**Министерство науки и высшего образования Российской Федерации**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**ITMO University**

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

**По дисциплине** Объектно-ориентированное программирование

**Обучающийся** Зорина Яна Сергеевна

**Факультет** Факультет инфокоммуникационных технологий

**Группа** К3222

**Направление подготовки** 11.03.02 Инфокоммуникационные технологии и системы связи

**Образовательная программа** Программирование в инфокоммуникационных системах

**Обучающийся** 24.10.2023  Зорина Я.С.

(дата) (подпись) (Ф.И.О.)

**Руководитель**  Васильев С.Ю.

(дата) (подпись) (Ф.И.О.)

Санкт Петербург

2023

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc149073632)

[ГЛАВА 1. Ход работы 4](#_Toc149073633)

[1.1 Упражнение 1 4](#_Toc149073634)

[**1.1.1 Задание 1** 4](#_Toc149073635)

[**1.1.2 Задание 2** 4](#_Toc149073636)

[1.2 Упражнение 2 7](#_Toc149073637)

[**1.2.1 Задание 1** 7](#_Toc149073638)

[**1.2.2 Задание 2** 8](#_Toc149073639)

[**1.2.3 Задание 3** 9](#_Toc149073640)

# Введение

Целью лабораторной работы является знакомство с языком программирования C# и программой «Visual Studio», изучение структуры программы на данном языке, освоение компиляции и отладки.

Отчёт содержит одну главу, каждая подглава посвящена одному упражнению из лабораторной работы.

Для выполнения лабораторной работы использовалась программа «Visual Studio».

# ГЛАВА 1. Ход работы

## 1.1 Упражнение 1

### **1.1.1 Задание 1**

Эта программа представляет собой простое консольное приложение на языке программирования C#. Она принимает входные значения "x" и "y" от пользователя, затем в зависимости от условий, проверяет находятся ли эти значения внутри, вне или на границе заданной области.

- Если "x" и "y" находятся внутри окружности с радиусом 3 и "y" положителен, программа выводит сообщение "внутри".

- Если "x" и "y" находятся вне окружности с радиусом 3 или "y" отрицателен, программа выводит сообщение "вне".

- Если "x" и "y" находятся на границе окружности с радиусом 3, программа выводит сообщение "на границе".

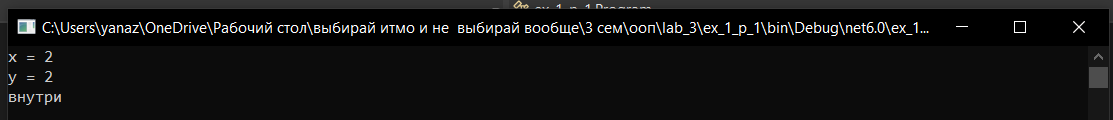
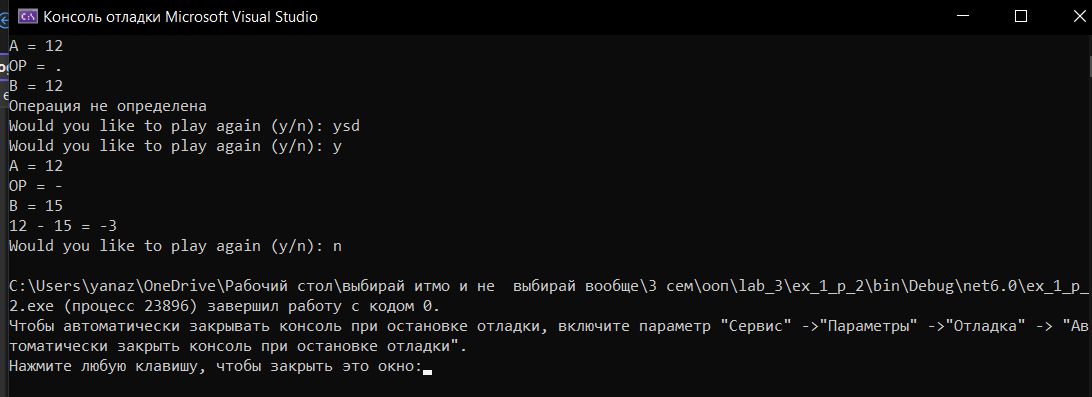
Программа использует условные операторы if-else для определения расположения точки относительно заданной области и выводит соответствующий результат на экран консоли. Результат запуска представлен на рисунке 1.

