

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп'ютерних систем та інфокомунікацій

Кафедра комп'ютерних систем, мереж і кібербезпеки

## Лабораторна робота

з Кросплатформенні технології

(назва дисципліни)

на тему: «Інструментальні засоби розроблення для платформонезалежного програмування»

Виконав: студент 4 курсу групи № 5456  
напряму підготовки (спеціальності)  
123 – комп'ютерна інженерія

\_\_\_\_\_  
(шифр і назва напряму підготовки (спеціальності))

Тимошенко О.О.

\_\_\_\_\_  
(прізвище й ініціали студента)

Прийняв: асистент каф.503

Годованюк П.А.

\_\_\_\_\_  
(посада, науковий ступінь, прізвище й ініціали)

Національна шкала: \_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

## 1. Постановка мети

1.1 Ознайомитися з інтегрованим середовищем розробки Netbeans (або іншою IDE), тому я ознайомлюсь з IntelliJ IDEA;

1.2 Ознайомитися з інтегрованим середовищем розробки Eclipse.

## 2. Створення простого додатку за допомогою IntelliJ IDEA

Спочатку я створив проект по шаблону «Console application», далі додав в метод main виведення простого тексту на термінал і збудував проект. Скріншот дерева файлів проекту та коду приведений на рис. 1.

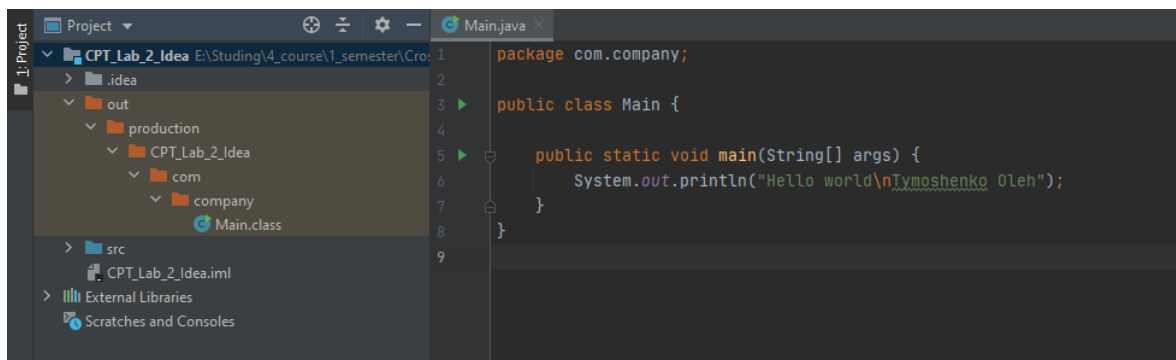


Рисунок 1 – Дерево файлів та код проекту

Скріншот виконання додатку, а саме виводу в термінал, приведений на рис.

2.

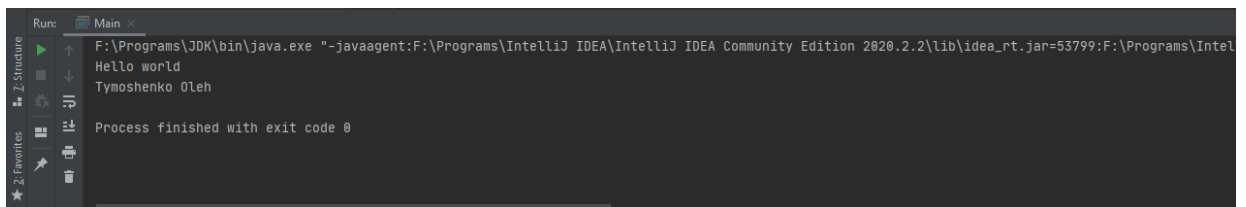


Рисунок 2 – Результат виводу в термінал

## 3. Основи розроблення Java застосунку в середовищі Eclipse

Спочатку я створив 2 проекти, додав бібліотеку до проекту основної програми, та написав в головній програмі простий код. Код приведений на рис. 3

```

LibClass.java Main.java
1 package acrostic;
2
3 import org.me.mylib.LibClass;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         String result = LibClass.acrostic(args);
9         System.out.println("Result = " + result);
10    }
11
12 }
13

```

Рисунок 3 – Код метода main

Тепер додам вхідним параметром в програму текст, щоб в результаті виконання програми отримати рядок – Hello, це зображено на рисунку 4.

