

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА  
(САМАРСКИЙ УНИВЕРСИТЕТ)»**

Институт (факультет) информатики и кибернетики

1. Кафедра программных систем

**ОТЧЁТ**

**по лабораторной работе**

Лабораторная работа № 04

Основы языка С#: Простейшие классыпо дисциплине «ОСНОВЫ ПРОГРАММИРОВАНИЯ»

Выполнил Фокин Е.А., 6103

Проверил Котенева С.Э.

Самара  
2023

ЗАДАНИЕ

Задание 0.

Прочитать теоретический материал.

Задание 1. «Десятичный счетчик »

Описать класс, реализующий десятичный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне.

Счетчик должен иметь следующую структуру:

три поля: рабочий диапазон (мин. и макс. значения) и текущее значение;

метод увеличения значения счетчика;

метод уменьшения значения счетчика

метод получения значения состояния счетчика.

При попытке выхода за одну из границ, текущее значение принимает значение противоположной границы диапазона.

В классе Program в методе Main() реализовать функциональность описанного

класса (написать программу, проверяющую все разработанные элементы класса).

Задание 2. «Многочлен»

Описать класс – многочлен типа ax2 + bx + c:.

В классе должно быть присутствовать следующие члены:

* поля – переменные вещественных типов – коэффициенты многочлена;
* метод, возвращающий решения уравнения.

Всю функциональность описанного класса проверить в методе Main() класса Program.

Задание 3.

Подготовить отчет о работе.

КОД ПРОГРАММЫ

var main = (Dictionary<string, Action> menuActions) =>

{

Console.WriteLine("Лабораторная работа 4. Выполнил студент 6103 Фокин Евгений\n");

while (true)

{

Console.WriteLine("Выберите один из пунтов меню");

Console.WriteLine("0 - Осознанное завершение");

for (int index = 0; index < menuActions.Count; index++)

{

var item = menuActions.ElementAt(index);

Console.WriteLine($"{index + 1} - {item.Key}");

}

string menuAction = Console.ReadLine()!.Trim();

bool parsed = int.TryParse(menuAction, out int actionIndex);

if (!parsed || 0 > actionIndex || actionIndex > menuActions.Count)

Console.WriteLine("Не выбран ни один пункт меню\n");

else if (actionIndex == 0)

return;

else

{

var element = menuActions.ElementAt(actionIndex - 1);

Console.WriteLine(element.Key + "\n");

element.Value();

Console.WriteLine("Все проверки прошли успешно\n");

}

}

};

main(new Dictionary<string, Action>{

{"Десятичный счетчик", () => {

Console.WriteLine("Введите минимально значение");

int minValue = int.Parse(Console.ReadLine()!);

Console.WriteLine("Введите максимальное значение");

int maxValue = int.Parse(Console.ReadLine()!);

Console.WriteLine("Введите начальное значение");

int initialValue = int.Parse(Console.ReadLine()!);

var counter = new DecimalCounter(minValue, maxValue, initialValue);

while (true)

{

Console.WriteLine("Выберите действие:");

Console.WriteLine("0 - вернуться в меню");

Console.WriteLine("1 - увеличить значение");

Console.WriteLine("2 - уменьшить значение");

Console.WriteLine("3 - получить значение");

string selectedAction = Console.ReadLine()!.Trim();

switch (selectedAction)

{

case "0":

return;

case "1":

Console.WriteLine("Введите число, на которое нужно увеличить");

counter.Increase(int.Parse(Console.ReadLine()!));

Console.WriteLine(counter.GetValue());

break;

case "2":

Console.WriteLine("Введите число, на которое нужно уменьшить");

counter.Decrease(int.Parse(Console.ReadLine()!));

Console.WriteLine(counter.GetValue());

break;

case "3":

counter.GetValue();

break;

default:

break;

}

}

}},

{"Многочлен", () => {

while (true)

{

Console.WriteLine("Выберите действие:");

Console.WriteLine("0 - вернуться в меню");

Console.WriteLine("1 - ввести значения");

string selectedAction = Console.ReadLine()!.Trim();

switch (selectedAction)

{

case "0":

return;

case "1":

Console.WriteLine("Введите a");

int a = int.Parse(Console.ReadLine()!);

Console.WriteLine("Введите b");

int b = int.Parse(Console.ReadLine()!);

Console.WriteLine("Введите c");

int c = int.Parse(Console.ReadLine()!);

var polynomial = new Polynomial(a, b, c);

Console.WriteLine(polynomial.GetSolution() + "\n");

break;

default:

break;

}

}

}}

});

class DecimalCounter

{

private readonly int minValue;

private readonly int maxValue;

private int value;

public DecimalCounter(int minValue, int maxValue, int value)

{

this.minValue = minValue;

this.maxValue = maxValue;

if (value > maxValue)

this.value = maxValue;

else if (value < minValue)

this.value = minValue;

else

this.value = value;

}

public void Increase(int incValue)

{

int newValue = value + incValue;

if (newValue > maxValue)

value = (newValue - maxValue) + minValue;

else

value = newValue;

}

public void Decrease(int decValue)

{

int newValue = value - decValue;

if (newValue < minValue)

value = maxValue - (minValue - newValue);

else

value = newValue;

}

public int GetValue()

{

return value;

}

};

class Polynomial

{

private readonly double a;

private readonly double b;

private readonly double c;

public Polynomial(double a, double b, double c)

{

this.a = a;

this.b = b;

this.c = c;

}

public string GetSolution()

{

double discriminant = Math.Pow(b, 2) - 4 \* a \* c;

if (discriminant > 0)

{

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

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

return $"Уравнение имеет два корня: x1 = {x1}, x2 = {x2}";

}

else if (discriminant == 0)

{

double x = -b / (2 \* a);

return $"Уравнение имеет один корень: x = {x}";

}

return "Уравнение не имеет корней";

}

}

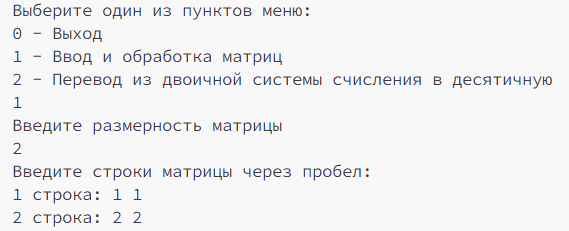


Рисунок 1 - Ввод матриц

ВЫВОДЫ

В лабораторной работе были использованы возможности языка C#:

* цикл while;
* цикл for;
* ветвления if;
* классы;
* конструкторы;
* поля;
* readonly поля;
* методы;
* лямбда-выражения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Полное руководство по языку программирования С# 11 и платформе .NET 7. – Режим доступа: URL: <https://metanit.com/sharp/tutorial/> (дата обращения: 17.10.23).