**Министерство образования и науки Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение

Высшего образования

**«Владимирский государственный университет**

**Имени Александра Григорьевича и Николая Григорьевича Столетовых»**

(ВлГУ)

Институт информационных технологий и электроники

Кафедра «Физика и прикладная математика»

**ЛАБОРАТОРНАЯ РАБОТА №1**

По дисциплине   
«Дискретная математика»

На тему:

**«Моделирование операции “пересечение” для двух числовых множеств»**

Выполнил:

ст. гр.ПМИ-123

Рушев А.М.

Принял:

к.т.н., доцент каф. ФиПМ

Горлов В.Н.

Владимир, 2025

**Цель работы:** изучить способы численного моделирования операции пересечения для множеств и составить компьютерную программу для выполнения пересечения двух конечных множеств.

**Задание:** составить компьютерную программу для выполнения пересечения двух конечных множеств.

**Теоретическая часть:**

Пересечение двух числовых множеств — это операция, которая находит общие элементы между этими множествами. Остановимся на основных моментах теории:

1. Определение множества: Множество — это коллекция различных объектов, рассматриваемых как единое целое. Числовые множества состоят из чисел и могут быть конечными (например, {1, 2, 3}) или бесконечными (например, натуральные числа).

2. Пересечение множеств: Пересечение двух множеств A и B обозначается как A ∩ B и определяет набор всех элементов, которые присутствуют как в A, так и в B. Формально:

- A ∩ B = {x | x ∈ A и x ∈ B}

3. Свойства пересечения:

- Коммутативность: A ∩ B = B ∩ A

- Ассоциативность: (A ∩ B) ∩ C = A ∩ (B ∩ C)

- Идентичность: A ∩ A = A

- Нейтральность: A ∩ ∅ = ∅ (пересечение с пустым множеством)

4. Примеры:

- Пусть A = {1, 2, 3, 4} и B = {3, 4, 5, 6}. Тогда A ∩ B = {3, 4}.

- Если A = {7, 8} и B = {1, 2}, то A ∩ B = ∅, так как нет общих элементов.

5. Применение в реальных задачах: Пересечение множеств часто используется в различных областях, таких как статистика, базы данных (для поиска общих записей) и анализ данных.

Эти основные аспекты дают общее представление о пересечении числовых множеств и его значении.

**Ход работы:**

Конструктор формы.

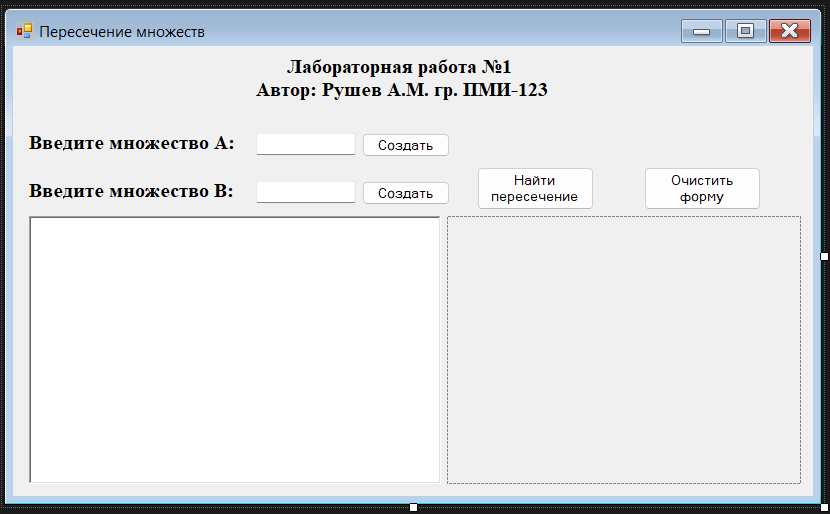
****

Рисунок 1 – Конструктор формы

Листинг приложения:

**MainForm.cs**

using System;

using System.Collections.Generic;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices;

using System.Windows.Forms;

namespace Lab1DM\_WinForms

{

public partial class MainForm : Form

{

private Random random = new Random();

private Int32[] massCircle = new Int32[0];

private Int32[] massA = new Int32[0];

private Int32[] massB = new Int32[0];

private readonly string imagePathTrue = Path.Combine(Directory.GetCurrentDirectory(), "imageTrue.png");

private readonly string imagePathFalse = Path.Combine(Directory.GetCurrentDirectory(), "imageFalse.png");

private readonly string imagePathNoDuplicated = Path.Combine(Directory.GetCurrentDirectory(), "imageNoDuplicated.png");

private readonly string filePath = Path.Combine(Directory.GetCurrentDirectory(), "fileMass.txt");

public MainForm()

{

InitializeComponent();

pictBxPrintCircle.SizeMode = PictureBoxSizeMode.StretchImage;

}

private void btnClearForm\_Click(object sender, EventArgs e)

{

tbxA.Clear();

tbxB.Clear();

rTbxPrintInfo.Clear();

pictBxPrintCircle.Image = null;

massA = null;

massB = null;

massCircle = null;

}

private async void btnSearch\_Click(object sender, EventArgs e)

{

try

{

if (massA == null || massA.Length == 0 || massB == null || massB.Length == 0)

{

MessageBox.Show("Один или несколько множеств пустые");

pictBxPrintCircle.Image = Image.FromFile(imagePathFalse);

return;

}

massCircle = massA.Concat(massB).ToArray();

Array.Sort(massCircle);

var duplicated = massCircle.GroupBy(x => x).Where(g => g.Count() > 1).Select(g => g.Key).ToList();

using (StreamWriter sw = new StreamWriter(filePath))

{

sw.Write("Множество A\n");

int countA = 0;

foreach (var item in massA)

{

sw.Write($"{item}\t");

countA++;

if (countA % 10 == 0)

{

sw.WriteLine();

}

}

sw.Write("Множество B\n");

int countB = 0;

foreach (var item in massB)

{

sw.Write($"{item}\t");

countB++;

if (countB % 10 == 0)

{

sw.WriteLine();

}

}

sw.Write("Пересечения множеств\n");

int countDuplicated = 0;

foreach (var item in duplicated)

{

sw.Write($"{item}\t");

countDuplicated++;

if (countDuplicated % 10 == 0)

{

sw.WriteLine();

}

}

}

rTbxPrintInfo.AppendText($"\nПересечение:\n");

foreach (var item in duplicated)

{

rTbxPrintInfo.AppendText($"{item.ToString()}\t");

}

if (duplicated.Any())

{

pictBxPrintCircle.Image = Image.FromFile(imagePathTrue);

}

else

{

MessageBox.Show($"Пересечений нет");

rTbxPrintInfo.AppendText($"Данных нет\n");

pictBxPrintCircle.Image = Image.FromFile(imagePathNoDuplicated);

}

}

catch (Exception ex)

{

MessageBox.Show($"{ex.Message}");

}

}

private void btnCreateA\_Click(object sender, EventArgs e)

{

try

{

if (!Int32.TryParse(tbxA.Text, out int a))

{

tbxA.Clear();

MessageBox.Show($"Число должно быть целым");

return;

}

HashSet<int> list = new HashSet<int>();

while (list.Count < a)

{

list.Add(random.Next(1, 101));

}

massA = list.ToArray();

Array.Sort(massA);

rTbxPrintInfo.AppendText($"\nМножество A:\n");

foreach (var item in massA)

{

rTbxPrintInfo.AppendText($"{item.ToString()}\t");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void btnCreateB\_Click(object sender, EventArgs e)

{

try

{

if (!Int32.TryParse(tbxB.Text, out int b))

{

tbxA.Clear();

MessageBox.Show($"Число должно быть целым");

return;

}

HashSet<int> list = new HashSet<int>();

while (list.Count < b)

{

list.Add(random.Next(1, 101));

}

massB = list.ToArray();

Array.Sort(massB);

rTbxPrintInfo.AppendText($"\nМножество B:\n");

foreach (var item in massB)