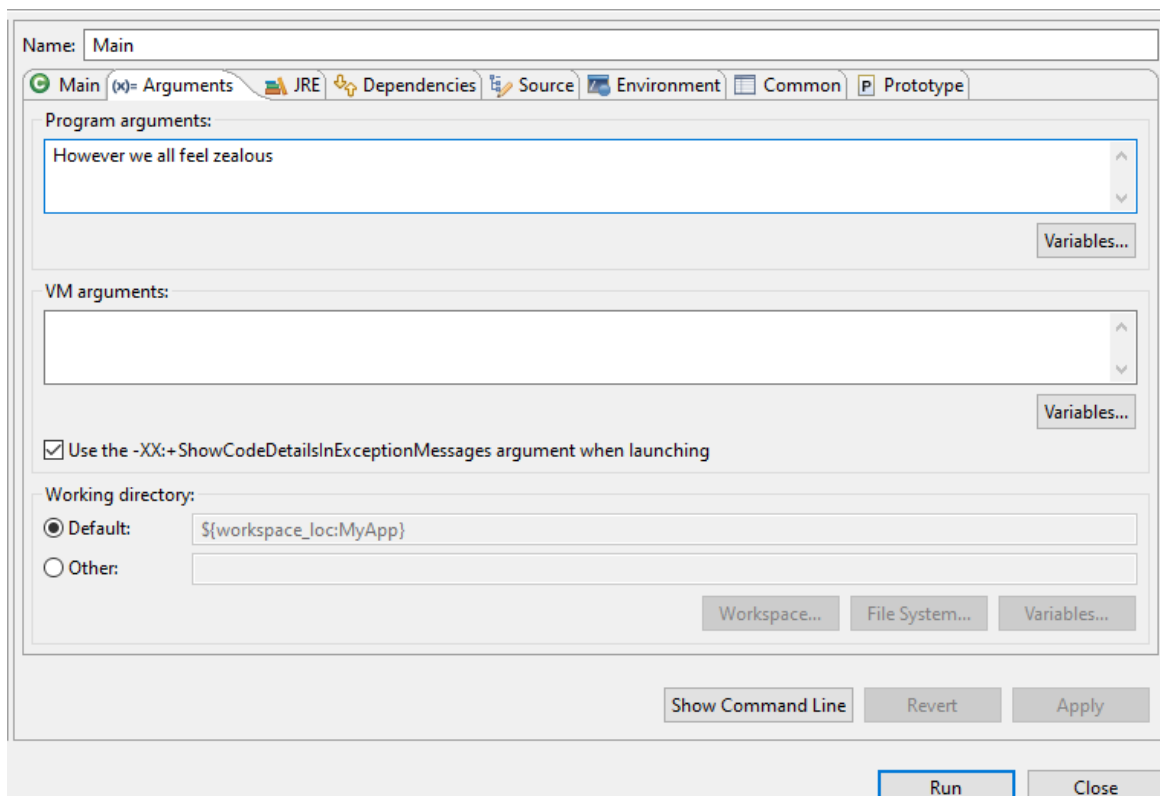


Рисунок 4 – Встановлення вхідних аргументів для додатку

Далі додам коментарі до коду та сгенерую документацію, приклад генерації приведений на рис. 5.