Рисунок 1. Задание 1 в упражнении 1

### **1.1.2 Задание 2**

Программа выполняет простые арифметические операции с двумя числами, вводимыми пользователем. Однако она не будет обрабатывать ситуации, когда происходит деление на ноль или вводится неправильный символ операции.

Проверка деления на ноль: Если пользователь вводит символ операции "/", программа выполнит деление первого числа (a) на второе число (b). Если второе число равно нулю (b = 0), то вместо результата программа выведет вопросительный знак. Это происходит потому, что в математике деление на ноль не определено.

Неправильный символ операции: Если пользователь вводит символ операции, отличный от "+", "-", "\*", "/" или ":", программа присваивает переменной ok значение false, что означает, что операция не определена. Затем программа выводит сообщение "Операция не определена".

Результат представлен на рисунке 2.

Рисунок 2. Результат выполнения операций

**1.1.3 Задание 3**

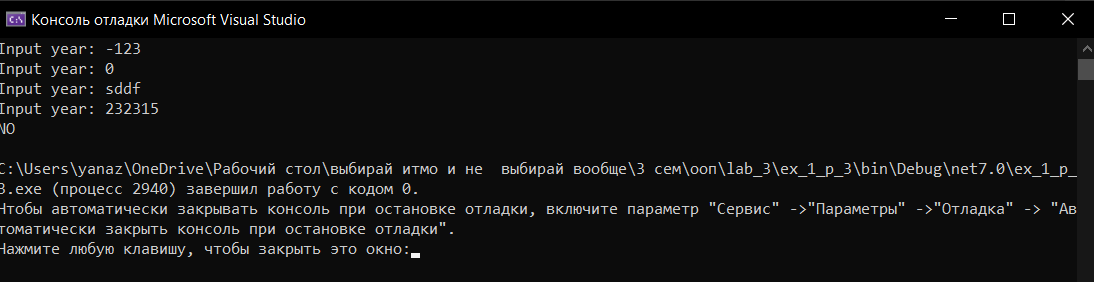
Программа простого консольного приложения, которое запрашивает у пользователя ввод года. Затем она проверяет введенное значение, чтобы убедиться, что оно является натуральным числом. Если ввод корректен, программа определяет, является ли год високосным, и выводит "YES" или "NO" в зависимости от результата. Результат выполнения программы представлен на рисунке 3.

Рисунок 3. Результат проверки года на високосность

## 1.2 Упражнение 2

### **1.2.1 Задание 1**

Данный код предлагает пользователю выбрать одну из трех опций: "cycle thing", "sin thing" или "evklid thing".

При выборе опции "cycle thing", программа запрашивает у пользователя ввод числа `n`, а затем использует цикл `while`, `do while` и `for` для вывода на консоль нечетных чисел от 1 до `n`. После завершения вывода программа возвращает пользователя в основное меню.

При выборе опции "sin thing", программа запрашивает у пользователя ввод чисел `x1` и `x2`. Затем с помощью цикла `do while` программа вычисляет значения синуса для значений `x` от `x1` до `x2` с шагом 0,01 и выводит их на консоль. Результаты округляются до двух знаков после запятой. После завершения вывода программа возвращает пользователя в основное меню.

При выборе опции "evklid thing", программа запрашивает у пользователя ввод двух целых чисел `a` и `b`. Затем с помощью цикла `while` программа находит наименьшее общее кратное (НОК) этих чисел с помощью алгоритма Евклида. Результат выводится на консоль. После вывода результата программа возвращает пользователя в основное меню.

Цикл `Main` повторяется бесконечно, позволяя пользователю выбирать и выполнять определенные опции. Если пользователь вводит значение, отличное от 1, 2 или 3, программа завершается.

