

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
федеральное государственное автономное образовательное учреждение высшего
образования

«Санкт–Петербургский государственный университет
аэрокосмического приборостроения»

ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ _____

преподаватель

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ ПО ЛАБОРАТОРНЫМ РАБОТАМ
по дисциплине МДК 01.02

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

С326

Э.С. Тигранян

подпись, дата

инициалы, фамилия

Санкт-Петербург 2025

Лабораторная работа 2.

Часть 1.

Тестирование программ методом “белого ящика”

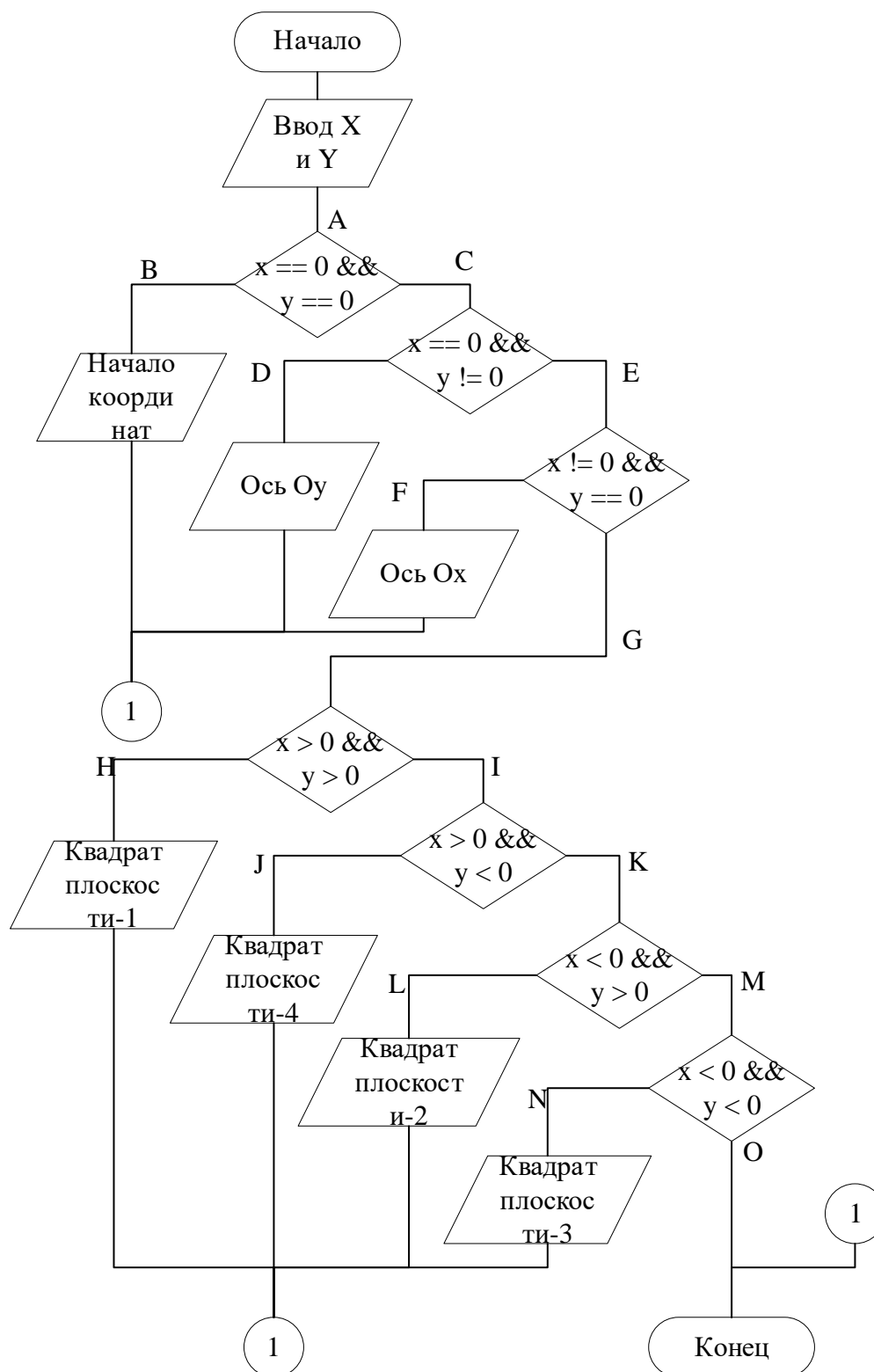
Цель работы: Усвоение студентами методов тестирования логики программы, формализованного описания результатов тестирования и стандартов по составлению схем программ.

