# Lekcja – testy jednostkowe

Testy jednostkowe to rodzaj testowania oprogramowania, którego celem jest weryfikacja poszczególnych jednostek kodu źródłowego, aby upewnić się, że działają one zgodnie z oczekiwaniami.

„Jednostka kodu” może być zdefiniowana na wiele sposobów, ale najczęściej odnosi się do najmniejszego fragmentu kodu, który można testować niezależnie, na przykład funkcji, metody lub klasy.

Testy jednostkowe są pisane i wykonywane przez programistów, którzy chcą upewnić się, że ich kod działa prawidłowo przed jego integracją z większym systemem. Kluczową zaletą testów jednostkowych jest to, że mogą one szybko identyfikować błędy na wczesnym etapie rozwoju oprogramowania, co może znacznie zmniejszyć koszty naprawy błędów w późniejszym czasie.

JUnit jest biblioteką do tworzenia testów jednostkowych dla aplikacji Java.

JUnit zapewnia prostą i skuteczną metodę tworzenia testów, dzięki użyciu adnotacji do oznaczania metod testowych, oraz asercji do sprawdzania zachowania kodu. Umożliwia programistom łatwe określenie, jakie wyniki są oczekiwane od testowanego kodu, i automatycznie weryfikuje te oczekiwania.

Przykład testu:

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertEquals;

class CalculatorTest {

@Test

void testAdd() {

// Utworzenie instancji klasy Calculator

Calculator calculator = new Calculator();

// Wywołanie metody add i przypisanie wyniku do zmiennej result

int result = calculator.add(10, 5);

// Sprawdzenie, czy wynik jest zgodny z oczekiwaniami

assertEquals(15, result);

}

}

class Calculator {

public int add(int number1, int number2) {

return number1 + number2;

}

}

## Zadanie 1

Przygotuj nowe repozytorium z kodem i umieść w nim zadania z ćwiczeń 2. Repozytorium powinno zostać umieszczona na GitHub’ie.

Pamiętaj, aby repozytorium było publiczne.

## Zadanie 2

Dodaj testy jednostkowe do przygotowanych wcześniej ćwiczeń z algorytmów i struktur danych.