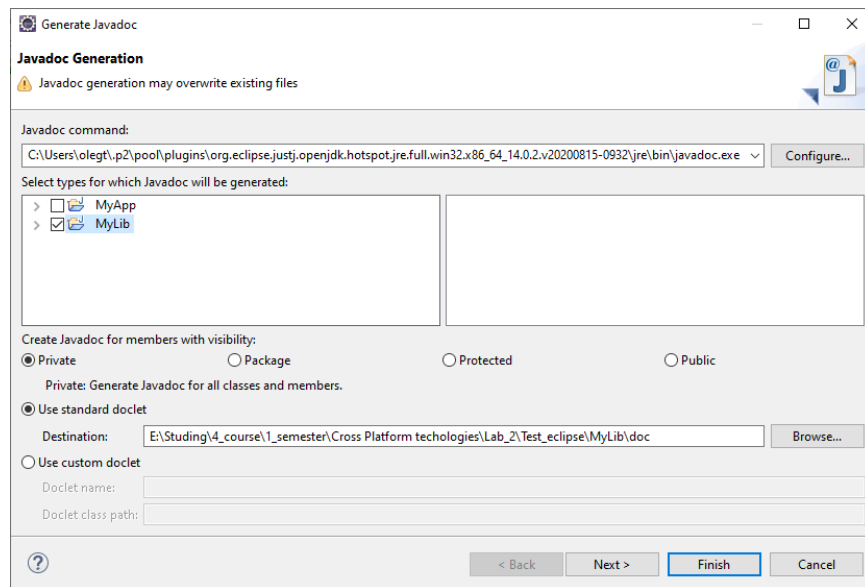


Рисунок 5 – Вікно для генерації документації

Тепер створю тест для методу бібліотеки, та виконаю його, це приведено на рис. 6.

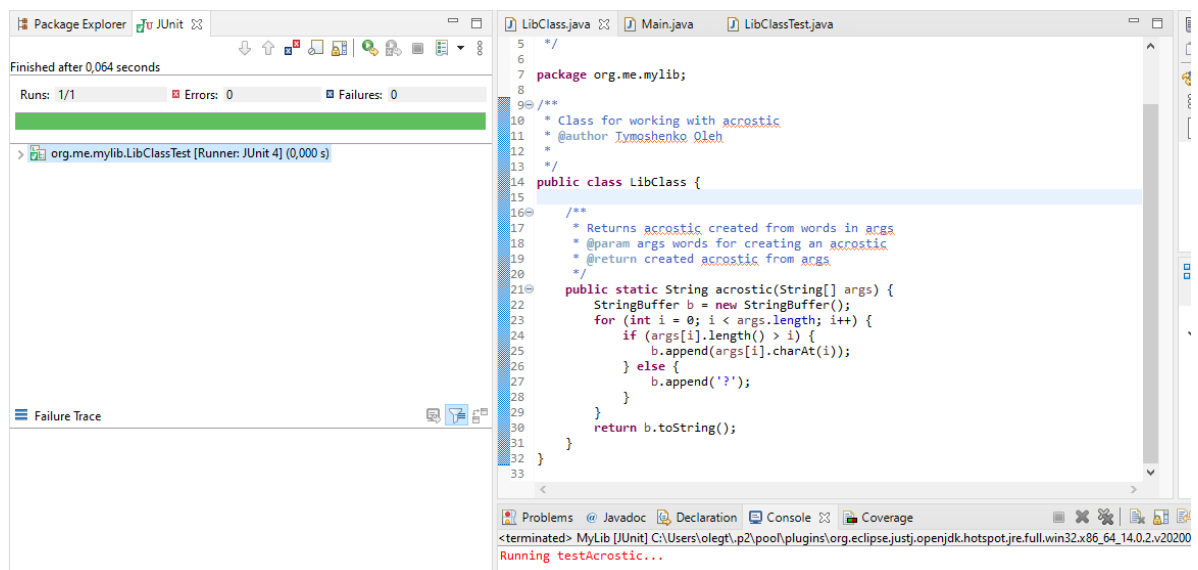


Рисунок 5 – Скріншот тесту та результату його виконання

Тепер спробую додати breakpoint в бібліотеку, і на цій точці зупинки подивитися значення всіх змінних, результат цих дій приведений на рис. 6.

