

Министерство науки и высшего образования РФ  
ФГАОУ ВПО  
Национальный исследовательский технологический университет «МИСиС»

---

Институт Информационных технологий и компьютерных наук (ИТКН)

Кафедра Инфокоммуникационных технологий (ИКТ)

**Отчет по лабораторной работе №2**  
по дисциплине «Программирование и Алгоритмизация»  
на тему «Организация разветвлений. Разветвления в цикле»

Выполнил:  
студент группы БИВТ-24-5

Черных Богдан

Проверил:  
Стучилин В. В.

Москва, 2024

Теоретическое введение. Разветвление – это структура, содержащая две ветви, из которых, в зависимости от условия, будет выполнена только одна.

Разветвления организуются с помощью условного оператора `if` : `if (выражение) { Операторы1 } [Else { Операторы2 }] . . .` Если выражение имеет значение `true`, то выполняются Операторы1, и происходит переход к оператору, следующему за `if`. Если выражение имеет значение `false`, то выполняются Операторы2, расположенные после `else`, и далее выполняется оператор, следующий за условным. Если вторая ветвь отсутствует (такая структура называется «обход»), то в случае невыполнения условия никаких действий не производится и сразу выполняется оператор, следующий за условным. Если блок (операторы, заключенные в `{ }`) состоит из одного оператора и не содержит описаний переменных, то скобки можно не использовать. Далее приводятся примеры программ с использованием разветвлений, разветвлений в цикле и множественного выбора.

**Цель** лабораторной работы: Изучить и приобрести навыки организации разветвлений и их интеграции в циклы с помощью условных операторов в языке программирования C#. Научиться использовать операторы условного ветвления `if`, оператор множественного выбора, а также их комбинирование с циклами для решения задач, где требуется обработка данных на основе условий.

Код требуемых заданий (1 уровень 5 задание, 1 уровень 6 задание, 1 уровень 7 задание, 2 уровень 5 задание, 2 уровень 6 задание, 2 уровень 7 задание).

(В коде используется `1 checker = 21` только для удобства – из-за того, что я использую одни и те же название переменных, например `res1`, если бы не базовое условие(`if`), мне бы пришлось каждый раз сбрасывать значение, или код бы у меня просто некорректно работал. Поэтому перед каждой новой задачей я пишу `if (checker == 21)` )

```
//svg does precious
using System;                                // Аналог <iostream> для работы с консолью
и основными функциями
using System.Collections.Generic;             // Аналог <vector>, <list>, <map>, <set>,
<unordered_map>, <unordered_set>, <stack>, <queue>
using System.Text;                           // Аналог <string>, <cstring> (для работы
со строками и StringBuilder)
using System.Linq;                           // Аналог <algorithm> (для работы с LINQ,
сортировок, поиска и т.д.)
using System.IO;                             // Аналог <cstdio>, <fstream> (работа с
файлами)
using System.Globalization;                  // Аналог <iomanip> (для форматирования)
using System.Collections;                    // Работа с различными коллекциями
(например, ArrayList)
using System.Threading;                      // Потоки и многопоточность
```

```

using System.Runtime.Serialization;    // Аналог <stdexcept> (работа с
исключениями)
using System.Reflection;              // Аналог <typeinfo> (информация о типах,
рефлексия)
using System.Diagnostics;              // Аналог <utility>, <std::pair>
(вспомогательные функции и классы)
using System.ComponentModel;           // Дополнительные утилиты и атрибуты
using System.Numerics;                // Работа с большими числами и
математическими операциями
using System.Globalization;
using System.Diagnostics;
using System.Net;
using System.Numerics;
// Для работы с потоками данных:
using System.Threading.Tasks;         // Асинхронные задачи

// Для работы с датами и временем:
using System.Timers;                 // Для работы с таймерами и временем
using System.Collections.Generic;
using System.Text;
using System.Linq;
using System.IO; //important
using C = System.Console; //console
using dl = System.Decimal; //decimal
using str = System.String; //string
using l = System.Int64; //long
using u = System.UInt64; //Ulong
using db = System.Double; //Double
//-----
-----
//-----
-----
//-----
-----

```

```

class Program
{
    //5 option(variant), Lab 2 Branching
    static void Main()
    {
        //lvl1 ex5
        l checker = 21;
        if (checker == 21)
        {
            static bool Circle(db r, db s)
            {
                db diag = Math.Sqrt(2 * s);
                db diametr = 2 * r;
                return diag <= diametr;
            }
            bool res1 = Circle(70, 36.74);
            C.WriteLine("Случай 1: r = 70, s = 36.74");
            if (res1 == true)
            {
                C.WriteLine($"Квадрат поместится в круг");
            }
            if (res1 == false)
            {
                C.WriteLine($"Квадрат не поместится в круг");
            }
            bool res2 = Circle(0.86, 0.74);
            C.WriteLine("Случай 2: r = 0.86, s = 0.74");
            if (res1 == true)
            {
                C.WriteLine($"Квадрат поместится в круг");
            }
            if (res1 == false)