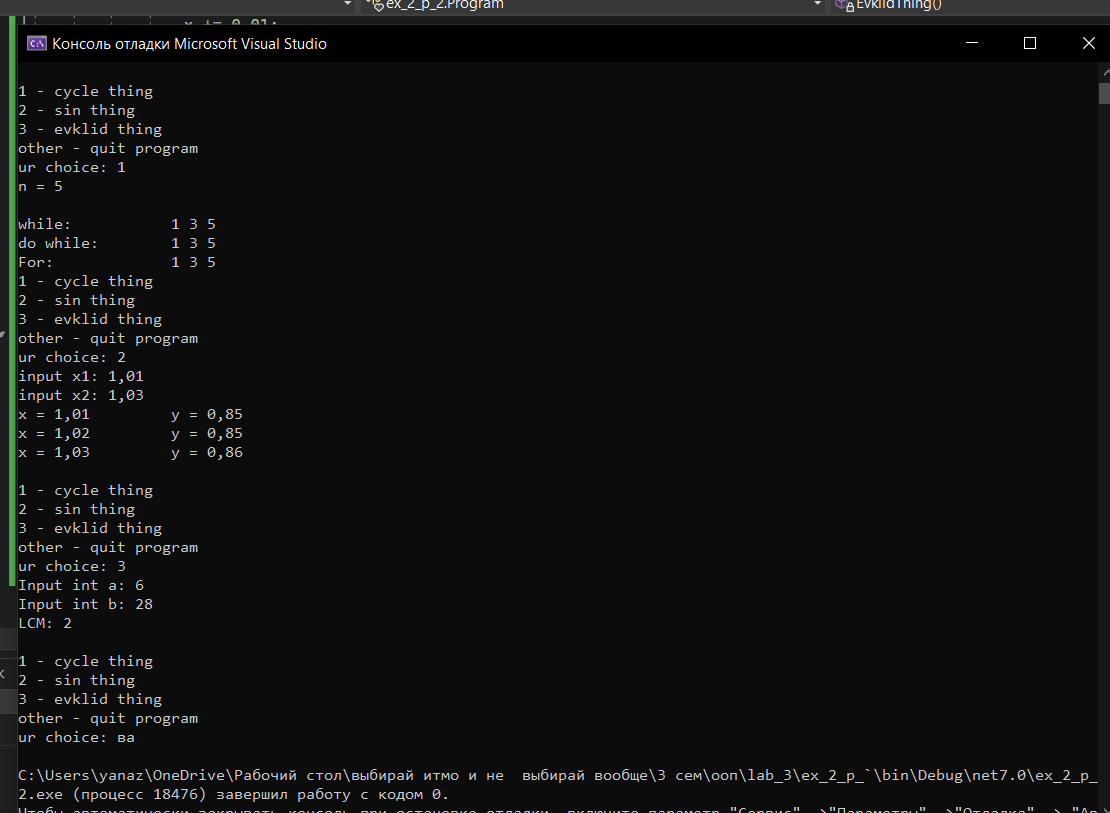
Выбор типа цикла зависит от конкретной задачи и требований к решению. В данном случае, использование разных типов циклов было обусловлено диапазоном значений, которые должны были быть обработаны, и удобством в использовании. Цикл while был выбран для обработки диапазона значений, когда заранее нельзя определить точное количество итераций. Цикл do while был использован для выполнения хотя бы одной итерации и продолжения выполнения в зависимости от условия. Цикл for был выбран для обработки диапазона значений с явными условиями начала, окончания и шага выполнения.

Рисунок 4. Результат выполнения всех трёх опций

Результат выполнения программы представлен на рисунке 4.

### **1.2.2 Задание 2**

Данный код запрашивает у пользователя ввод двух чисел `k` и `m`. Затем программа выполняет цикл от 1 до 100, пропуская числа, которые находятся между `k` и `m`. Каждое число, не попадающее в этот диапазон, добавляется к переменной `result`. По завершению цикла, программа выводит сумму всех чисел, которые не находятся между `k` и `m`, на консоль. Результат выполнения программы представлен на рисунке 5.