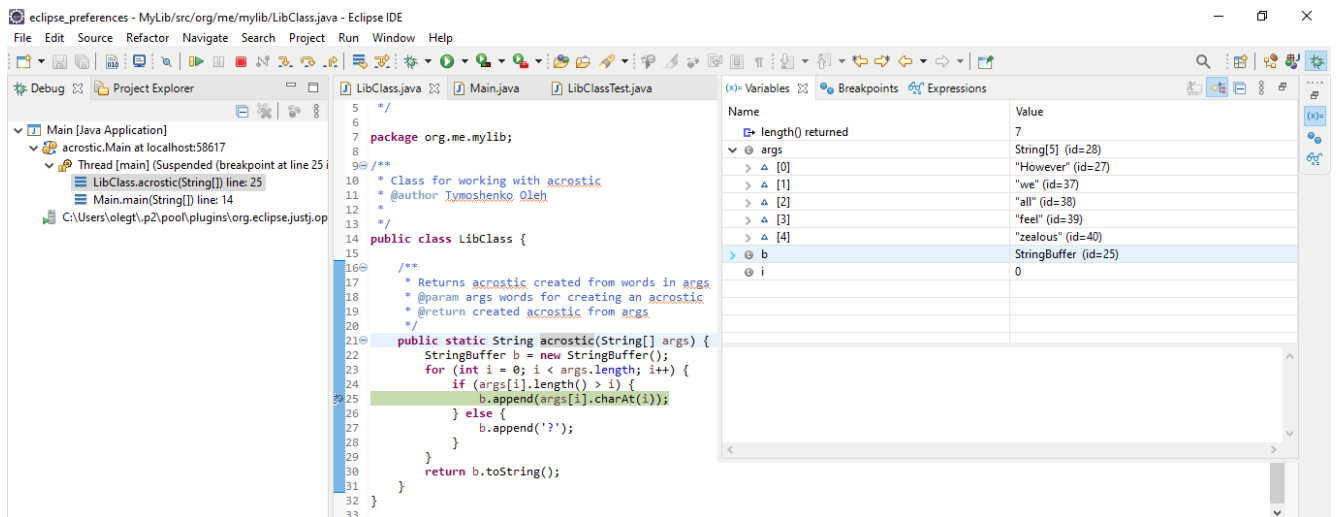


Рисунок 6 – Відладка додатку з відображення значення всіх змінних

Структура проекту після всіх проведених дій представлена на рис. 7.

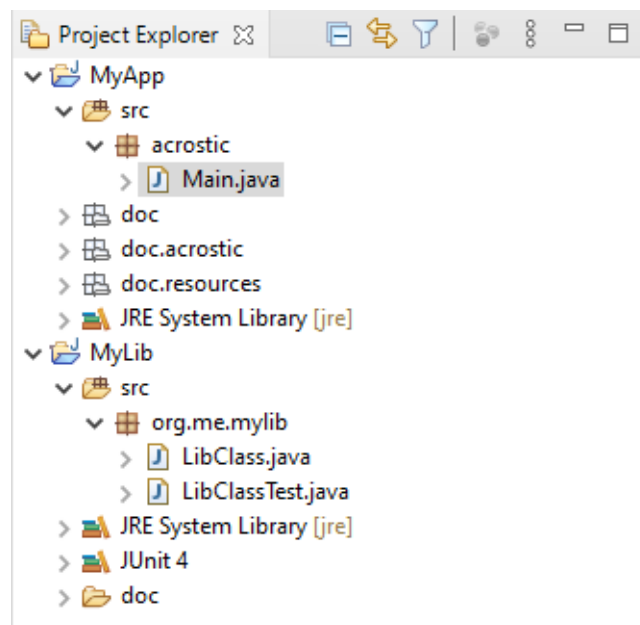


Рисунок 7 – Структура проекту після всіх проведених дій

#### 4. Основи розроблення Java застосунків в Eclipse IDE

Package Explorer з доданим архівом Junit та розгорнутим файлом `TestCase.java` представлений на рис. 8.

