Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Кубанский государственный технологический университет»

Кафедра информационных систем и программирования

**ТЕСТИРОВАНИЕ И ОТЛАДКА  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Отчет по лабораторной работе №2

«Тестирование методом черного ящика»

Выполнил:

студент 3 курса

группы 19-КБ-ПР1

Кравцов Олег Юрьевич.

Проверил:

ст. преп. каф. ИСП

А. Г. Волик

Краснодар

2021

# 1 Цель работы

Изучить подход к тестированию методом черного ящика.

# 2 Задание

1) Создать класс (в соответствии с вариантом задания из п.5), реализующий проверку принадлежности точки различным областям плоскости, задаваемых пересечением фигур.

2) Выделить классы эквивалентности.

3) Проверить граничные значения для каждого из классов эквивалентности.

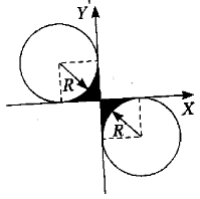
4) Составить наборы тестовых данных для созданного метода.

5) Протестировать метод на основе тестового набора с использованием программных отладочных средств.

6) Составить отчет о результатах проведенного тестирования.

# 3 Вариант задания

Вариант №12.



# 4 Текст программы

Класс проверки точки на принадлежность различным областям.

using System;

using System.Collections.Generic;

using System.Text;

namespace ConsoleApp5

{

class Program

{

public static void Main(string[]args)

{

}

}

public static class Point

{

public static int CheckPoint(int X, int Y, int R)

{

if (X <= 0 && Y >= 0 && X >= (-1 \* R) && Y <= R && (Math.Pow(X + R, 2) + Math.Pow(Y - R, 2) >= Math.Pow(R, 2))) return 1;

if (X >= 0 && Y <= 0 && X <= R && Y >= (-1 \* R) && (Math.Pow(X - R, 2) + Math.Pow(Y + R, 2) >= Math.Pow(R, 2))) return 2;

return 3;

}

}

}

# 5 Текст модульных тестов

using NUnit.Framework;

namespace Lb\_2\_tests

{

public class Tests

{

[Test]

public void CheckLeftTopAreaTest()

{

int X = -1;

int Y = 1;

int R = 10;

int actually = ConsoleApp5.Point.CheckPoint(X, Y, R);

int expect = 1;

Assert.AreEqual(expect, actually);

}

[Test]

public void CheckRightBotAreaTest()

{

int X = 1;

int Y = -1;

int R = 10;

int actually = ConsoleApp5.Point.CheckPoint(X, Y, R);

int expect = 2;

Assert.AreEqual(expect, actually);

}

[Test]

public void CheckOuterAreaTest()

{

int X = -3;

int Y = 2;

int R = 4;

int actually = ConsoleApp5.Point.CheckPoint(X, Y, R);

int expect = 3;

Assert.AreEqual(expect, actually);

}

[Test]

public void CheckCriticalPointLeftTopAreaTest()

{

int X = -5;

int Y = 0;

int R = 5;

int actually = ConsoleApp5.Point.CheckPoint(X, Y, R);

int expect = 1;

Assert.AreEqual(expect, actually);

}

[Test]

public void CheckCriticalPointLeftTopAreaTest2()

{

int X = 0;

int Y = 5;

int R = 5;

int actually = ConsoleApp5.Point.CheckPoint(X, Y, R);

int expect = 1;

Assert.AreEqual(expect, actually);

}

[Test]

public void CheckCriticalPointRightBotAreaTest()

{

int X = 5;

int Y = 0;

int R = 5;

int actually = ConsoleApp5.Point.CheckPoint(X, Y, R);

int expect = 2;

Assert.AreEqual(expect, actually);

}

[Test]

public void CheckCriticalPointRightBotAreaTest2()

{

int X = 0;

int Y = -5;

int R = 5;

int actually = ConsoleApp5.Point.CheckPoint(X, Y, R);

int expect = 2;

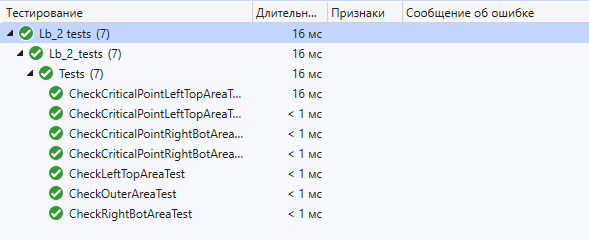
Assert.AreEqual(expect, actually);

}

}

}

**6 Результаты тестирования программы**



**7 Выводы**

В результате проделанной работы был изучен метод черного ящика, позволяющий опираясь на спецификации программы получать достаточный для тестирования тестовый набор. Был разработан класс Point, который проверяет принадлежность точки заданной области, а также был составлен набор тестов к разработанному классу.