość pracy oraz kodu.

Mocne strony szkolenia

- Zapoznanie się z powszechnie stosowanymi narzędziami i bibliotekami oraz różnic pomiędzy nimi
- Zapoznanie się z różnymi aspektami wytwarzania oprogramowania oraz narzędziami je wspomagającymi
- Możliwość praktycznego wypróbowania omawianych narzędzi

Grupa docelowa

Programista

Adresaci szkolenia

Szkolenie adresowane jest do programistów Java, którzy pragną zapoznać się bądź poszerzyć swoją wiedzę na temat narzędzi i bibliotek, które są powszechnie wykorzystywane w procesie wytwarzania oprogramowania.

Wymagania dla uczestników

Uczestnik szkolenia powinien posiadać podstawową znajomość programowania w języku Java.



Parametry szkolenia

3*8 godzin (3*7 godzin netto) wykładów i warsztatów w proporcji: 60% warsztaty, dyskusje; 40% wykłady.

Wielkość grupy: maks. 8-10 osób.

Trener w sieci:

Program szkolenia:

1. Wprowadzenie (0,5h)
 - Ciągłe doskonalenie i automatyzacja jako techniki zwiększenia produktywności
 - Software Craftsmanship
 - DevOps
2. Budowanie aplikacji (2h)
 - Różnice pomiędzy Maven, a Gradle
 - Budowanie aplikacji z wykorzystaniem Maven
 - Tworzenie pliku POM
 - Informacje
 - Zależności
 - Pluginy
 - Profile
3. Zarządzanie zmianami kodu (2h)
 - Czym jest system kontroli wersji?
 - Git czy SVN?
 - Omówienie systemu kontroli wersji na przykładzie Git'a
 - Różne sposoby dostarczania zmian
4. Praca z IDE na przykładzie IntelliJ (2h)
 - Integracja z Maven oraz Git
 - Wsparcie dla refaktoryzacji
 - Pluginy
 - Debuggowanie kodu
 - Pomocne skróty
5. Statyczna analiza kodu (2h)
 - Czym jest statyczna analiza kodu?
 - Checkstyle
 - PMD
 - Softbugs
 - SonarQube oraz SonarCloud
 - ArchUnit
 - Code review z wykorzystaniem Github'a
6. Dynamiczna analiza kodu (2h)



- Czym jest dynamiczna analiza kodu?
 - JUnit 5
 - AssertJ
 - Mockito
 - Code Coverage z wykorzystaniem Jacoco
 - Testy mutacyjne z wykorzystaniem Pitest
7. Testy akceptacyjne i wydajnościowe (2h)
- Selenium
 - JBehave
 - JMeter
8. Continuous Integration/Delivery/Deployment (2h)
- Czym jest Continuous Integration/Delivery/Deployment?
 - Omówienie na przykładzie Jenkins
 - Definicja zadań
 - Integracja
 - Pluginy
9. JVM (1h)
- Co warto wiedzieć o JVM?
 - VisualVM
 - Java Mission Control
10. Logowanie i monitorowanie (2h)
- Log4J
 - ELK Stack: Elasticsearch, Logstash, Kibana
 - Monitorowanie aplikacji na przykładzie Splunk'a
11. Zarządzanie pracą na przykładzie Jira (1h)
12. Dodatkowe narzędzia i biblioteki, które warto znać (2h)
- Spring Boot
 - Groovy
 - Docker
 - Cloud Computing
13. Podsumowanie (0,5h)