```

```

        {
            C.WriteLine($"Квадрат не поместится в круг");
        }
    }
    //lvl1 ex6
    if (checker == 21)
    {
        static bool Square(db r, db s)
        {
            db diametr = 2 * r;
            db side = Math.Sqrt(s);
            return diametr <= side;
        }
        C.WriteLine("Случай 1: r = 3.2, s = 3.5");
        bool res1 = Square(3.2, 3.5);
        if (res1 == true)
        {
            C.WriteLine("Круг поместится в квадрат");
        }
        if (res1 == false)
        {
            C.WriteLine("Круг не поместится в квадрат");
        }
        C.WriteLine("Случай 2: r = 3.2, s = 4");
        bool res2 = Square(3.2, 4);
        if (res2 == true)
        {
            C.WriteLine("Круг поместится в квадрат");
        }
        if (res2 == false)
        {
            C.WriteLine("Круг не поместится в квадрат");
        }
        C.WriteLine("Случай 3: r = 6, s = 9");
        bool res3 = Square(6, 9);
        if (res3 == true)
        {
            C.WriteLine("Круг поместится в квадрат");
        }
        if (res3 == false)
        {
            C.WriteLine("Круг не поместится в квадрат");
        }
    }
    //lvl1 ex7

    if (checker == 21)
    {
        static double CalcY(double x)
        {
            if (x > 1)
                return 1;
            else
                return x;
        }
        Console.WriteLine("Введите значение x - ");
        double x = Convert.ToDouble(Console.ReadLine());
        double y = CalcY(x);
        Console.WriteLine($"При x = {x}, y = {y}");
    }

    //lvl2 ex5
    if (checker == 21)
    {
        const int n = 30;
        int svg = 0;
        Console.Write("Введите норматив (сек) - ");
    }

```

```

double normativ = Convert.ToDouble(Console.ReadLine());
for (int i = 1; i <= n; i++)
{
    Console.Write($"Введите результат спортсмена {i} (сек) - ");
    double res = Convert.ToDouble(Console.ReadLine());
    if (res >= normativ)
    {
        svg++;
    }
}
Console.WriteLine($"Количество спортсменов выполнивших норматив -
{svg}");
}
//lvl2 ex6
if (checker == 21)
{
    Console.Write("Введите количество точек - ");
    int n = Convert.ToInt32(Console.ReadLine());
    int cntsvg = 0;
    for (int i = 1; i <= n; i++)
    {
        Console.Write($"Введите x координату точки {i} - ");
        double x = Convert.ToDouble(Console.ReadLine());
        Console.Write($"Введите y координату точки {i} - ");
        double y = Convert.ToDouble(Console.ReadLine());
        if (0 <= x && x <= Math.PI && 0 <= y && y <= Math.Sin(x))
        {
            cntsvg++;
        }
    }
    Console.WriteLine($"Количество точек принадлежащих фигуре -
{cntsvg}");
}
}
//lvl2 ex7
if (checker == 21)
{
    static int detectorQuad(double x, double y)
    {
        if (x > 0 && y > 0) return 1;
        if (x < 0 && y > 0) return 2;
        if (x < 0 && y < 0) return 3;
        if (x > 0 && y < 0) return 4;
        return 0;
    }
    Console.Write("Введите количество точек - ");
    int n = Convert.ToInt32(Console.ReadLine());
    int cnt1 = 0;
    int cnt3 = 0;
    for (int i = 1; i <= n; i++)
    {
        Console.Write($"Введите x координату точки {i} - ");
        double x = Convert.ToDouble(Console.ReadLine());
        Console.Write($"Введите y координату точки {i} - ");
        double y = Convert.ToDouble(Console.ReadLine());
        int quad = detectorQuad(x, y);
        Console.WriteLine($"Точка {i} находится в {quad} квадранте");
        if (quad == 1) cnt1++;
        if (quad == 3) cnt3++;
    }
    Console.WriteLine($"Количество точек в 1-м квадранте - {cnt1}");
    Console.WriteLine($"Количество точек в 3-м квадранте - {cnt3}");
}
}
}

```

Результат выполнения программы, то есть вывод консоли:

Случай 1:  $r = 70$ ,  $s = 36.74$

Квадрат поместится в круг

Случай 2:  $r = 0.86$ ,  $s = 0.74$

Квадрат поместится в круг

Случай 1:  $r = 3.2$ ,  $s = 3.5$

Круг не поместится в квадрат

Случай 2:  $r = 3.2$ ,  $s = 4$

Круг не поместится в квадрат

Случай 3:  $r = 6$ ,  $s = 9$

Круг не поместится в квадрат

Введите значение  $x$  –

1

При  $x = 1$ ,  $y = 1$

Введите норматив (сек) – 21

Введите результат спортсмена 1 (сек) – 1

Введите результат спортсмена 2 (сек) – 2

Введите результат спортсмена 3 (сек) – 3

Введите результат спортсмена 4 (сек) – 4

Введите результат спортсмена 5 (сек) – 5

Введите результат спортсмена 6 (сек) – 6

Введите результат спортсмена 7 (сек) – 7

Введите результат спортсмена 8 (сек) – 8

Введите результат спортсмена 9 (сек) – 9



Консоль отладки Microsoft V



```
Введите результат спортсмена 8 (сек) - 8
Введите результат спортсмена 9 (сек) - 9
Введите результат спортсмена 10 (сек) - 10
Введите результат спортсмена 11 (сек) - 11
Введите результат спортсмена 12 (сек) - 12
Введите результат спортсмена 13 (сек) - 13
Введите результат спортсмена 14 (сек) - 14
Введите результат спортсмена 15 (сек) - 15
Введите результат спортсмена 16 (сек) - 16
Введите результат спортсмена 17 (сек) - 17
Введите результат спортсмена 18 (сек) - 18
Введите результат спортсмена 19 (сек) - 19
Введите результат спортсмена 20 (сек) - 20
Введите результат спортсмена 21 (сек) - 21
Введите результат спортсмена 22 (сек) - 22
Введите результат спортсмена 23 (сек) - 23
Введите результат спортсмена 24 (сек) - 24
Введите результат спортсмена 25 (сек) - 25
Введите результат спортсмена 26 (сек) - 26
Введите результат спортсмена 27 (сек) - 27
Введите результат спортсмена 28 (сек) - 28
Введите результат спортсмена 29 (сек) - 29
Введите результат спортсмена 30 (сек) - 30
```