{

rTbxPrintInfo.AppendText($"{item.ToString()}\t");

}

}

catch (Exception ex)

{

MessageBox.Show($"{ex.Message}");

}

}

}

}

Результат выполнения приложения:

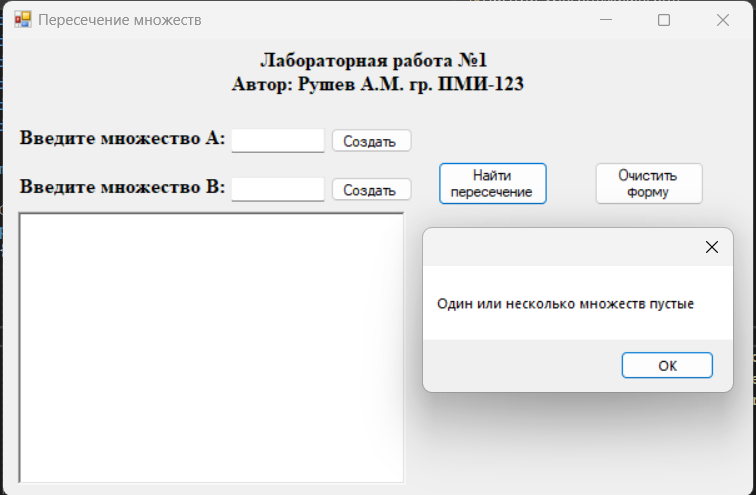


Рисунок 2 – Ошибка при пустом множестве

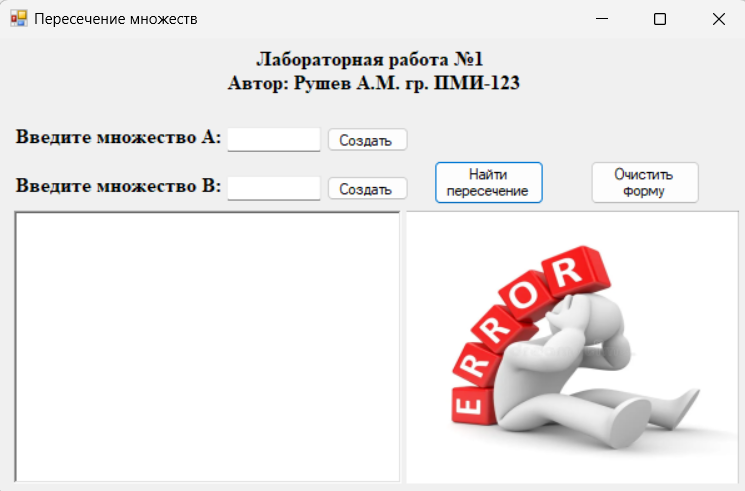


Рисунок 3 – Результат ошибки

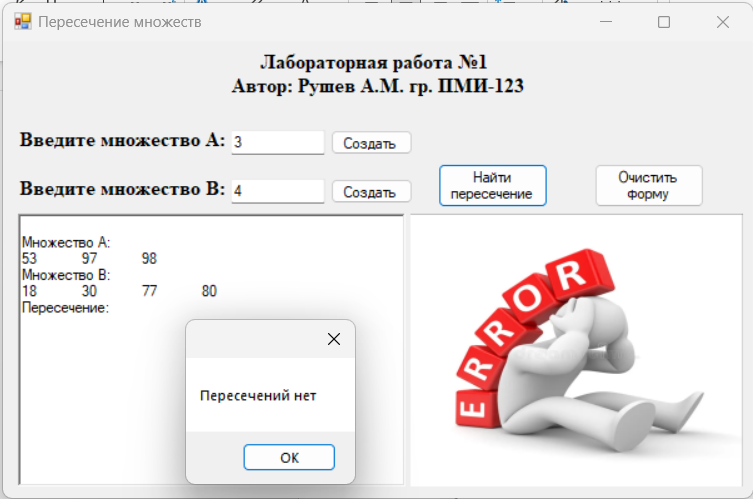


Рисунок 4 – Ошибка пересечений

