Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра Информатики

Дисциплина «Программирование»

**ОТЧЕТ**

к лабораторной работе №2

на тему:

**«Управление потоком выполнения программы**

БГУИР 6-05-0612-02 05

|  |
| --- |
| Выполнила студент группы 353504  АНТОНОВА Лидия Сергеевна |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 задание

1. Написать программу, которая определяет: имеет ли уравнение ax^2+bx+c=0 решение, где a, b, c – данные вещественные числа;

2. Дана точка на плоскости с координатами (х, у). Составить программу, которая выдает одно из сообщений "Да", "Нет", "На границе" в зависимости от того, лежит ли точка внутри заштрихованной области, вне заштрихованной области или на ее границе. Области задаются графически следующим образом: (см. рис. 1)

Изображение выглядит как диаграмма, линия, круг, шаблон

Автоматически созданное описание

Рис. 1 — Область, внутри которой должна лежать точка

# Выполнение работы

1. Написать программу, которая определяет: имеет ли уравнение ax^2+bx+c=0 решение, где a, b, c – данные вещественные числа. Ниже представлен листинг кода.

using System;

namespace \_01

{

internal class Program

{

static void SolutionToEqation()

{

Console.Write("ax^2 + bx + c = 0\n");

double a, b, c;

Console.Write("Enter a: ");

while (!double.TryParse(Console.ReadLine(), out a))

{

Console.Write("Invalid input. Please enter a valid number for a: ");

}

Console.Write("Enter b: ");

while (!double.TryParse(Console.ReadLine(), out b))

{

Console.Write("Invalid input. Please enter a valid number for b: ");

}

Console.Write("Enter c: ");

while (!double.TryParse(Console.ReadLine(), out c))

{

Console.Write("Invalid input. Please enter a valid number for c: ");

}

double x1, x2, D;

if (a == 0.0 && b == 0.0 && c == 0.0)

{

Console.Write("Infinite number of solutions");

}

else if (a == 0.0)

{

if (b == 0.0)

{

Console.Write("There are no roots");

}

else

{

if (c != 0.0)

{

x1 = (-c / b);

Console.Write("x = " + x1);

}

else Console.Write("0");

}

}

else

{

D = b \* b - 4 \* a \* c;

if (D > 0.0)

{

x1 = (-b + Math.Sqrt(D)) / (2 \* a);

x2 = (-b - Math.Sqrt(D)) / (2 \* a);

Console.Write("x1 = " + x1 + " x2 = " + x2);

}

else if (D == 0.0)

{

x1 = -b / (2 \* a);

Console.Write("x = " + x1);

}

else Console.Write("Complex");

}

}

public static void Main(string[] args)

{

SolutionToEqation();

while (true)

{

Console.Write("\nDo you want to complete: " +

"\n1) Yes" +

"\n2) No");

short complete;

while (!short.TryParse(Console.ReadLine(), out complete) || (complete != 1 && complete != 2))

{

Console.Write("Invalid input. Please enter 1 for Yes or 2 for No: ");

}

switch (complete) {

case 1:

SolutionToEqation();

break;

case 2:

System.Environment.Exit(0);

break;

}

}

}

}

}

SolutionToEqation(): Этот метод решает квадратное уравнение вида ax^2 + bx + c = 0. Он просит пользователя ввести коэффициенты a, b и c, а затем вычисляет корни уравнения. Если дискриминант D больше нуля, то уравнение имеет два решения. Если D равен нулю, то уравнение имеет одно решение. Если D меньше нуля, то у уравнения только комплексные корни, и выводится сообщение “Complex”.

Главный выполняемый код находится в функции Main(): Этот метод является точкой входа в приложение. Он вызывает метод SolutionToEqation(), а затем в бесконечном цикле спрашивает пользователя, хочет ли он продолжить решение уравнений. Если пользователь выбирает “1”, то метод SolutionToEqation() вызывается снова. Если пользователь выбирает “2”, то приложение завершает свою работу.

