

## Lab 4

### Testy jednostkowe + wyrażenia regularne

#### Wstęp

Instrukcja przedstawia realizację ćwiczenia w języku java z wykorzystaniem JUnit. Dopuszcza się realizację ćwiczenia z wykorzystaniem innego języka programowania i/lub frameworku do testów jednostkowych.

#### Przebieg ćwiczenia

##### JUnit

- W środowisku Netbeans utworzyć projekt Maven.
- W nowej klasie utworzyć metodę walidującą kod pocztowy:

```
public boolean validateZipCode(String zipCode) {  
    return zipCode.matches("regex");  
}
```
- Utworzyć metodę zwracającą adresy email z długiego tekstu (np kodu strony internetowej):

```
public List<String> findEmails(String longText) {  
    Pattern p = Pattern.compile("regex");  
    Matcher m = p.matcher(longText);  
    List<String> results = new ArrayList<>();  
    while(m.find()) {  
        results.add(m.group());  
    }  
    return results;  
}
```
- Zamiast "regex" wstawić prawidłowe wyrażenia regularne.
- Utworzyć klasę testu (kliknąć prawym klawiszem myszy na klasie testowanej w menu Tools, następnie „Create/Update Tests” i zaznaczyć metody do testowania).
- Zaimplementować minimum 3 testy do każdej testowanej metody.

#### Sprawozdanie

- Sprawozdanie wspólne dla lab 4 i 5.
- W sprawozdaniu umieścić kod źródłowy klasy testowanej i testującej.

#### FAQ

P: "Jak napisać testy?" - zgodnie z treścią wykładów: utworzyć obiekt testowanej klasy, parametr wejściowy, zmienną do przechowania wyniku, zadeklarować oczekiwany wynik i/lub porównać otrzymany wynik z oczekiwanym.

P: "Czemu przeklejony kod wyrzuca błędy?" - Należy zaimportować brakujące biblioteki (Ctrl+spacja). Problem może również wystąpić w przypadku błędnego kodowania. Metodą na to jest przepisanie lub refaktoryzacja kodu.