Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Кубанский государственный технологический университет»

(ФГБОУ ВО «КубГТУ»)

Факультет информационных технологий и кибербезопасности

Кафедра информационных систем и программирования

**ОТЧЕТ**

по лабораторной работе №12.

Тема работы: «Тестирование программ при отсутствии исходного кода методом черного ящика».

Наименование дисциплины: «Тестирование и отладка программного обеспечения».

Подготовил:

Студент группы 23-КБ-ПР1

Юданов М.Д.

Краснодар

2025

**1 Цель работы**

Цель работы – закрепить принципы тестирования методом черного ящика и использования внешних библиотек.

2 **Краткое теоретическое описание**

Метод черного ящика используется для тестирования программ, когда исходный код недоступен, позволяя проверять функциональность только по входным и выходным данным. Для работы с внешними библиотеками их необходимо подключить к проекту либо с помощью сборщика (Maven/Gradle), указав нужную зависимость, либо локально через возможности IntelliJ IDEA. С помощью модульных тестов фиксируется поведение классов библиотеки, что позволяет выявлять ошибки и обеспечивать корректную работу при повторном использовании функциональности.

**3 Задание**

1) Подключить внешнюю библиотеку к новому проекту.

2) В код части классов библиотеки целенаправленно были внесены одна или несколько ошибок. Часть классов работает корректно.

3) Создать проект с модульными тестами для подключенной библиотеки и написать тесты для класса Area реализующей проверку попадания в закрашенную область фигуры в соответствии с вариантом задания из пункта 5. Необходимо обнаружить ошибки в коде (если они есть) и указать при каких условиях они возникают.

4) Оформить отчёт.

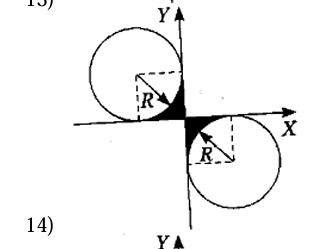


Рисунок 1 – Вариант задания

**4** **Ход работы**

1) Написал код, содержащий интерфейс и класс Area для 14 варианта, создал с помощью Maven .jar файл, после с помощью IntelliJ IDEA в новом проекте добавил его.

