

Metody Programowania

Laboratorium 2 : „Testy Jednostkowe”

Zadania do samodzielnego wykonania :

Zadanie 1: Prosty Kalkulator (10 pkt)

1. Utwórz nowy projekt typu Maven
2. Rozszerz plik pom.xml o dependencję dla biblioteki JUnit w wersji 4.12
3. Utwórz klasę *Calculator* w pakiecie pl.pjatk.unit_tests oraz odpowiadającą jej klasę testową w odpowiednim pakiecie
4. Dodaj metodę statyczną *power* :
 - a. przyjmującą 2 parametry typu int
 - b. zwracającą wynik również w postaci liczby prostej
 - c. zaimplementuj metodę nie korzystając z narzędzi typu Math.pow()
 - d. napisz 2 testy automatyczne weryfikujące poprawność utworzonej metody
5. Dodaj metodę statyczną sprawdzającą czy pierwszy przesłany parametr jest podzielny przez drugi
 - a. 2 argumenty typu int na wejściu
 - b. metoda zwraca true/false
 - c. napisz 2 testy automatyczne weryfikujące poprawność utworzonej metody
6. Dodaj metodę statyczną obliczającą największy wspólny dzielnik dla przekazanych liczb
 - a. 2 argumenty typu int na wejściu
 - b. metoda zwraca największy wspólny dzielnik
 - c. napisz 2 testy automatyczne weryfikujące poprawność utworzonej metody

Zadanie 2: Proste narzędzia tekstowe (10 pkt)

1. Utwórz klasę *StringUtils* w pakiecie pl.pjatk.unit_tests oraz odpowiadającą jej klasę testową w odpowiednim pakiecie
2. Zaimplementuj, a następnie utwórz po 2 testy automatyczne dla każdej z niżej wymienionych metod:
 - a. toUpperCase(String): String [„ala ma kota” → „ALA MA KOTA”]
 - b. reverseString(String):String [„lubie programować” → „ćawomargorp eibul”]
 - c. multiplyString(String, int):String [“test “,3 → “test test test “]
 - d. removeSmallChars(String):String [“Jan Kowalski” → “J K”]
 - e. removeBigChars(String): String [“Jan Kowalski” → “an owalski”]
 - f. removeAllButLetters(String):String [“123Test!” → “Test”]

Zadanie dodatkowe* (4 pkt*)

Dodaj metodę niestatyczną do liczenia iloczynu dwóch liczb prostych. Implementacja nie może zawierać znaku ‘*’ oraz wykorzystywać zewnętrznych narzędzi. Następnie przygotuj 4 testy jednostkowe weryfikujące poprawność metody. Ogranicz tworzenie obiektu Calculator do jednego na całą klasę.

Rozwiązanie zadań powinno mieć następującą formę:

- dokument .pdf
- nazwa [IMIE]_[NAZWISKO]_MPR_LAB2
- W środku powinien znaleźć się kod klas Calculator oraz StringUtils oraz odpowiadające im klasy z testami (może być przeklejone lub screen – ważne aby było czytelne)
- Dodatkowo załączyć po screenie z wynikami testów (może być jeden screen z wykonania wszystkich testów dla projektu)