**Федеральное государственное автономное образовательное учреждение высшего образования   
"Национальный исследовательский университет   
"Высшая школа экономики"**

*Факультет социально-экономических и компьютерных наук*

Чепкасов Виктор Алексеевич

**ЛАБОРАТОРНАЯ РАБОТА № 1**

*Вариант №24*

студента образовательной программы «Разработка информационных систем для бизнеса» по направлению подготовки *38.03.05 Бизнес-информатика*

Доцент  
кафедры ИТБ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

О.Л. Викентьева

Пермь, 2025 год**Оглавление**

[1 Задание № 1 3](#_Toc210601156)

[1.1 Постановка задачи 3](#_Toc210601157)

[1.2 Анализ 3](#_Toc210601158)

[1.3 Проектирование 5](#_Toc210601159)

[1.4 Разработка 6](#_Toc210601160)

[2. Задание № 2 8](#_Toc210601161)

[2.1 Постановка задачи 8](#_Toc210601162)

[2.2 Анализ 8](#_Toc210601163)

[2.3 Проектирование 10](#_Toc210601164)

[2.4 Разработка 10](#_Toc210601165)

[3. Задание №3 12](#_Toc210601166)

[3.1 Постановка задачи 12](#_Toc210601167)

[3.2 Анализ 12](#_Toc210601168)

[3.3 Проектирование 13](#_Toc210601169)

[3.4 Разработка 13](#_Toc210601170)

[Приложение А Код программы для задачи № 1 14](#_Toc210601171)

[Приложение Б Код программы для задачи № 2 20](#_Toc210601172)

[Приложение В Код программы для задачи № 3 22](#_Toc210601173)

# Задание № 1

Первое задание лабораторной работы направлено на то, чтобы научиться работать с операторами языка С#. Оно проверяет понимание операций инкремента и декремента, а также учит анализировать выражение для вывода правильной ошибки.

## Постановка задачи

Общая постановка задачи:

1. Найти значения выражений, обработав исходные данные.
2. Составить систему тестов и вычислить полученное выражение для разных X.
3. Результаты вывести на печать в консоль.

Частная постановка задачи состоит в том, чтобы для выражений 1-3 посчитать их значения после ввода пользователем m и n. Область определения в 1-3 выражениях нас не интересует. Для выражений 2 и 3 вывести результат сравнения. Для 4 выражения необходимо вычислить его значение. Область определения нас не интересует, так как область определения все числа.

Конкретные задачи:

1. m - ++n
2. m++ > --n
3. m-- < ++n
4. .

## Анализ

В 1-3 выражениях пользователь вводит данные типа int, в 4 типа double.

Анализ классов входных и выходных данных приведён в таблицах 1, 2, 3. Для каждой переменной создана своя таблица.

Стандартные выходные данные для первых 3-х заданий (ожидаемый вывод):

1. string
2. string
3. string.

Таблица 1 ⎯ Классы входных и выходных данных для переменной n

| **Классы входных данных** | **Условия** | **Классы выходных данных** |
| --- | --- | --- |
| *n* - строка | Невозможно вычислить | Сообщение об ошибке из-за неправильного типа введённых данных |
| *n* – вещественное число или целое за пределами Int32 | Невозможно вычислить | Сообщение об ошибке из-за неправильного типа введённых данных |
| *n* – любое целое число в пределах типа Int32 | Программа выполняется | Стандартные выходные данные |

Похожая таблица, сделанная для переменной m, представлена ниже.