Метод TryParse() принимает два аргумента: строку, которую нужно преобразовать, и переменную, в которую будет записано преобразованное число. Если преобразование успешно, метод возвращает true и записывает преобразованное число в переменную. Если преобразование не удалось (например, если пользователь ввел текст, который не является числом), метод возвращает false.

В данном случае, while (!double.TryParse(Console.ReadLine(), out c)) будет продолжать запрашивать ввод от пользователя, пока он не введет число, которое можно преобразовать в double. Если введенная строка не может быть преобразована в число, код выводит сообщение об ошибке и снова запрашивает ввод. Это продолжается до тех пор, пока пользователь не введет корректное число. Таким образом, этот код помогает предотвратить ошибки ввода и обеспечивает, что переменная c будет содержать корректное числовое значение.

**2.** Дана точка на плоскости с координатами (х, у). Составить программу, которая выдает одно из сообщений "Да", "Нет", "На границе" в зависимости от того, лежит ли точка внутри заштрихованной области, вне заштрихованной области или на ее границе. Области задаются графически следующим образом: (см. рис. 1). Ниже представлен листинг кода.

using System;

namespace \_02

{

internal class Program

{

static void Graph()

{

Console.Write("Enter x: ");

double x;

while (!double.TryParse(Console.ReadLine(), out x))

{

Console.Write("Invalid input. Please enter a valid number for x: ");

}

Console.Write("Enter y: ");

double y;

while (!double.TryParse(Console.ReadLine(), out y))

{

Console.Write("Invalid input. Please enter a valid number for y: ");

}

if (x \* x + y \* y < 64.0 && x \* x + y \* y > 9.0 && x >= 0.0)

{

Console.Write("Yes");

}

else if ((x \* x + y \* y == 64.0 || x \* x + y \* y == 9.0) && x >= 0)

{

Console.Write("On border");

}

else

{

Console.Write("No");

}

}

public static void Main(string[] args)

{

Graph();

while (true)

{

Console.Write("\nDo you want to complete: " +

"\n1) Yes" +

"\n2) No");

short complete;

while (!short.TryParse(Console.ReadLine(), out complete) || (complete != 1 && complete != 2))

{

Console.Write("Invalid input. Please enter 1 for Yes or 2 for No: ");

}

switch (complete) {

case 1:

Graph();

break;

case 2:

System.Environment.Exit(0);

break;

}

}

}

}

}

Graph(): Этот метод определяет лежит ли точка внутри заштрихованной области. Он просит пользователя ввести координаты x и y, а затем проверяет, где лежит данная точка. Точка может оказаться внутри заштрихованной области, вне ее и на границе. В результате программа выдает информацию о том, где лежит точка.

Главный выполняемый код находится в функции Main(): Этот метод является точкой входа в приложение. Он вызывает метод Graph(), а затем в бесконечном цикле спрашивает пользователя, хочет ли он продолжить. Если пользователь выбирает “1”, то метод Graph() вызывается снова. Если пользователь выбирает “2”, то приложение завершает свою работу.

Метод TryParse() принимает два аргумента: строку, которую нужно преобразовать, и переменную, в которую будет записано преобразованное число. Если преобразование успешно, метод возвращает true и записывает преобразованное число в переменную. Если преобразование не удалось (например, если пользователь ввел текст, который не является числом), метод возвращает false.

В данном случае, while (!double.TryParse(Console.ReadLine(), out c)) будет продолжать запрашивать ввод от пользователя, пока он не введет число, которое можно преобразовать в double. Если введенная строка не может быть преобразована в число, код выводит сообщение об ошибке и снова запрашивает ввод. Это продолжается до тех пор, пока пользователь не введет корректное число. Таким образом, этот код помогает предотвратить ошибки ввода и обеспечивает, что переменная c будет содержать корректное числовое значение.

# Вывод

В ходе выполнения Лабораторной работы №2 были приобретены навыки создания методов и работы с операторами ветвления и циклами. Также был рассмотрен метод TryParse(), необходимый для предотвращения некорректного ввода пользователем.