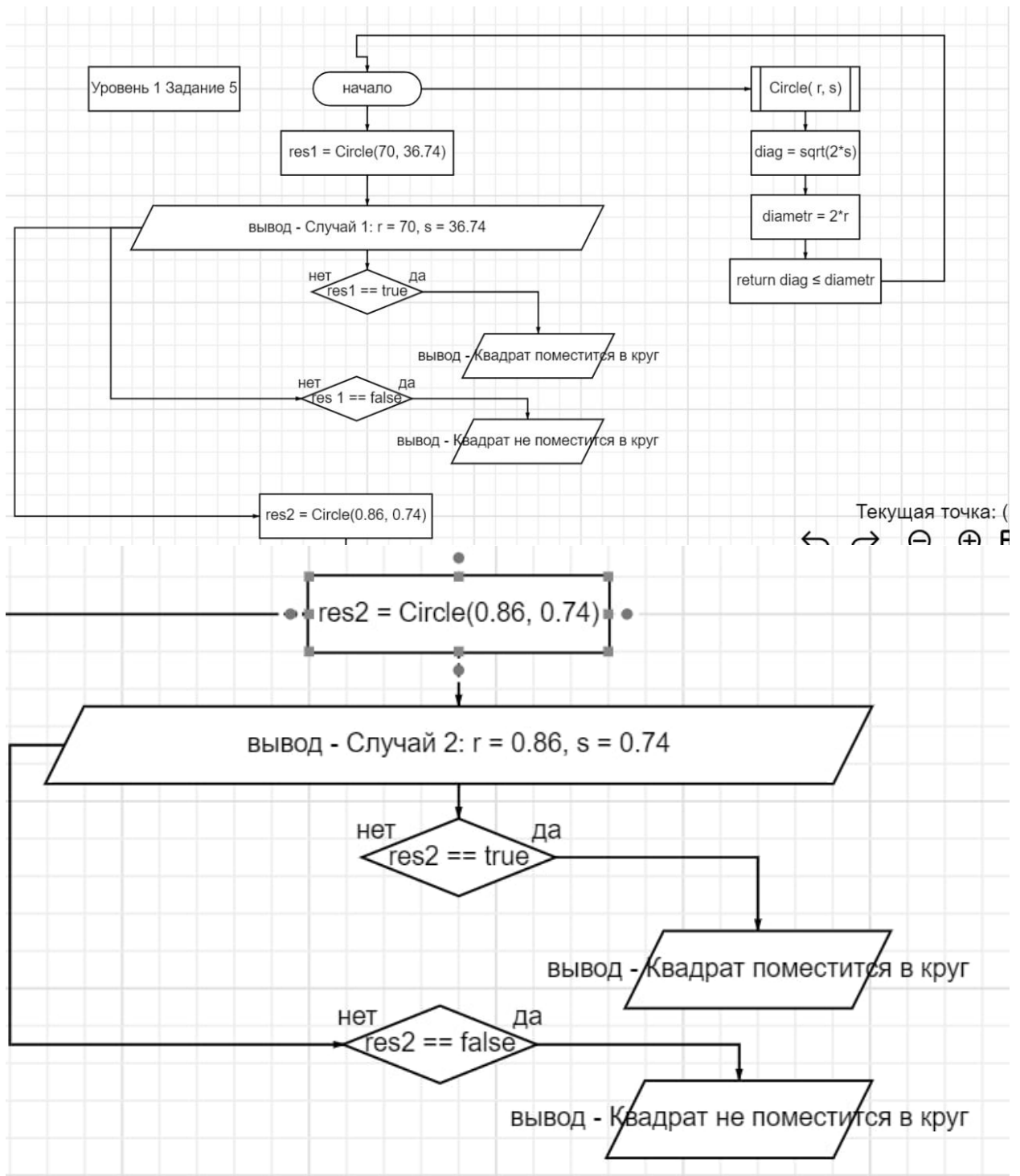
```
Консоль отладки Microsoft V  ×  +  ∨

Введите результат спортсмена 29 (сек) – 29
Введите результат спортсмена 30 (сек) – 30
Количество спортсменов выполнивших норматив – 10
Введите количество точек – 2
Введите x координату точки 1 – 1
Введите y координату точки 1 – 2
Количество точек принадлежащих фигуре – 0
Введите x координату точки 2 – 3
Введите y координату точки 2 – 9
Количество точек принадлежащих фигуре – 0