Вариант 2 - Определить, в каком квадранте плоскости или на оси находится заданная координатами x и y точка.

Код программы:

```
namespace code
{
    class Program
    {
        static void Main(string[] args)
        {
            double x, y;
            Console.WriteLine("Введите x ");
            x = double.Parse(Console.ReadLine());
            Console.WriteLine("Введите y");
            y = double.Parse(Console.ReadLine());
            if (x == 0 && y == 0)
                Console.WriteLine("Начало координат");
            else if (x == 0 && y != 0)
                Console.WriteLine("Ось Oy");
            else if (x != 0 && y == 0)
                Console.WriteLine("Ось Ox");
            else if (x > 0 && y > 0)
                Console.WriteLine("Квадрант плоскости-1");
            else if (x > 0 && y < 0)
                Console.WriteLine("Квадрант плоскости-4");
            else if (x < 0 && y > 0)
                Console.WriteLine("Квадрант плоскости-2");
            else if (x < 0 && y < 0)
                Console.WriteLine("Квадрант плоскости-3");
        }
    }
}
```

Схема тестирования:



Был выбран метод покрытия решений так как этот метод гарантирует, что каждое условие в коде будет протестировано как в истинном, так и в ложном варианте.

Тест	Путь	Ожидаемый результат	Фактический результат	Результат тестирования: ошибка
$x = 0; y = 0$	AB	Точка в начале координат	Точка в начале координат	Не обнаружена
$x = 0; y = 2$	ACD	Точка на Оси Oy	Точка на Оси Oy	Не обнаружена
$x = -5; y = 0$	ACEF	Точка на Оси Ox	Точка на Оси Ox	Не обнаружена
$x = 5; y = 42$	ACEGH	Точка лежит на квадранте плоскости-1	Точка лежит на квадранте плоскости-1	Не обнаружена
$x = 7; y = -1$	ACEGIJ	Точка лежит на квадранте плоскости-4	Точка лежит на квадранте плоскости-4	Не обнаружена
$x = -42; y = 32$	ACEGIKL	Точка лежит на квадранте плоскости-2	Точка лежит на квадранте плоскости-2	Не обнаружена
$x = -52; y = -9$	ACEGIKMN	Точка лежит на квадранте плоскости-3	Точка лежит на квадранте плоскости-3	Не обнаружена

Контрольные вопросы

1. Какие подходы к тестированию программ Вы знаете? В чем разница этих подходов?

Подходы к тестированию программ:

- Функциональное тестирование (стратегия "черного ящика"):

Тесты проектируются на основе внешних спецификаций программы.

Логика программы не учитывается.

Цель: проверить, соответствует ли программа внешним требованиям.

- Структурное тестирование (стратегия "белого ящика"):

Тесты проектируются на основе анализа внутренней логики программы.

Внешние спецификации не учитываются.

Цель: проверить каждый путь, каждую ветвь алгоритма.

Разница подходов:

Функциональное тестирование сосредоточено на том, что делает программа (входные и выходные данные).

Структурное тестирование сосредоточено на том, как программа работает (внутренняя логика и код).

2. Что является основой для структурного тестирования?

Основой для структурного тестирования является исходный код программы. Тестирование проводится на уровне кода, где проверяются:

- Логика программы (условия, циклы, ветвления).
- Покрытие кода (строк, ветвей, путей).
- Потоки данных и управляющие структуры.

3. Какие методы структурного тестирования Вы знаете? Каковы недостатки этих методов и пути их устранения?