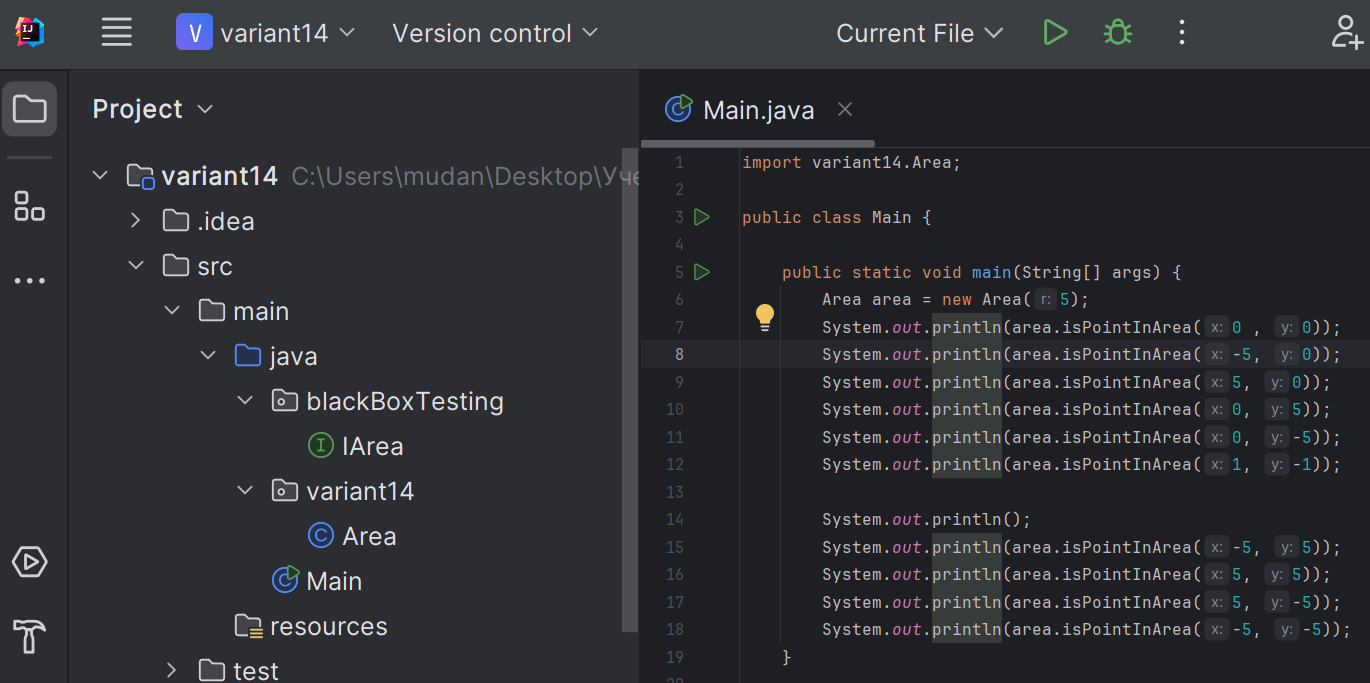


Рисунок 2 – Класс Area, интерфейс IArea

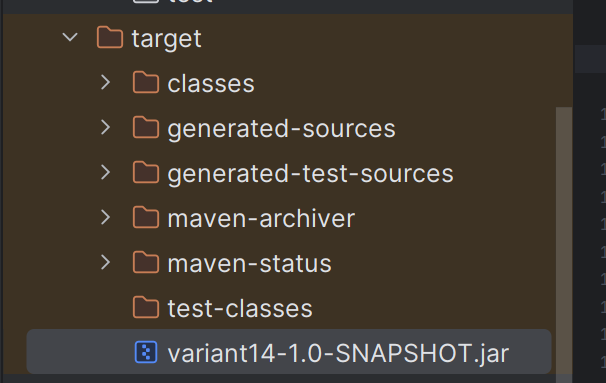


Рисунок 3 – Полученный .jar файл

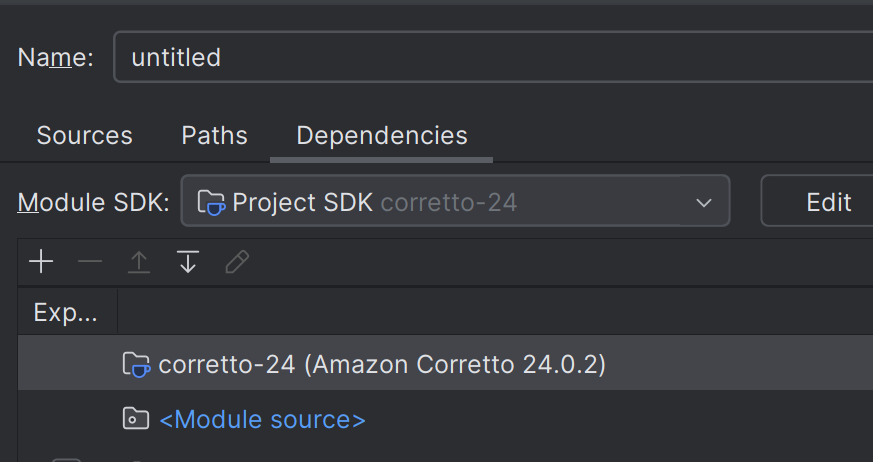


Рисунок 4 – Подключение .jar в новом проекте «lab»

2-3) Определим классы эквивалентности:

1. Правильные классы эквивалентности.

a. Тёмная область между кругами (включая точки на окружности).

1. Неправильные классы эквивалентности.

a. Введено не число (отбрасывается на моменте компиляции);

b. Всё, что не попадает в тёмную область.

Согласно классам эквивалентности напишем тесты.

import org.junit.jupiter.api.BeforeAll;

import org.junit.jupiter.api.Test;

import variant14.Area;

import static org.junit.jupiter.api.Assertions.\*;

public class AreaTest {

private static Area area;

@BeforeAll

static void setUp() {

area = new Area(5);

}

@Test

void inArea() {

assertTrue(area.isPointInArea(0, 0));

assertTrue(area.isPointInArea(0, 5));

assertTrue(area.isPointInArea(0, -5));

assertTrue(area.isPointInArea(-5, 0));

assertTrue(area.isPointInArea(5, 0));

assertTrue(area.isPointInArea(-1,1));

assertTrue(area.isPointInArea(1, -1));

}

@Test

void notInArea() {

assertFalse(area.isPointInArea(50, 50));

assertFalse(area.isPointInArea(-5, 5));

assertFalse(area.isPointInArea(5, -5));

assertFalse(area.isPointInArea(-4, 1));

}

}

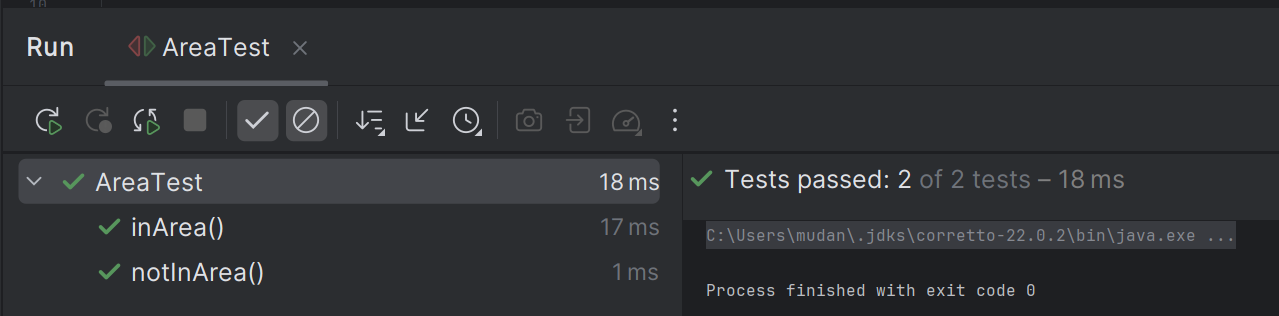


Рисунок 5 – Результат выполнения тестирования

**5 Вывод**

В ходе выполнения данной лабораторной работы удалось закрепить принципы тестирования методом черного ящика и использования внешних библиотек.