Введите количество точек – 2
Введите x координату точки 1 – 1
Введите y координату точки 1 – 2
Точка 1 находится в 1 квадранте
Введите x координату точки 2 – 5
Введите y координату точки 2 – 7
Точка 2 находится в 1 квадранте
Количество точек в 1-м квадранте – 2
Количество точек в 3-м квадранте – 0

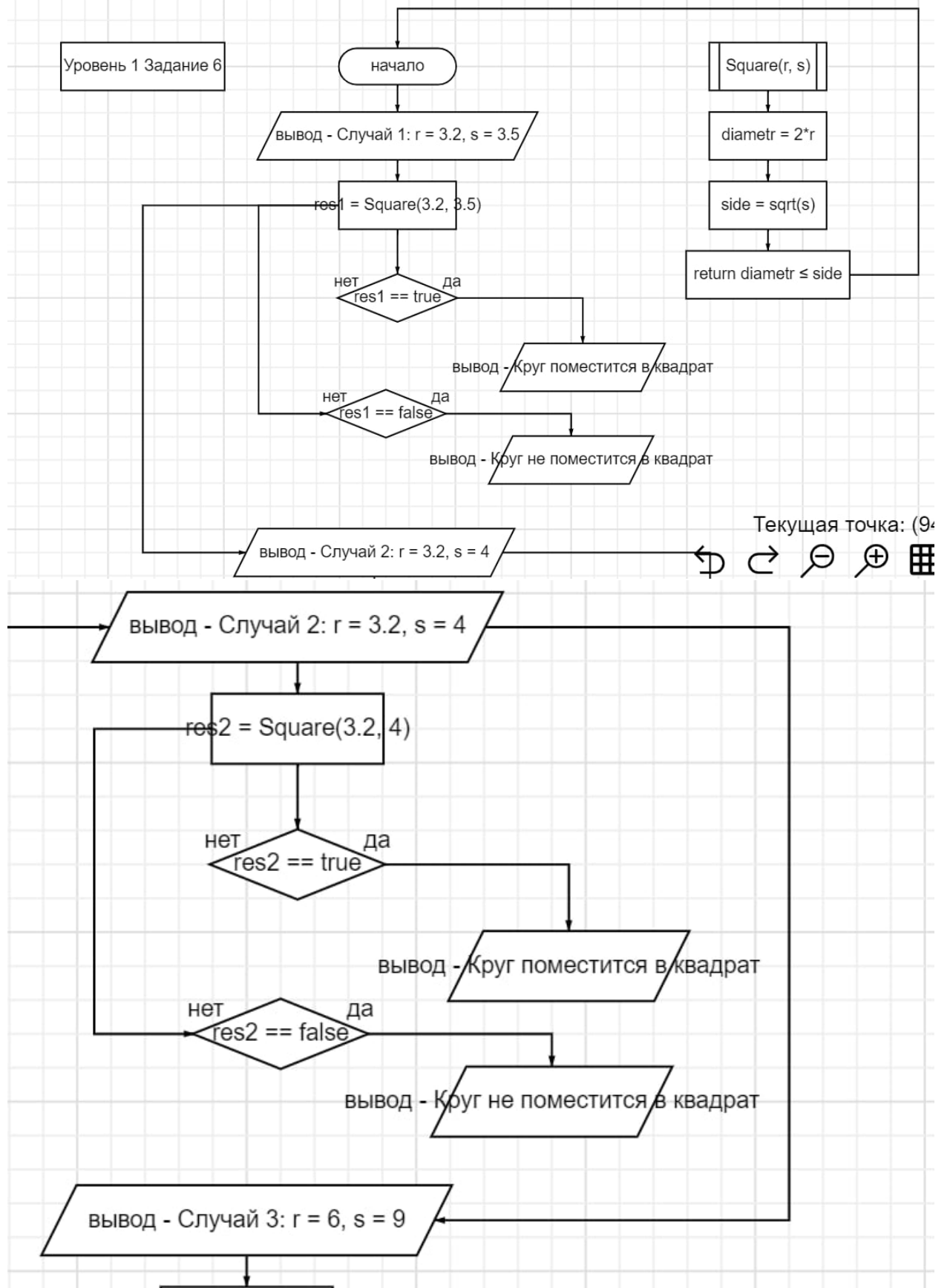
C:\Users\Lenovo\source\repos\ConsoleApp1\ConsoleAp
1416) завершил работу с кодом 0 (0x0).
Нажмите любую клавишу, чтобы закрыть это окно:|
```