Таблица 2 ⎯ Классы входных данных для переменной m

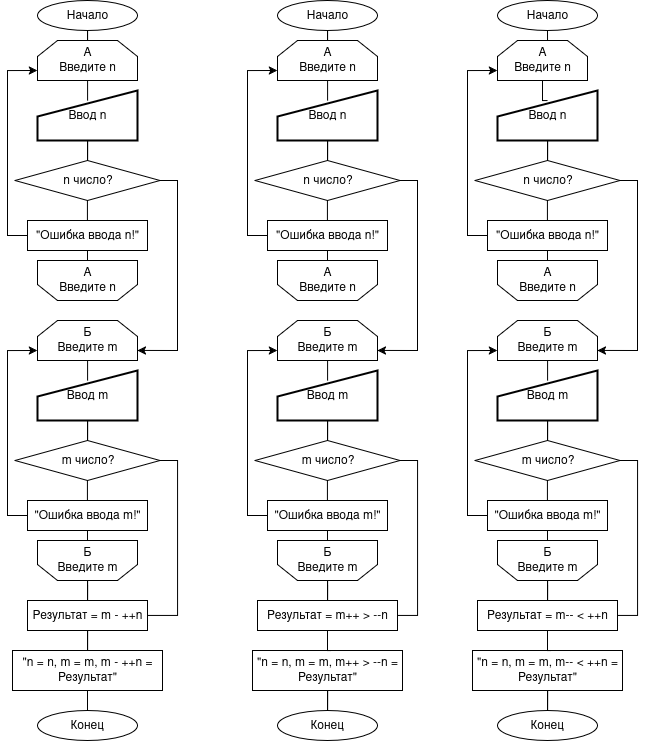
| **Классы входных данных** | **Условия** | **Классы выходных данных** |
| --- | --- | --- |
| *m* - строка | Невозможно вычислить | Сообщение об ошибке из-за неправильного типа введённых данных |
| *m* – вещественное число или целое за пределами Int32 | Невозможно вычислить | Сообщение об ошибке из-за неправильного типа введённых данных |
| *m* – любое целое число в пределах типа Int32 | Программа выполняется | Стандартные выходные данные |

В 4-м задании вводится переменная x типа double. Ожидаемый стандартный вывод для 4-го задания – число типа double. Ниже приведена таблица анализа классов входных и выходных данных для переменной x. В качестве sin(x) используется число в радианах. Переменная x может быть любым вещественным числом, поэтому другие критерии рассматриваться не будут. В задании у нас есть дробь, но её знаменатель никогда не будет равен 0.

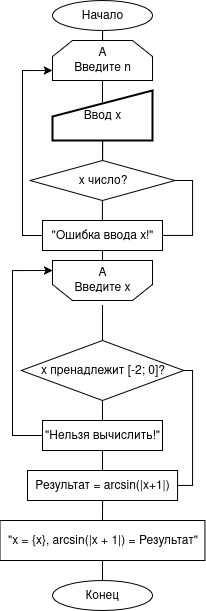
Таблица 3 ⎯ Классы входных данных для переменной x

| **Классы входных данных** | **Условия** | **Классы выходных данных** |
| --- | --- | --- |
| *x* – строка или вещественное число за пределами типа double | Невозможно вычислить | Сообщение об ошибке из-за неправильного типа введённых данных |
| *x* >= 0 | Программа выполняется | Стандартные выходные данные |
| *x* < 0 | Программа выполняется | Стандартные выходные данные |

## Проектирование

Для реализации проектирования необходимо составить блок схемы, они представлены на рисунке 1 для выражений 1-3 (слева – 1, по середине – 2, справа - 3) и на рисунке 2 для задачи 4.

*Рисунок 1 ⎯ схемы алгоритмов для выражений 1, 2 и 3 соответственно*

Рисунок 2 ⎯ схемы алгоритмов для выражений 1, 2 и 3 соответственно

## Разработка

Программы считывает значения переменных и считает выражения 1-4. Код программа находится в приложении А

Здесь объявлены переменные int m, int n, которые будут использованы в выражении 1, 2 и 3. Для того, чтобы данная конструкция получения данных для переменных работала бесконечное количество раз, был создан цикл while. После вычисления всех выражений пользователь увидит результат выражений и состояния переменных. Во всех выражения внутри выполняется проверка на переполнение типов данных в переменных.

Выражение 1.