Методы структурного тестирования:

- Метод покрытия операторов (C0):
 - Цель: выполнить каждый оператор программы хотя бы один раз.
 - Недостаток: не проверяет все возможные пути выполнения.
 - Устранение: дополнение другими методами, например, тестированием ветвей.
- Метод покрытия решений (C1):
 - Цель: проверить каждое направление перехода (ветвь) хотя бы один раз.
 - Недостаток: не учитывает все комбинации условий.
 - Устранение: использование тестирования путей или условий.
- Метод покрытия условий:
 - Цель: проверить все возможные результаты каждого условия.
 - Недостаток: может не учитывать комбинации условий.
 - Устранение: применение комбинаторного покрытия условий.
- Метод комбинаторного покрытия условий:
 - Цель: проверить все возможные комбинации результатов условий.
 - Недостаток: требует большого количества тестов.
 - Устранение: оптимизация тестовых сценариев.

4. Что является целью тестирования? Какие выводы можно сделать, если тестирование не обнаруживает ошибок программы?

Выявление дефектов (ошибок) в программе.

Обеспечение качества программы и её соответствия требованиям.

Если тестирование не обнаруживает ошибок:

1. Программа может быть действительно качественной и соответствовать требованиям.
2. Возможно, тестовые сценарии недостаточно полны или не покрывают все возможные случаи.
3. Ошибки могут быть скрыты в сложных или редко используемых частях программы.
4. Тестирование может быть проведено на недостаточно глубоком уровне (например, только функциональное тестирование без проверки кода).

Выводы:

- Необходимо пересмотреть тестовые сценарии и увеличить покрытие тестами.
- Следует использовать комбинацию различных подходов к тестированию (функциональное, структурное, нагрузочное и т.д.).
- Важно учитывать, что отсутствие ошибок не всегда означает их реальное отсутствие.

Часть 2.

Тестирование программ методом «белого ящика»

Цель работы: отработать навыки составления и тестирования программ как «белого ящика», освоение на практике метода базового пути.

Вариант 6 - Даны натуральное число N и одномерный массив A₁, A₂, ..., A_N целых чисел. В данном массиве определить число соседств двух чисел разного знака.

Код программы:

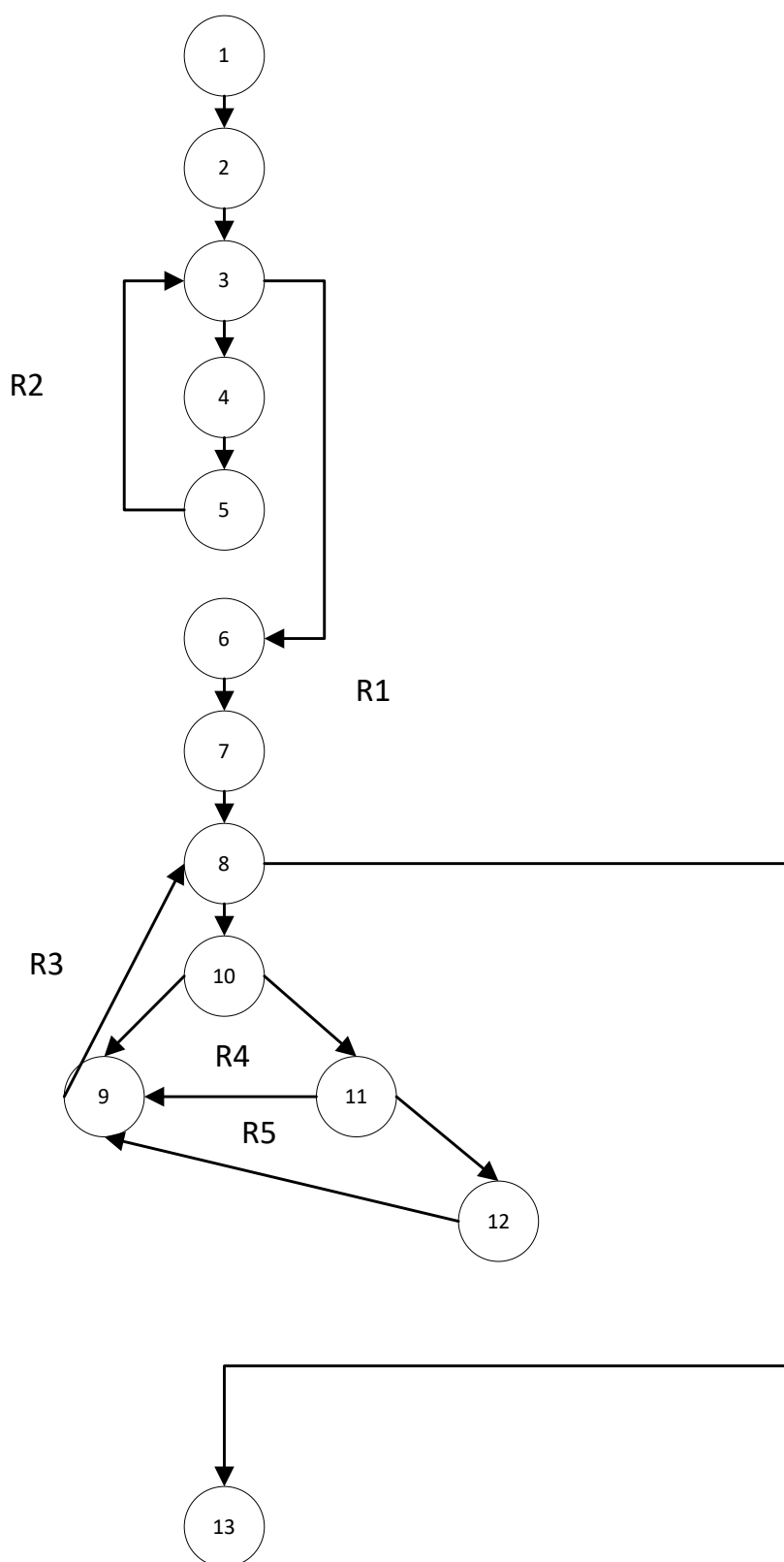
```
namespace code
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Введите N: ");
            int N = Convert.ToInt32(Console.ReadLine());
            double[] arr = new double[N];

            Console.WriteLine("Введите элементы массива:");
            for (int i = 0; i < N; i++)
            {
                arr[i] = Convert.ToDouble(Console.ReadLine());
            }

            int twoPositiveNumCounter = 0;
            for (int i = 0; i < N - 1; i++)
            {
                if (arr[i] > 0 && arr[i + 1] > 0)
                {
                    twoPositiveNumCounter++;
                }
            }

            Console.WriteLine("Число соседств двух положительных чисел: " +
twoPositiveNumCounter);
        }
    }
}
```

Потоковый граф



Путь R1: 1-2-3-6-7-8-12.

Путь R2: 1-2-3-5-4-3-6-7-8-13.

Путь R3: 1-2-3-5-4-3-6-7-8-10-9-8-13.

Путь R4: 1-2-3-5-4-3-6-7-8-10-11-9-8-13.

Путь R5: 1-2-3-5-4-3-6-7-8-10-11-12-9-8-12.

$V(G) = 5$

Базовые пути	Тест	Ожидаемый результат	Фактический результат	Результат тестирования: ошибка
R1	$N = 0$	Сообщение: «Размер массива должен быть больше 0!»	Завершение программы	обнаружена
R2, R3	$N = 1, A = \{29\}$	Число соседств двух положительных чисел 0	Число соседств двух положительных чисел 0	Не обнаружено
R3, R4,	$N = 5, A = \{42, -$ 12, 0, -43, 2}	Число соседств двух положительных чисел 0	Число соседств двух положительных чисел 0	Не обнаружено
R4, R5	$N = 5, A = \{42, 12,$ 53, 43, 2}	Число соседств двух положительных чисел 4	Число соседств двух положительных чисел 4	Не обнаружено