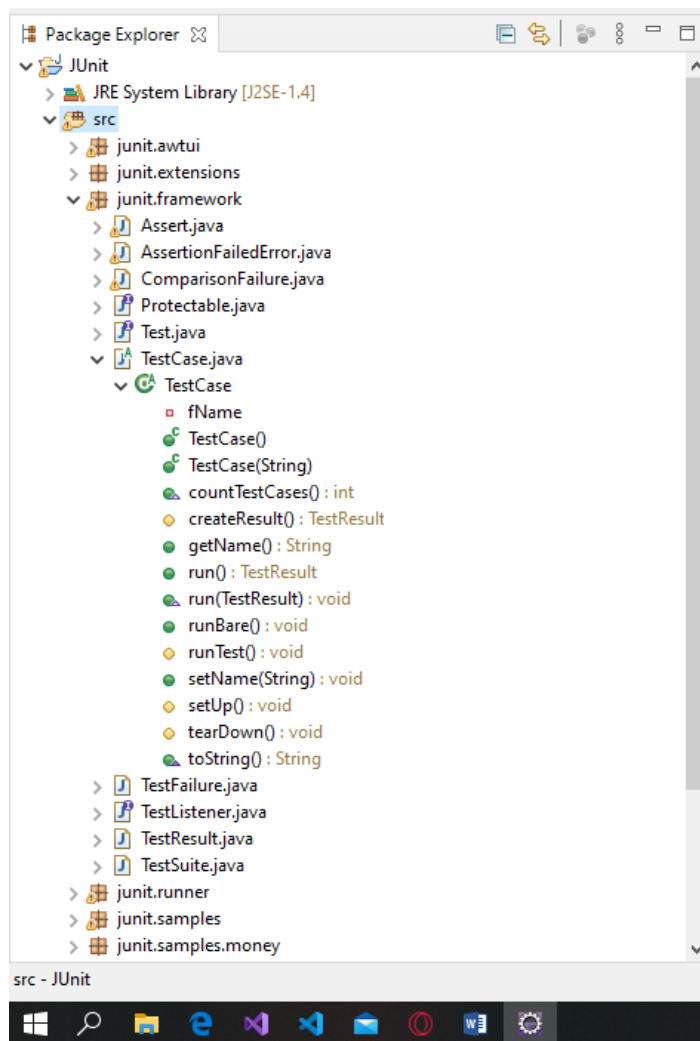


Рисунок 8 – Package Explorer з доданим архівом Junit та розгорнутим файлом  
TestCase.java

Згідно завданню, далі я додав новий метод в VectorTest.java, скріншот методу в коді і нової структури проекту приведений на рис. 9