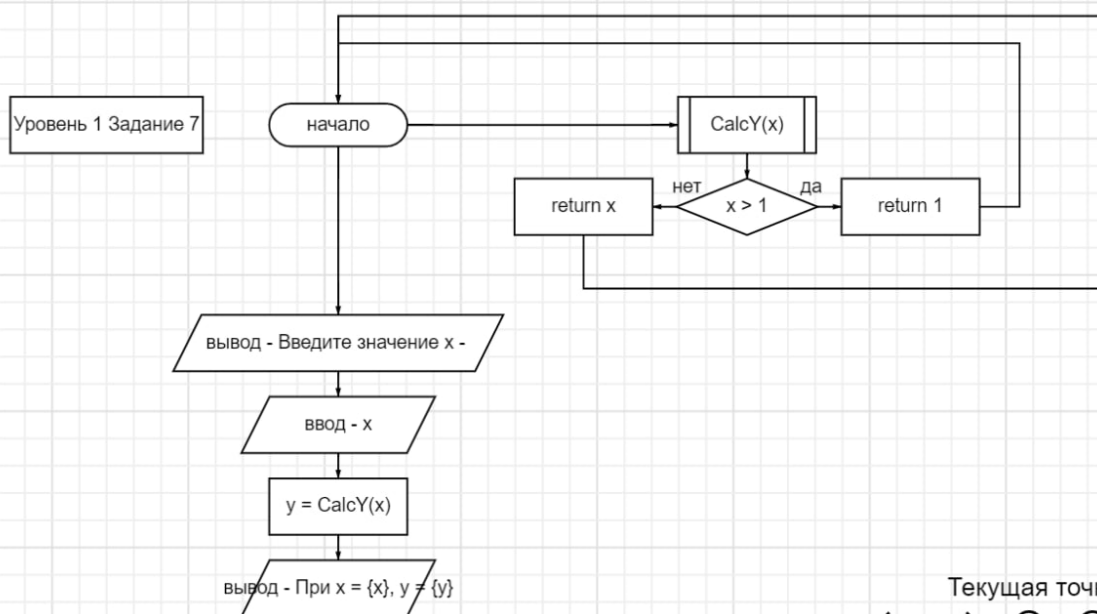
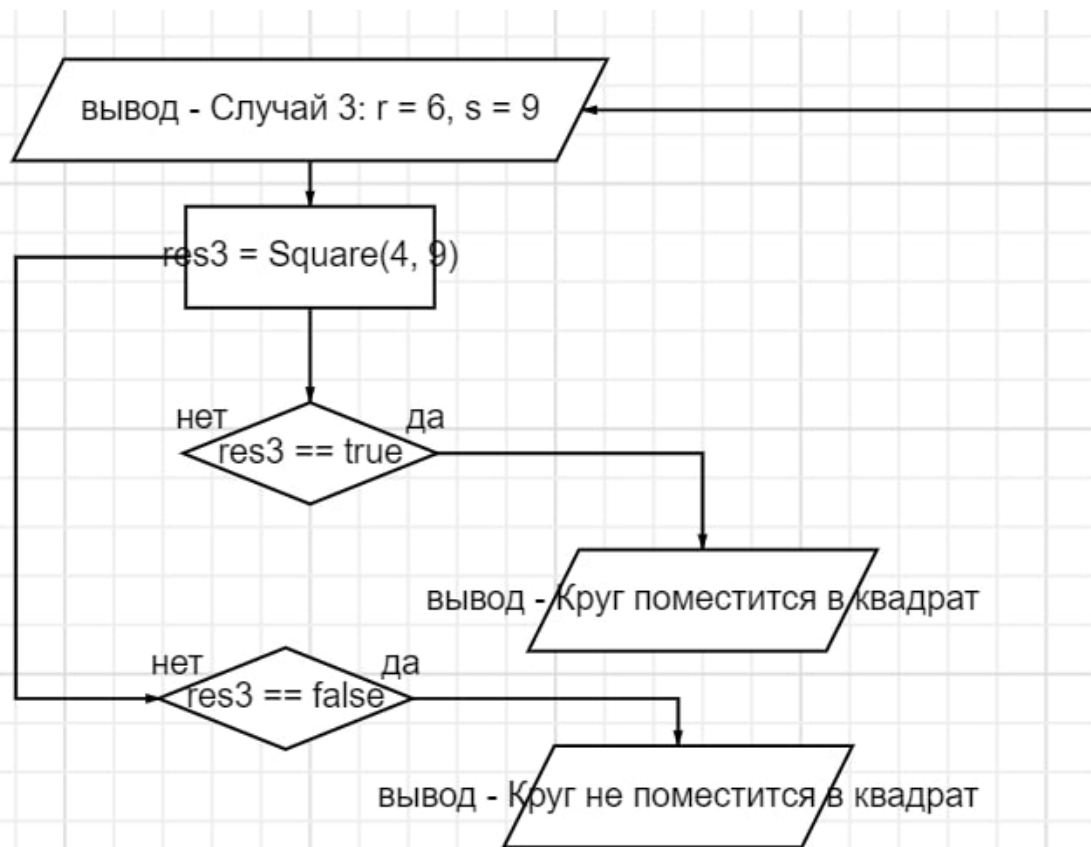
Блок-схемы для каждого задания:





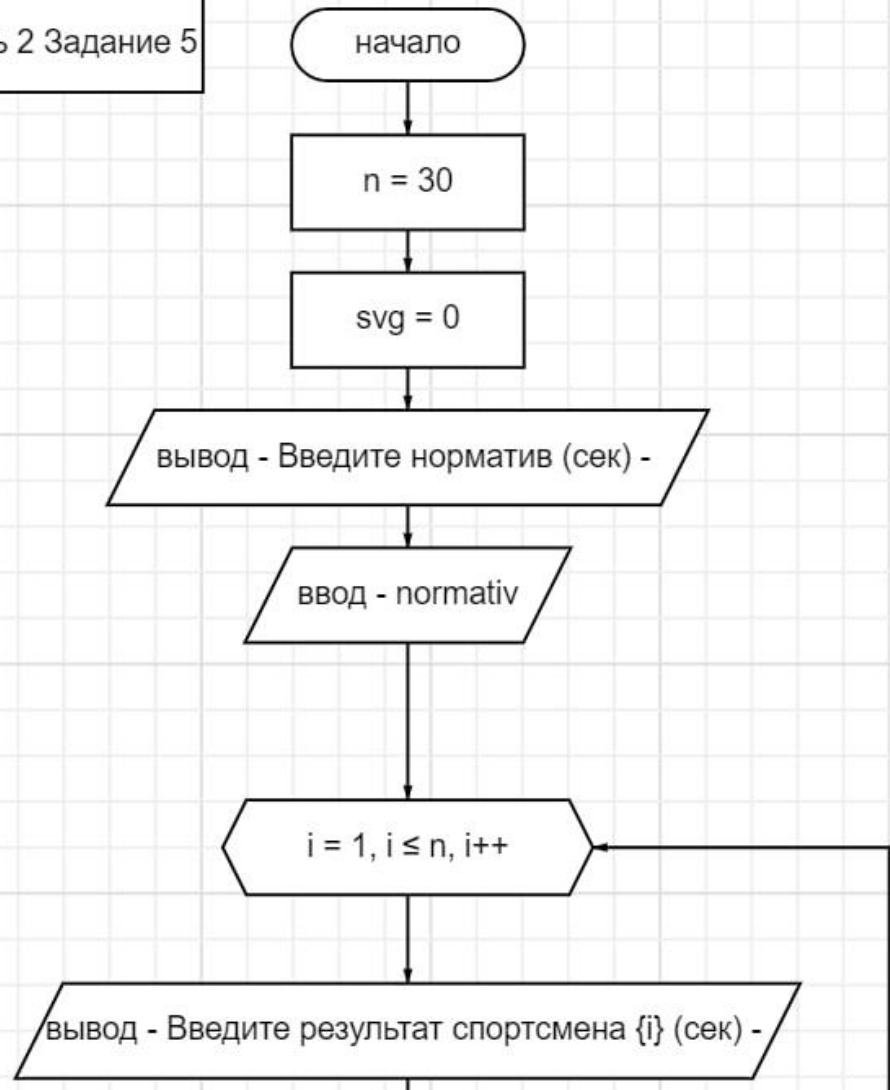
Уровень 1 Задание 6

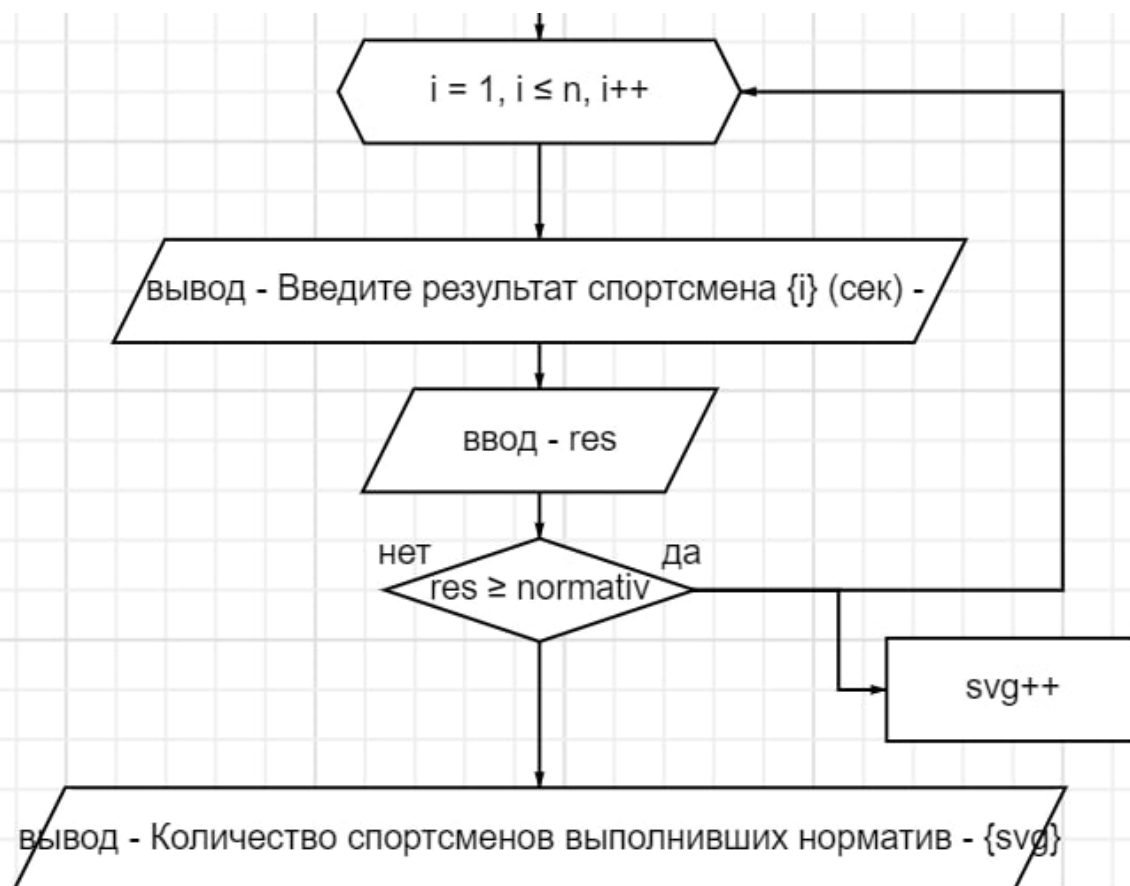




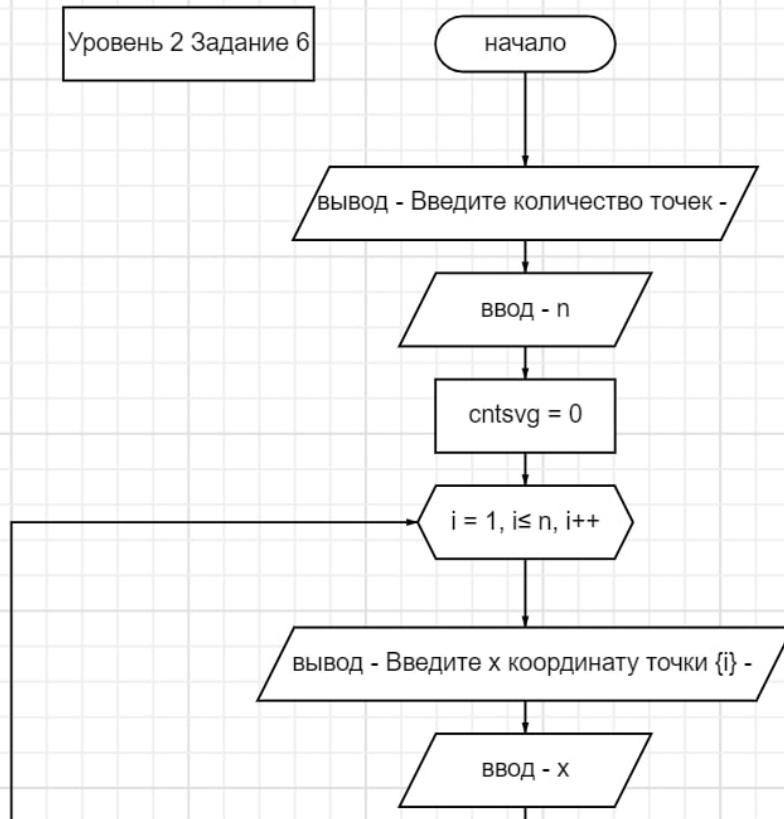
Текущая точка: (848,

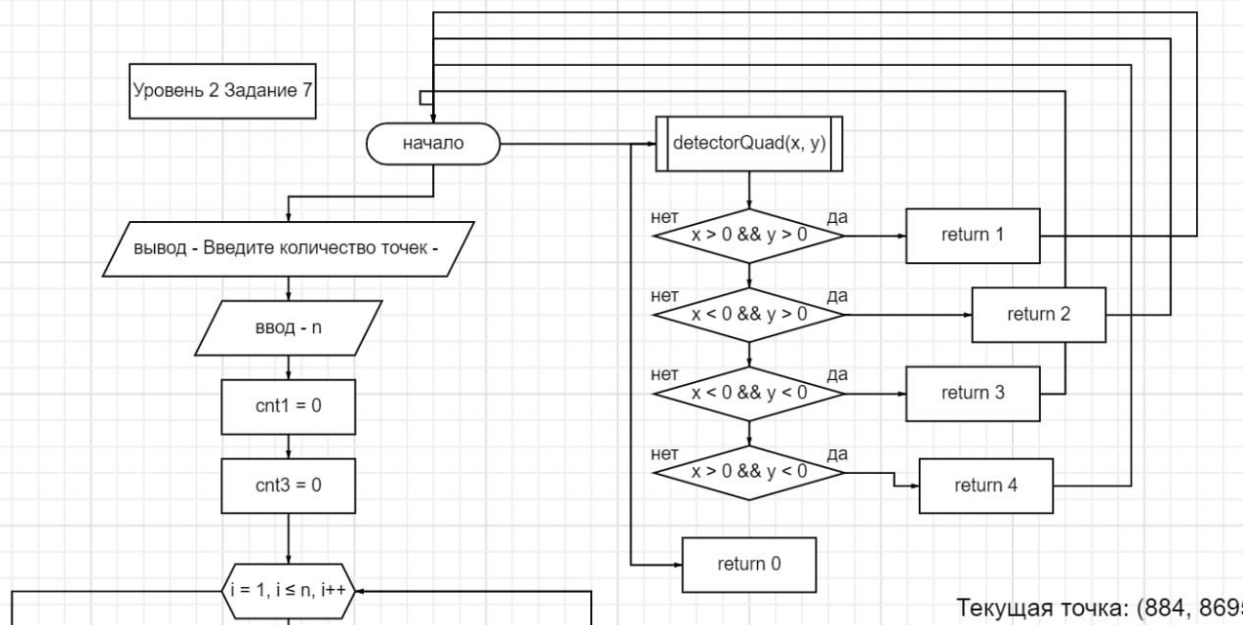
Уровень 2 Задание 5



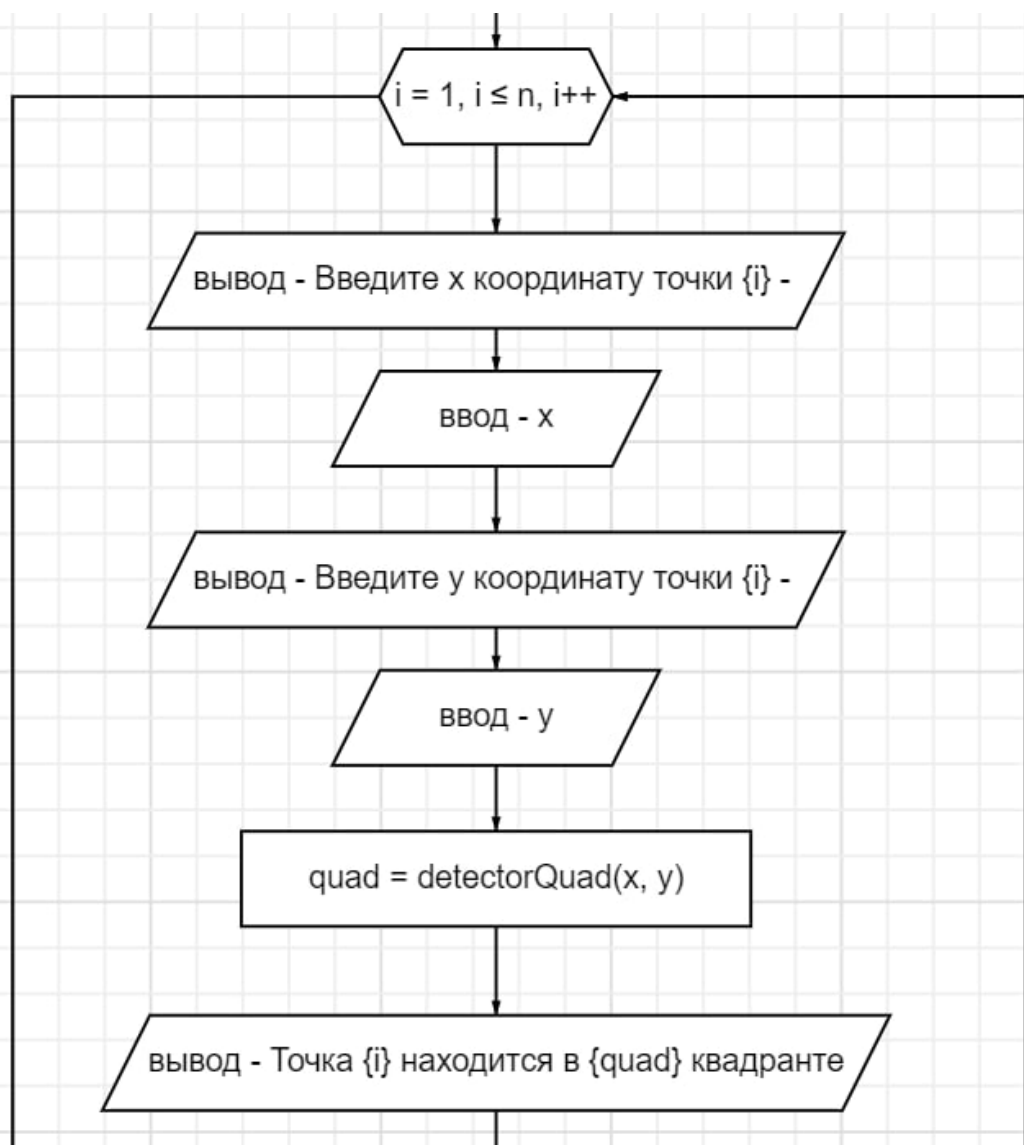