Далее следует цикл while, в котором методом int.Parse мы пытаемся преобразовать строку в int. При успешной конвертации цикл прерывается командой break. Если это не удаётся, то пользователю выводится сообщение об ошибке, после чего требуется повторный ввод.

Далее идёт такой же цикл, но в отношении переменной m.

Затем в переменной r1 вычисляется значение выражения m - ++n. Пользователю выводится результат выражения.

Выражение 2.

Здесь использованы такие же циклы, как и в прошлом выражении.

Результат выражения m++ > --n помещается в переменную r2 и выводится пользователю.

Выражение 3.

Здесь использованы такие же циклы, как и в предыдущих двух заданиях.

Результат выражения m-- < ++n помещается в переменную r3 и выводится пользователю

Выражение 4.

Переменной x присваивается значение double. Далее тем же циклом происходит преобразование строки в double. После чего с помощью Math происходит вычисление выражения и вывод его результата пользователю.

# Задание № 2

Второе задание посвящено получению навыков работы с операторами языка С# и работы со сложными условиями.

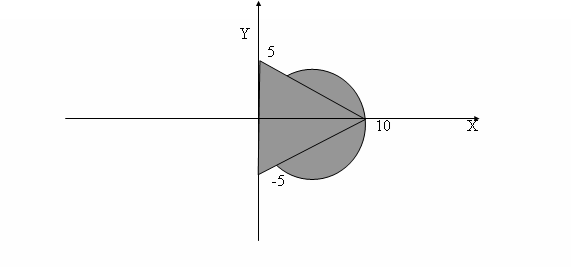
## Постановка задачи

Общая постановка задачи:

1. Определить принадлежит ли точка области, обработав исходные данные.
2. Составить систему тестов и проверить на разных x и y.
3. Результаты вывести на печать.

Частная постановка задачи.

Сравнить координаты x и y с данной нам областью и вывести пользователю ответ, принадлежит точка этой области или нет.

Рисунок 3 ⎯ Постановка задачи 2

Конкретная задача расположена на рисунке 3

## Анализ

Во втором задании пользователь вводит числа в переменные x и y, которым присвоены значения double. Затем проверяется принадлежность точки с координатами (x; y) к области, представляющей собой два треугольника и окружность.

В таблицах 4 и 1 предоставлены классы входных и выходных данных для переменных x и y.

Таблица 4 – Классы входных и выходных данных для переменной x

| **Классы входных данных** | **Условия** | **Классы выходных данных** |
| --- | --- | --- |
| *x* - строка | Невозможно вычислить | Сообщение об ошибке из-за неправильного типа введённых данных |
| *x* – вещественное число за пределами double | Невозможно вычислить | Сообщение об ошибке из-за неправильного типа введённых данных |
| *x* – любое число в пределах типа double | Программа выполняется | Стандартные выходные данные |

Таблица 5 – Классы входных и выходных данных для переменной y

| **Классы входных данных** | **Условия** | **Классы выходных данных** |
| --- | --- | --- |
| *y* - строка | Невозможно вычислить | Сообщение об ошибке из-за неправильного типа введённых данных |
| *y* – вещественное число за пределами double | Невозможно вычислить | Сообщение об ошибке из-за неправильного типа введённых данных |
| *y* – любое число в пределах типа double | Программа выполняется | Стандартные выходные данные |

## Проектирование

Для реализации проектирования необходимо составить блок схему, она представлена на рисунке 1.

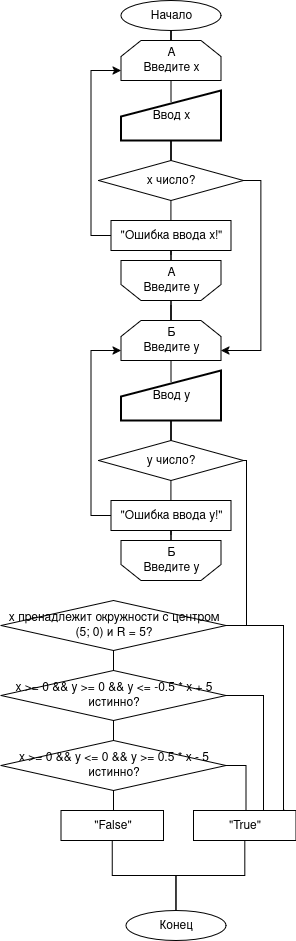


Рисунок 4 ⎯ Блок схема задачи 2

## Разработка

Код программы находится в приложении Б. В начале программы создаются переменные x и y и присваиваются значение double. Далее повторяется цикл по валидации данных, если нет, то выводится ошибка.