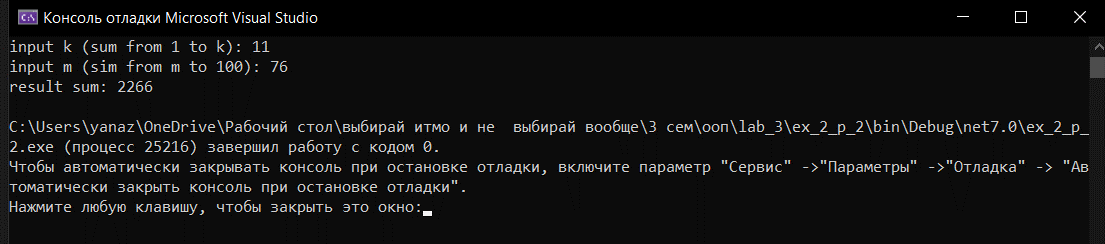
**1.2.3 Задание 3**

Рисунок 5. Результат суммирования по заданным условиям

Программа, представленная в задании, имитирует стрельбу по мишени. Она позволяет пользователю вводить координаты выстрела (x и y) несколько раз. После каждого выстрела выводится информация о сумме набранных очков.

Функциональность программы включает:

- Повторный ввод данных о выстреле в цикле.

- Использование случайного значения для определения центра мишени, чтобы стрелок не знал точного местоположения мишени.

- Введение случайной помехи при выстреле, чтобы усложнить задачу для стрелка.

Программа также задает некоторые константы, определяющие диапазон координат и значения очков за попадание.

Количество попыток стрельбы ограничено числом 5. После окончания всех попыток выводятся полученные очки, а затем игроку предлагается продолжить или завершить игру.

Игрок получает очки в зависимости от расстояния между точкой выстрела и центром мишени. Если расстояние меньше определенного значения, игрок получает соответствующее количество очков.

Наконец, программа отслеживает общее количество очков игрока и выводит его после каждого раунда.

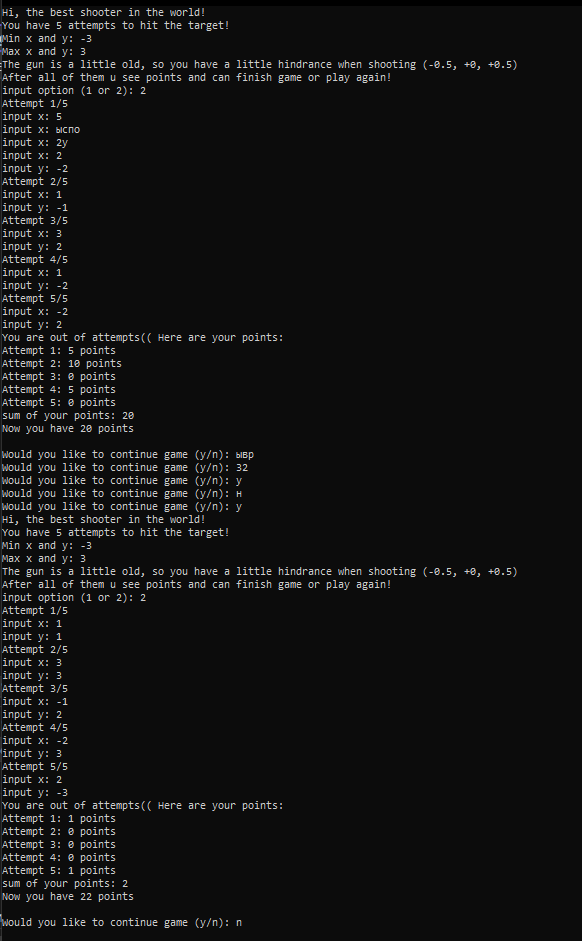
Результат работы программы представлен на рисунке 6.

Рисунок 6. Результат игры со стрельбой