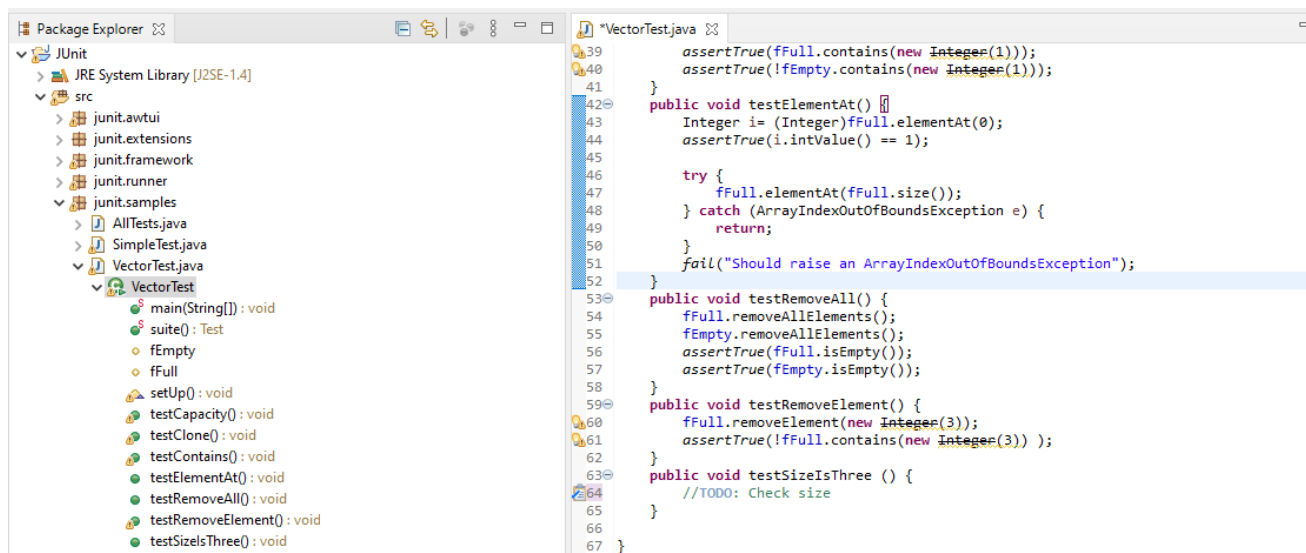


Рисунок 9 – Скріншот методу в кодї і нової структури проекту

Тепер внесу в код помилку, видаливши символ ';' в файлі TestCase.java та збережу файл. В результаті IDE підсвітить помилки в декількох місцях, скріншот цих помилок приведений на рис. 10.

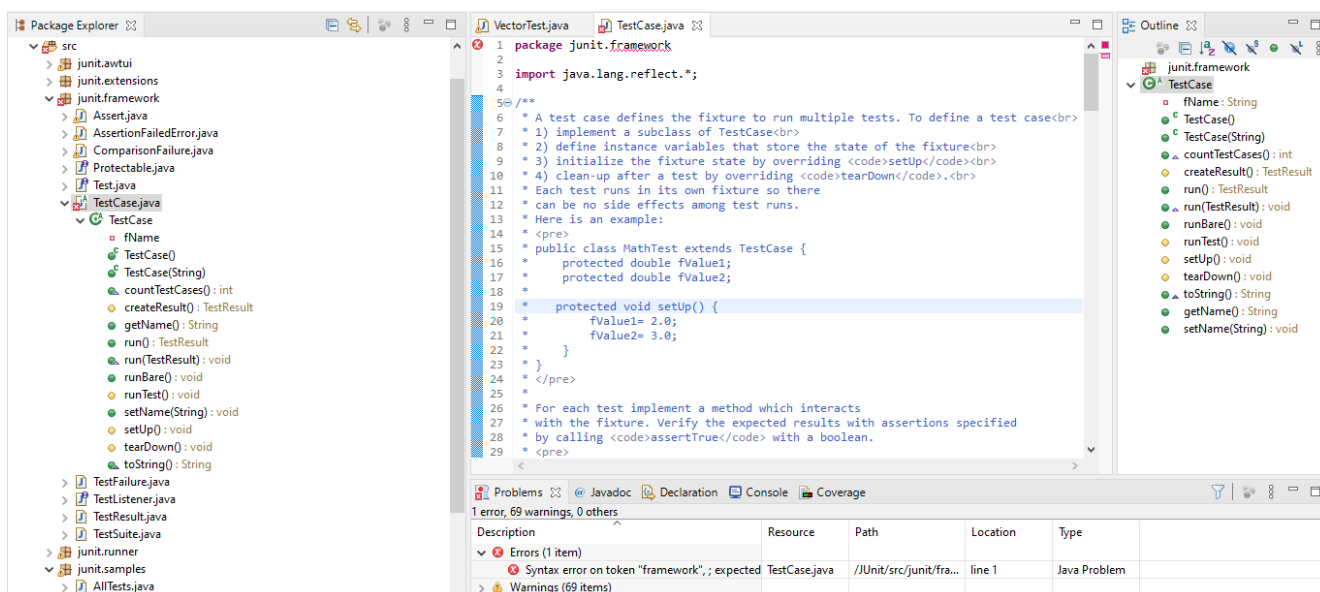


Рисунок 10 – Скріншот помилок в декількох місцях

Далі я створив новий метод testValues, і написав у ньому код за допомогою шаблонів, результат цього приведений на рисунку 11.