Далее с помощью условия if проверяем принадлежность одной из трёх фигур на рисунке 3.

После чего на основе результата сравнения x и y выводится сообщение о том, принадлежит ли точка в области или нет.

# Задание №3

Задание 3 направлено на анализ типов данных, которыми мы оперируем и анализ результатов, которые мы получаем на основе тех типов данных, которые используем.

## Постановка задачи

Общая постановка задачи:

1. Посчитать значение выражения присвоив разные вещественные типы данных a и b.
2. Результат всех вычислений вывести на печать.
3. Проанализировать и объяснить полученные результаты.

Частная постановка задачи.

Посчитать значение выражения в разных типах данных при заданных a = 1000, b =0.0001.

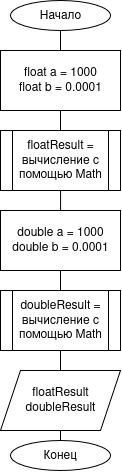
## Анализ

Входных данных нет, переменные объявлены в виде констант. В качестве выходных данных получится число вещественного типа. Результаты в типе float и double будут не одинаковыми, из-за того, что в какой-то момент тип float, за счет 4 битов вместо 8 типа double, обрежет часть числа, что приведёт к потере точности и итоговое выражение будет иметь значительно большую погрешность в расчёте.

Так как в задании содержаться формулы сокращённого умножений, данная формула после элементарного раскрытия скобок и сокращения принимает вид единицы.

## Проектирование

Для дальнейшей реализации проектирования и последующей разработки программы составлена блок схема, представленная ниже на рисунке 5.

Рисунок 5 ⎯ Блок схема задачи 3

## Разработка

Код программы находится в приложении В. Для демонстрации различной точности типов данных float и double выражение будет считаться в несколько этапов, каждый их которых хранится в отдельной переменной типа float или double соответственно

Создаются переменные a, b, floatResult типа float, с помощью Math вычислется значение выражения. Далее создаются переменные a1, b1, doubleResult типа double, с помощью Math так же вычисляются значения выражений.

Далее выводятся результаты и объясняются различия.

# Приложение А Код программы для задачи № 1

using System;

namespace Task1

{

internal class Program

{

static void Main(string[] args)

{

int m, n, r1;

bool r2, r3;

double x, r4;

//ввод данных

while (true)

{

try

{

System.Console.Write("n ? ");

string buff = Console.ReadLine();

n = int.Parse(buff);

break;

}

catch (FormatException)

{

System.Console.WriteLine("Ошибка ввода n!");

}

catch (OverflowException)

{

System.Console.WriteLine("Значение n вышло за границы заданного типа!");

}

};

while (true)

{

try

{

System.Console.Write("m ? ");

string buff = Console.ReadLine();

m = int.Parse(buff);

break;

}

catch (FormatException)

{

System.Console.WriteLine("Ошибка ввода m!");

}

catch (OverflowException)

{

System.Console.WriteLine("Значение m вышло за границы заданного типа!");

}

};

while (true)

{

try

{

System.Console.Write("x ? ");

string buff = Console.ReadLine();

x = double.Parse(buff);

break;

}

catch (FormatException)

{

System.Console.WriteLine("Ошибка ввода x!");

}

};

//вычисления

try

{

checked

{

try

{

checked { ++n; }

}

catch (Exception)

{

System.Console.WriteLine("Значение n вышло за границы заданного типа!");

}

r1 = m - n;

