

Министерство науки и высшего образования Российской Федерации  
Муромский институт (филиал)  
Федерального государственного бюджетного образовательного учреждения  
высшего образования  
«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет \_\_\_\_\_ ИТР \_\_\_\_\_

Кафедра \_\_\_\_\_ ПИН \_\_\_\_\_

# ЛАБОРАТОРНАЯ РАБОТА №1

По Дискретной математике

---

Руководитель

Кульков Я.Ю.

\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Студент \_\_\_\_\_ ПИН - 121 \_\_\_\_\_

(группа)

Ермилов М.В.

\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Муром 2022

## Лабораторная работа №1

### Тема: МОДЕЛИРОВАНИЕ ОПЕРАЦИЙ НАД МНОЖЕСТВАМИ.

**Цель работы:** Изучить способы численного моделирования операций пересечения, дополнения и разности для множеств.

Задание на лабораторную работу:

Подготовить класс, реализующий пересечение, объединение и разности двух

конечных множеств. С использованием данного класса реализовать оконное приложение

для работы с двумя множествами.

Интерфейсная часть должна включать:

- полей ввода элементов для двух множеств;
- поля вывода результатов;
- управляющих кнопок для выполнения операций над множествами.  $A \cap B = \{x \mid x \in A \text{ и } x \in B\}$   $A \cup B = \{x \mid x \in A \text{ или } x \in B\}$   $A \setminus B = \{x \mid x \in A \text{ и } x \notin B\}$

Множества должны быть представлены наборами однотипных данных. В текущей

работе должны рассматриваться наборы целых чисел, а также произвольные наборы.

Для хранения множества целых чисел будем использовать массивы `int`, на произвольных наборов – массивы `String`.

Так как множества должны состоять из неповторяющихся элементов, то в классе

должны быть реализованы методы для проверки на уникальность элементов:

`bool isSet(int[] A)`

`bool isSet(String[] A)`

					МИ ВлГУ 09.03.04		
Изм.	Лист	№ докум.	Подпись	Дата	МОДЕЛИРОВАНИЕ ОПЕРАЦИЙ НАД МНОЖЕСТВАМИ.		
Разраб.		Ермилов М.В.					
Провер.		Кульков Я.Ю.					
Реценз.							
Н. Контр.							
Утверд.					ПИН-121		
					Лит.	Лист	Листов
						2	7

Реализовать метод для исключения повторяющихся элементов:

int[] Remove(int[] A)

String[] Remove(String[] A)

Метод выполнения объединения двух множеств:

int[] Union(int[] A, int[] B)

String[] Union(String[] A, String[] B)

Метод выполнения пересечения двух множеств:

int[] Intersection(int[] A, int[] B)

String[] Intersection(String[] A, String[] B)

Метод выполнения разности двух множеств:

int[] Difference(int[] A, int[] B)

String[] Difference(String[] A, String[] B)

Код по заданию:

Класс Operation:

```
class Operation
{
    static public bool IsSet(string[] array)
    {
        for (int i = 0; i < array.Length - 1; i++)
            for (int j = i + 1; j < array.Length; j++)
                if (array[i].Equals(array[j]))
                    return false;
        return true;
    }
    static public string[] Remove(string[] array)
    {
        string[] a = new string[array.Length];
        int k = 0;
        /* изменён
        for (int i = array.Length - 1; i >= 0; i--)
        {
            bool b = true;
            for (int j = i - 1; j >= 0; j--)
                if (array[i].Equals(array[j]))
                {
                    b = false;
                    break;
                }
            if (b)
            {
                a[k] = array[i];
                k++;
            }
        }
    }
}
```

