

Kod szkolenia: **J/TEST**

Tytuł szkolenia: **Testowanie na platformie Java**

Dni: 2

Opis:

## Cel szkolenia

Uczestnicy poznają narzędzia i techniki umożliwiające zwiększenie jakości wytwarzanego oprogramowania poprzez efektywne wykorzystanie testów automatycznych na różnych poziomach. Dowiedzą się jak wykorzystywać poznane narzędzia i techniki, aby projekt był łatwiejszy w utrzymaniu.

## Mocne strony szkolenia

- Szkolenie prowadzone jest w formie warsztatów.
- Szkolenie wychodzi poza testy jednostkowe i pokazuje jak wykorzystać testy na innych poziomach.
- Szkolenie nie tylko pokazuje jak pisać testy, ale także jak pisać kod który łatwo jest testować i utrzymywać.
- Przedstawiane są techniki umożliwiające wdrażanie testów automatycznych w odziedziczonych projektach.
- Pokazywane jest jak zmierzyć poziom pokrycia testami i prawidłowo go interpretować.

## Grupa docelowa

Programista

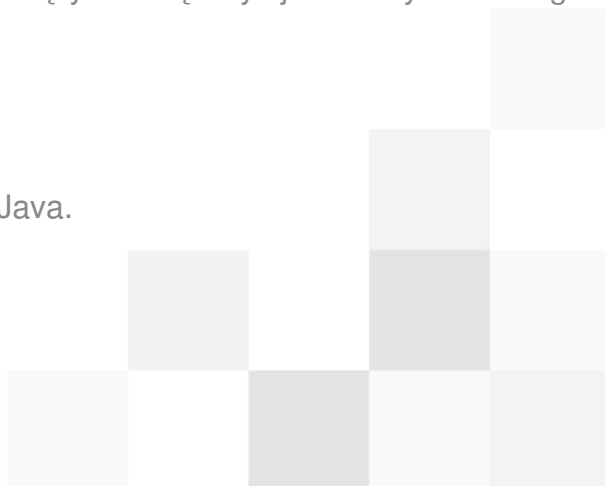
## Adresaci szkolenia

Szkolenie adresowane jest do programistów Java, chcących zwiększyć jakość wytwarzanego oprogramowania.

## Wymagania dla uczestników

Podstawowa umiejętność programowania w języku Java.

## Parametry szkolenia



2\*8 godzin (2\*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów).

Wielkość grupy: maks. 8-10 osób.

## Program szkolenia:

### 1. Wprowadzenie (1 h)

- Retrospekcja bieżącej sytuacji w projekcie
- Poziomy testowania
- Metody testowania
- Miejsce testowanie w różnych podejściach do rozwoju oprogramowania (Agile, Devops, Waterfall)

### 2. Testy jednostkowe (1.5 h)

- Właściwości
- Narzędzia
- Dobre praktyki

### 3. Rozwój oprogramowania sterowany testami - TDD (1.5 h)

- Cykl Red-Green-Refactor
- Zalety i zagrożenia
- TDD vs BDD
- Praktyczne podejście
- Kata

### 4. Zależności od innych obiektów (1.5 h)

- Obiekty współpracujące
- Mockito
- Dobre praktyki

### 5. Testy integracyjne (2 h)

- wsparcie popularnych frameworków
- Spring Boot



## 6. Testy systemowe (1.5 h)

- Klient REST
- Selenium

## 7. Testy niefunkcjonalne (1 h)

- JMeter

## 8. Metryki pokrycia testami (1 h)

- JaCoCo
- Testy mutacyjne

## 9. Praca z odziedziczonym kodem (2 h)

- Testowalność kodu
- Refaktoring

## 10. Integracja testów z innymi narzędziami (1 h)

- Maven/Gradle
- CI