Рисунок 11 – Скріншот створеного за допомогою шаблонів методу та структури проекту

Далі я створив новий пакет, клас, додав туди методи, властивість та геттер і сеттер. Скріншот коду і структури проекту в результаті приведенний на рис. 12.

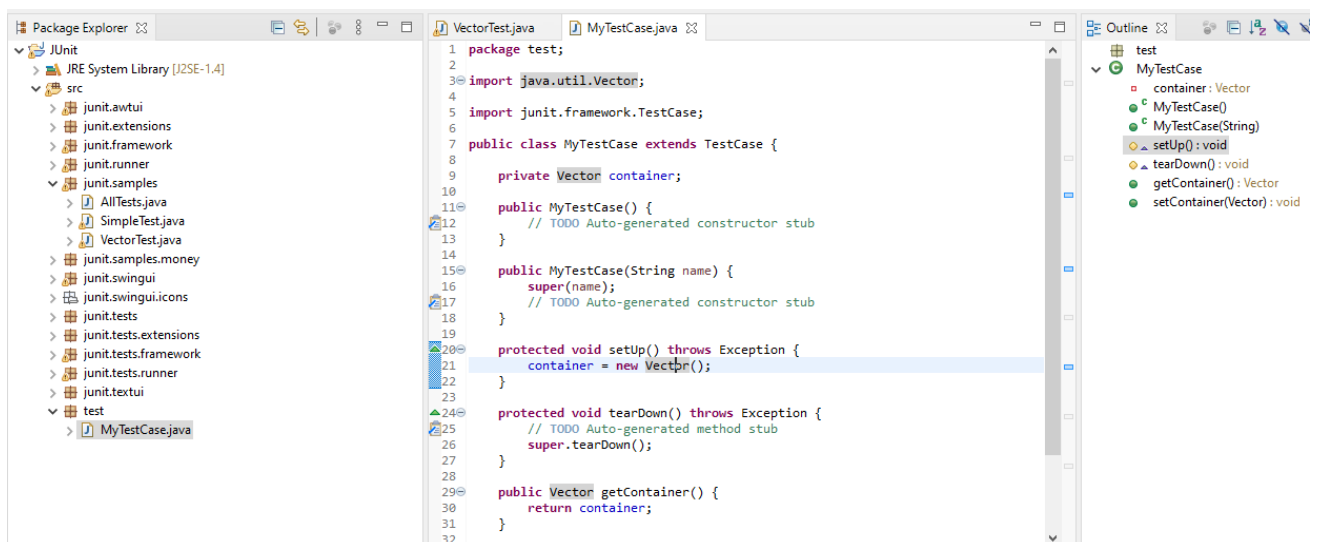


Рисунок 12 – Скріншот коду і структури проекту

5. Посилання на репозитарій із проектом у системі керування версіями.

<https://github.com/OlehTymoshenko/Cross-platform-technologies>



## **Висновки:**

В результаті виконання лабораторної роботи я ознайомився з 2-ма основними IDE для розробки на Java, це IntelliJ IDEA та Eclipse. Я створив простий додаток, який складався з основного додатку, бібліотеки та юніт-тесту методу бібліотеки. Також я попрацював з архівом Junit в Eclipse IDE, навчився використовувати основні можливості цієї середовища розробки для більш швидкого та зручного написання коду. Після недовгого ознайомлення з цими 2-ма середовищами розробки, для себе я вирішив, що IntelliJ IDEA більше сподобалася мені через більшу простоту та вигляд дизайну ide. В плані функціоналу, як мінімум для початково рівня, обидві середовища розробки дають хороші можливості для зручного написання коду.