					МИ ВлГУ 09.03.04	Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

```

    }
}
*/
for(int i = 0; i < array.Length; i++)
{
    bool b = true;
    for (int j = 0; j < k; j++)
        if(array[i].Equals(a[j]))
        {
            b = false;
            break;
        }
    if (b)
    {
        a[k] = array[i];
        k++;
    }
}

return arrays(a, k);
}
static public string[] Union(string[] A, string[] B)
{
    /* Лишние операции и циклы
    if (!isSet(A))
        A = Remove(A);
    if (!isSet(B))
        B = Remove(B);
    */
    string[] C = new string[A.Length + B.Length];
    //можно попробовать асинхронно заполнять массив C
    for (int i = 0; i < A.Length; i++)
        C[i] = A[i];
    for (int i = 0; i < B.Length; i++)
        C[i + A.Length] = B[i];
    return Remove(C);
}
static public string[] Intersection(string[] A, string[] B)
{
    /* Лишние операции и циклы
    if (!isSet(A))
        A = Remove(A);
    if (!isSet(B))
        B = Remove(B);
    */
    string[] C = new string[A.Length < B.Length ? A.Length : B.Length];
    int k = 0;
    for(int i = 0; i < A.Length; i++)
    {
        bool b = false;
        for(int j = 0; j < B.Length; j++)
            if (A[i].Equals(B[j]))
            {
                b = true;
                break;
            }
        if (b)
        {
            C[k] = A[i];
            k++;
        }
    }

    return Remove(arrays(C, k));
}

```

```

static public string[] Difference(string[] A, string[] B)
{
    /* Лишние операции и циклы
    if (!isSet(A))
        A = Remove(A);
    if (!isSet(B))
        B = Remove(B);
    */
    string[] C = new string[A.Length];
    int k = 0;
    for(int i = 0; i < A.Length; i++)
    {
        bool b = true;
        for(int j = 0; j < B.Length; j++)
            if (A[i].Equals(B[j]))
            {
                b = false;
                break;
            }
        if (b)
        {
            C[k] = A[i];
            k++;
        }
    }

    return Remove(arrays(C, k));
}

static private string[] arrays(string[] a, int k)
{
    string[] r = new string[k];
    for (int i = 0; i < k; i++)
        r[i] = a[i];
    return r;
}
}

```

Класс FormMain:

```

public partial class FormMain : Form
{
    public FormMain()
    {
        InitializeComponent();
    }
    string[] A { get => textBoxA.Text.Split(' '); }
    string[] B { get => textBoxB.Text.Split(' '); }
    private void RemoveA(object sender, EventArgs e) => Result(Operation.Remove(A));
    private void RemoveB(object sender, EventArgs e) => Result(Operation.Remove(B));
    private void UnionAB(object sender, EventArgs e) => Result(Operation.Union(A,
B));
    private void IntersectionAB(object sender, EventArgs e) =>
Result(Operation.Intersection(A, B));
    private void DifferenceAB(object sender, EventArgs e) =>
Result(Operation.Difference(A, B));
    private void DifferenceBA(object sender, EventArgs e) =>
Result(Operation.Difference(B, A));
    private void Result(string[] a)
    {
        string A = "";
        for (int i = 0; i < a.Length; i++)
        {

```

					МИ ВлГУ 09.03.04	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

```

        A += a[i];
        if (i < a.Length - 1)
            A += " ";
    }
    textBoxResult.Text = A;
}
}

```

The screenshot shows a Windows form titled 'Form1'. It contains two text boxes for input: 'Массив A' with the value '1 2 3 4 4 4 3 3 2 1 1 1' and 'Массив B' with the value '2 3 4 2 6 7 6 7 7 1 1 1'. Below these are six buttons: 'Исключить повторения A' (highlighted with a blue border), 'Исключить повторения B', 'Объединение A и B', 'Пересечение A и B', 'Разность A от B', and 'Разность B от A'. At the bottom, a 'Результат' text box displays the value '1 2 3 4'.

Рис 1 - пример работы программы

The screenshot shows the same 'Form1' window. The input arrays are identical to the first screenshot. In this example, the 'Исключить повторения B' button is highlighted with a blue border. The 'Результат' text box now displays the value '2 3 4 6 7 1'.

Рис 2 - пример работы программы

The screenshot shows the same 'Form1' window. The input arrays are identical to the previous examples. In this example, the 'Объединение A и B' button is highlighted with a blue border. The 'Результат' text box now displays the value '1 2 3 4 6 7'.

Рис 3 - пример работы программы

Рис 4 - пример работы программы

Рис 5 - пример работы программы

Рис 6 - пример работы программы