System.Console.WriteLine($"n = {n}, m = {m}, m - ++n = {r1}");

}

}

catch (OverflowException)

{

System.Console.WriteLine("Невозможно вычислить разницу в заданном типе!");

}

catch (ArgumentOutOfRangeException)

{

Console.WriteLine("Невозможно вычислить разность в заданном типе!");

}

try

{

checked

{

r2 = m++ > --n;

System.Console.WriteLine($"n = {n}, m = {m}, m++ > --n = {r2}");

}

}

catch (OverflowException)

{

System.Console.WriteLine("Невозможно вычислить разницу в заданном типе!");

}

try

{

checked {

try { checked { ++n; }} catch (OverflowException) {System.Console.WriteLine("Значение n вышло за границы заданного типа!");}

r3 = m < n;

try { checked { m--; }} catch (OverflowException) {System.Console.WriteLine("Значение m вышло за границы заданного типа!");}

System.Console.WriteLine($"n = {n}, m = {m}, m-- < ++n = {r3}");

}

}

catch (Exception)

{

System.Console.WriteLine("Невозможно вычислить разницу в заданном типе!");

}

if (x < -2 || x > 0)

{

System.Console.WriteLine("Нельзя вычислить!");

}

else

{

try

{

checked {

r4 = System.Math.Asin(System.Math.Abs(x + 1));

System.Console.WriteLine($"x = {x}, arcsin(|x + 1|) = {r4}");

}

}

catch (OverflowException) {

System.Console.WriteLine("Число слишком большое/малое для вычисления!");

}

catch (Exception)

{

Console.WriteLine("Нельзя вычислить!");

}

}

}

}

}

# Приложение Б Код программы для задачи № 2

namespace System

{

internal class Program

{

static void Main(string[] args)

{

double x, y;

bool isConvert, inCircle, inTri1, inTri2;

//ввод данных

do

{

System.Console.Write("x ? ");

string buff = System.Console.ReadLine();

isConvert = double.TryParse(buff, out x);

if (isConvert == false)

System.Console.WriteLine("Ошибка ввода x!");

} while (isConvert == false);

do

{

System.Console.Write("y ? ");

string buff = System.Console.ReadLine();

isConvert = double.TryParse(buff, out y);

if (isConvert == false)

System.Console.WriteLine("Ошибка ввода y!");

} while (isConvert == false);

//определяем, входит ли переменная в область

inCircle = Math.Pow(5 - x, 2) + Math.Pow(0 - y, 2) <= 25;

inTri1 = x >= 0 && y >= 0 && y <= -0.5 \* x + 5;

inTri2 = x >= 0 && y <= 0 && y >= 0.5 \* x - 5;

System.Console.Write(inCircle || inTri1 || inTri2);

}

}

}

# Приложение В Код программы для задачи № 3

namespace Task3

{

internal class Program

{

static void Main(string[] args)

{

const float floatA = 1000;

const float floatB = 0.0001F;

const double doubleA = 1000;

const double doubleB = 0.0001;

//считаем выражение с float

float floatR1 = (float)Math.Pow(floatA - floatB, 3);

float floatR2 = (float)Math.Pow(floatA, 3) + 3 \* floatA \* floatB \* floatB;

float floatR3 = -3 \* (float)Math.Pow(floatA, 2) \* floatB - (float)Math.Pow(floatB, 3);

float floatResult = (floatR1 - floatR2) / floatR3;

//считаем выражение с double

double doubleR1 = Math.Pow(doubleA - doubleB, 3);

double doubleR2 = Math.Pow(doubleA, 3) + 3 \* doubleA \* doubleB \* doubleB;

double doubleR3 = -3 \* Math.Pow(doubleA, 2) \* doubleB - Math.Pow(doubleB, 3);

double doubleResult = (doubleR1 - doubleR2) / doubleR3;

System.Console.WriteLine($"Результат вычисления (double) = {doubleResult}");

System.Console.WriteLine($"Результат вычисления (float) = {floatResult}");

}

}

}