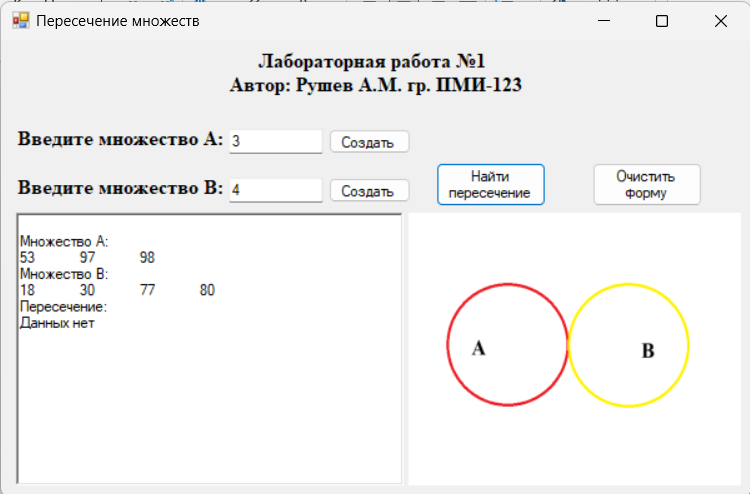


Рисунок 5 – Пересечений нет

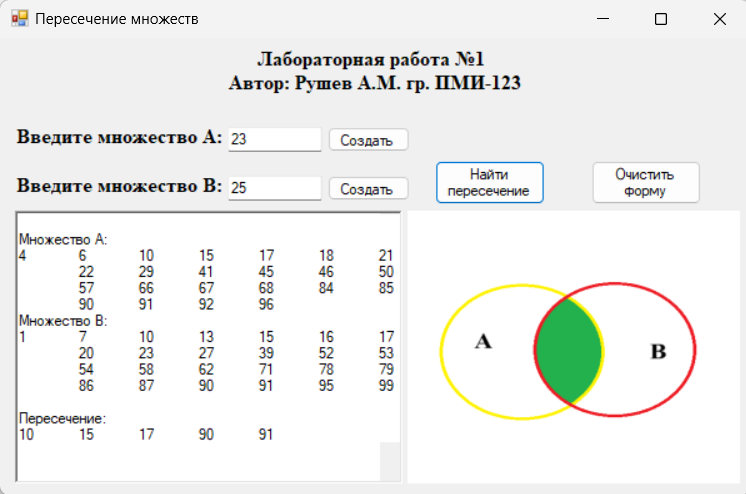


Рисунок 6 – Результат пересечений

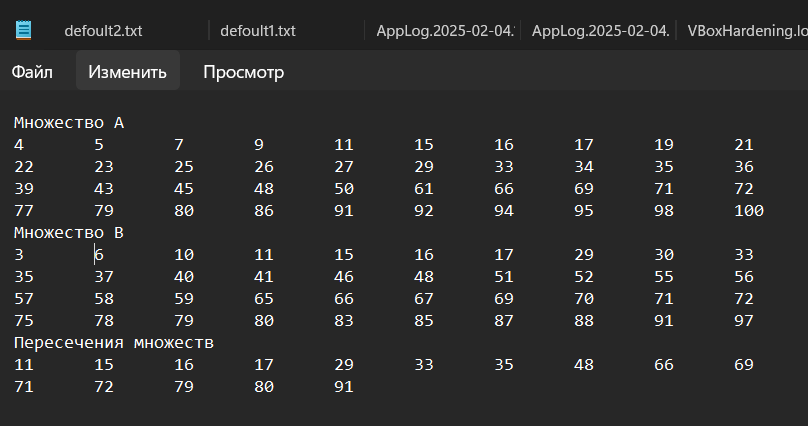


Рисунок 7 – Вывод данных в файл

**Вывод:** в ходе лабораторной работы были изучены способы численного моделирования операции пересечения для двух множеств и реализовано оконное приложение на языке программирования C#.