Уровень 2 Задание 6

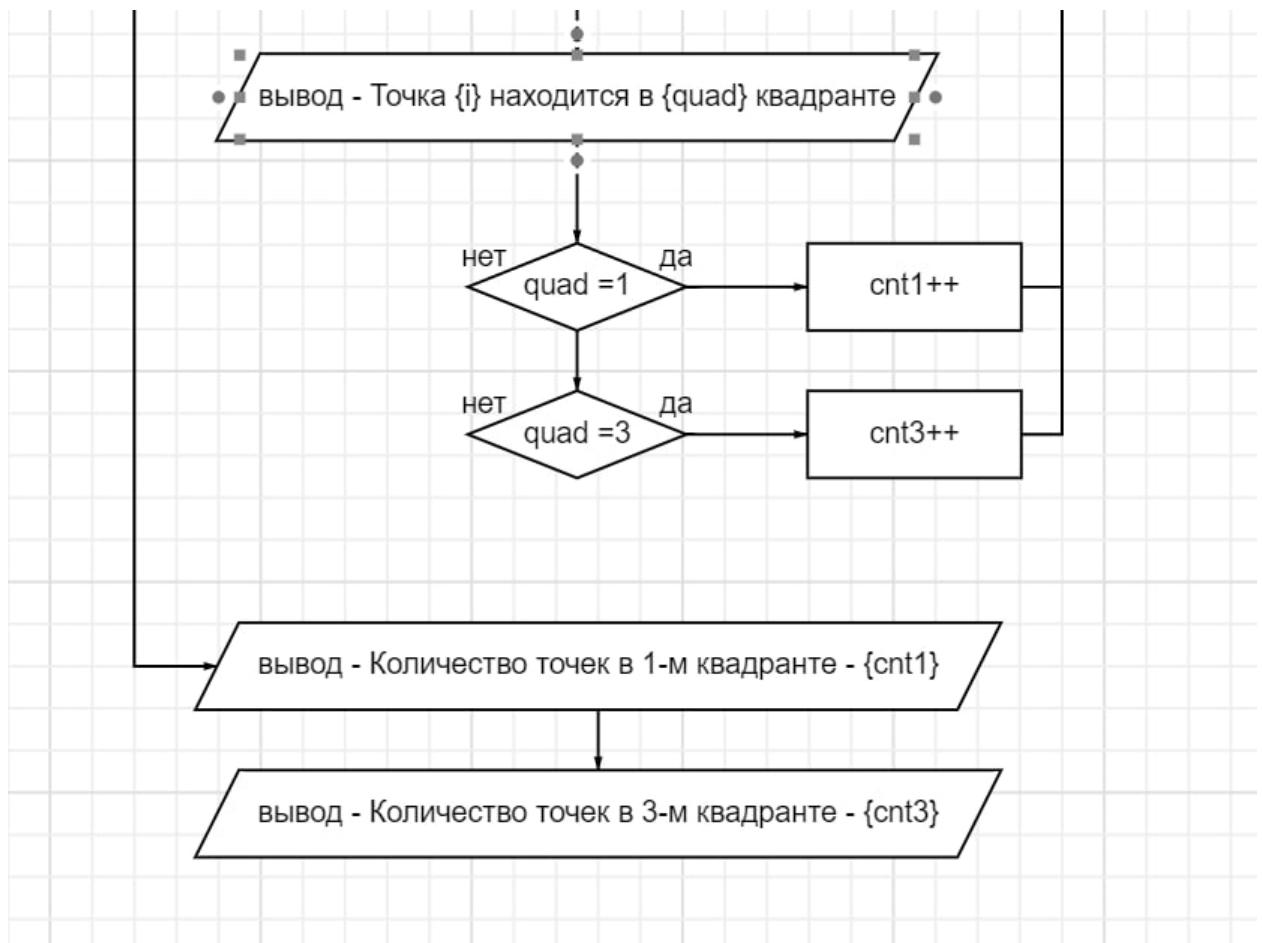




Текущая точка: (884, 869:







**Вывод:** в ходе лабораторной работы были освоены базовые принципы организации разветвлений и их интеграция с циклами в языке C#. Было изучено применение условного оператора if и оператора множественного выбора для обработки данных на основе различных условий. Полученные навыки позволяют создавать более гибкие и адаптивные программы, способные реагировать на изменение входных данных и выполнять соответствующие действия.