Харківський національний університет ім. В. Н. Каразіна Факультет комп'ютерних наук

Лабораторна робота №3

3 дисципліни

«Математичні методи і технології тестування та верифікації програмного забезпечення»

Тема: «Вивчення модульного тестування (unit testing) на прикладі фреймворка JUnit»

Виконав:

Студенте групи КС-21

Зоренко Я. С.

Перевірив:

Доц. Нарєжній О. П.

На початку роботи ми встановлюємо інтегроване середовище розробки програмного забезпечення IntelliJ IDEA (Рис. 1).



Рисунок 1 – Установка інтегрованого середовища IntelliJ IDEA

Одразу після цього, завантажуємо JDK для IntelliJ IDEA (Рис. 2).

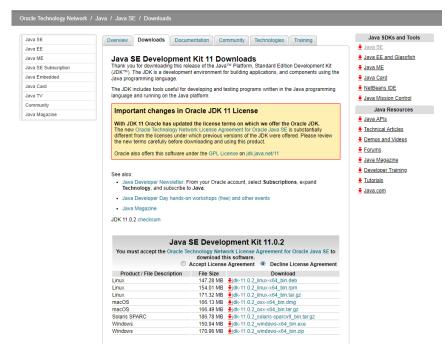


Рисунок 2 – Завантаження JKD

Опісля завантажуємо та додаємо бібліотеку модульного тестування JUnit до інтегрованого середовища IntelliJ IDEA (Рис. 3).

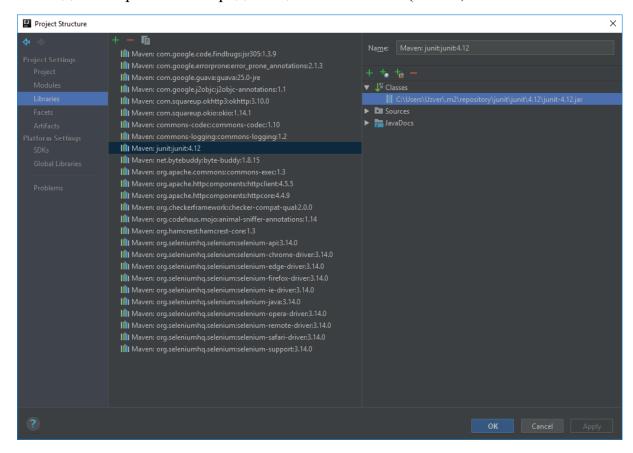


Рисунок 3 – Додання бібліотеки JUnit

Створюємо Maven проект та даємо назву "ProjectMaven" (Рис. 4-5)

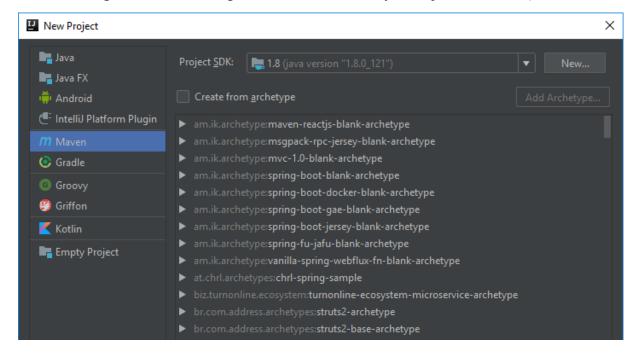


Рисунок 4 – Створення Maven проекту

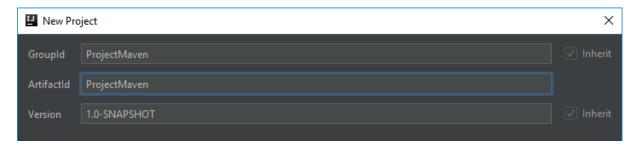


Рисунок 5 - Створення "ProjectMaven" проекту

Пишемо невеличку програму «Калькулятор» для проведення тестів (Рис. 6).

```
<u>F</u>ile <u>E</u>dit <u>V</u>iew <u>N</u>avigate <u>C</u>ode Analy<u>z</u>e <u>R</u>efactor <u>B</u>uild R<u>u</u>n <u>T</u>ools VC<u>S <u>W</u>indow <u>H</u>elp</u>
ProjectMaven ⟩ ■ src ⟩ ■ main ⟩ ■ java ⟩ 🚮 Calculator.java
                                               😌 😤 💠 — 🚮 Calculator.java × 💰 CalculateTest.java × 🎯 Calculator.class × 🚳 CalculateTest.class
                                                                                import java.util.Scanner;
   ▼ II- ProjectMaven I:\2й курс -4сем-\QA\3\ProjectMaven
     ▶ 🖿 .idea
      ▼ 🖿 src
                                                                                       Scanner scanner = new Scanner(System.in);
Calculator calculator = new Calculator();
                  of Calculator.java
             ▼ 🖿 java
                                                                                        System.out.println("Choose operation +, -, *,/:");
String operation = new String ();
                  ProjectMaven.iml
    ► Ill External Libraries
       Scratches and Consoles
                                                                                          return (a+b);
                                                                                          System.out.println("Result "+ (a*b));
                                                                                          return (a*b);
```

Рисунок 6 – Програма «Калькулятор»

Після закінчення підготовки, ми нарешті тестуємо нашу готову програму «Калькулятор» (Рис. 7).

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

ProjectMaven Strc Stratches and Consoles

Calculaterest, java

Calculaterest {

Calculator calculaterest {

Calculator calculator = new Calculator();

Calculator. Division (123,0);

double expected result = Infinity;

Assert. assert Equals (expected result, result, 0);

ProjectMaven. inl

ProjectMaven. inl

Calculaterest, java

Calculaterest (

Calculator calculator = new Calculator. Division (123,0);

double expected result = Infinity;

Assert. assert Equals (expected result, result, 0);

Calculator calculator. Division (123,0);

double expected result = Infinity;

Assert. assert Equals (expected result, result, 0);

Calculator calculator. Division (123,0);

double expected result, result, 0);

Calculator calculator. Division (123,0);

double expected result, result, 0);

Calculator calculator. Division (123,0);

double expected result, result, 0);

Calculator calculator. Division (123,0);

double expected result, result, 0);
```

Рисунок 7 – Тестування програми «Калькулятор»

Висновок: в цій лабораторній роботі ми навчились встановлювати інтегроване середовище розробки програмного забезпечення IntelliJ IDEA та підключати до нього JDK. Також ми використовували додаткову бібліотеку модульного тестування JUnit, за допомогою якої ми тестували заздалегідь написану програму «Калькулятор».