

Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

Лабораторная работа №3  
По дисциплине: «СПП»

Выполнил:  
Студент 3 курса  
Группы ПО-3  
Новикевич А.А.  
Проверил:  
Монтик Н.С.

## Цель работы: научиться создавать и использовать классы в программах на языке C#

### Вариант 5

#### Задания:

1)

- 5) **Множество целых чисел ограниченной мощности** – Предусмотреть возможность объединения двух множеств, вывода на печать элементов множества, а так же метод, определяющий, принадлежит ли указанное значение множеству. Класс должен содержать методы, позволяющие добавлять и удалять элемент в/из множества. Конструктор должен позволить создавать объекты с начальной инициализацией. Мощность множества задается при создании объекта. **Реализацию множества осуществить на базе одномерного массива.** Реализовать метод equals, выполняющий сравнение объектов данного типа.

2)

5) **Моделирование файловой системы**

Составить программу, которая моделирует заполнение гибкого диска (1440 Кб). В процессе работы файлы могут записываться на диск и удаляться с него.

С каждым файлом (File) ассоциированы следующие данные:

- Размер
- Распирение
- Имя файла
- Как файлы могут трактоваться и директории, которые в свою очередь содержат другие файлы и папки.

Если при удалении образовался свободный участок, то вновь записываемый файл помещается на этом свободном участке, либо, если он не помещается на этом участке, то его следует разместить после последнего записанного файла. Если файл превосходит длину самого большого участка, выдается аварийное сообщение. Рекомендуется создать список свободных участков и список занятых участков памяти на диске.

Код программы:

#### Задание 1:

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            Mnoj first = new Mnoj();
            Mnoj second = new Mnoj();

            Console.WriteLine(first.Equals(second));
            first.Equals(second);
            first.Add(2);
            first.Connect(second);
            first.Search(1);
            first.Show();
        }
    }
}
```

```

        Console.ReadKey();
    }
}
public class Mnoj
{
    public int[] mnojestvo;

    public Mnoj()
    {
        mnojestvo = new int[0];
        Console.WriteLine("Enter number/ q to stop: ");
        string temp = "";
        int Temp, size = 0;
        while (true)
        {
            temp = Console.ReadLine();
            bool check = int.TryParse(temp, out Temp);
            if (check)
            {
                Array.Resize(ref mnojestvo, ++size);
                mnojestvo[size-1] = Temp;
            }
            if (temp == "q") break;
        }
    }

    public override bool Equals(Object obj)
    {
        if (obj == null)
            return false;
        Mnoj temp = obj as Mnoj; //проверка на принадлежность классу Equila
        if (mnojestvo.Length != temp.mnojestvo.Length)
            return false;
        for (int i = 0; i < mnojestvo.Length; i++)
        {
            if (mnojestvo[i] != temp.mnojestvo[i])
                return false;
        }
        return true;
    }

    public void Show()
    {
        for (int i = 0; i < mnojestvo.Length; i++)
        {
            Console.WriteLine($"{mnojestvo[i]}");
        }
    }

    public void Add(int num)
    {
        for (int i = 0; i < mnojestvo.Length; i++)
        {
            if(mnojestvo[i] == num)
            {
                Console.WriteLine("Данное число есть в множестве");
                return;
            }
        }
    }
}

```

```

        Array.Resize(ref mnojestvo, mnojestvo.Length + 1);
        mnojestvo[mnojestvo.Length - 1] = num;
    }

    public void Delete(int number)
    {
        List<int> list = mnojestvo.ToList<int>();
        list.RemoveAt(number - 1);
        mnojestvo = list.ToArray();
    }

    public void Connect(Mnoj mnoj)
    {
        bool check;
        int[] arr = mnojestvo;
        for (int i = 0; i < mnoj.mnojestvo.Length; i++)
        {
            check = true;
            for (int j = 0; j < arr.Length; j++)
            {
                if (mnoj.mnojestvo[i] == arr[j])
                    check = false;
            }
            if (check)
            {
                Array.Resize(ref arr, arr.Length + 1);
                arr[arr.Length - 1] = mnoj.mnojestvo[i];
            }
        }
        Array.Sort(arr);
        mnojestvo = arr;
    }

    public void Search(int number)
    {
        bool check = false;
        for (int i = 0; i < mnojestvo.Length; i++)
        {
            if (mnojestvo[i] == number)
            {
                check = true; break;
            }
        }
        if (check) Console.WriteLine("СИМВОЛ ЕСТЬ!!!!");
        else Console.WriteLine("СИМВОЛА НЕТ(");
    }
}
}

```

```

Enter number/ q to stop:
1
2
3
4
5
q
Enter number/ q to stop:
4
5
6
7
8
q
False
Данное число есть в множестве
СИМВОЛ ЕСТЬ!!!!
1
2
3
4
5
6
7
8

```

## Задание 2:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Reflection.Metadata.Ecma335;
using System.Runtime.InteropServices.ComTypes;

namespace secondPart
{
    class Program
    {
        static void Main(string[] args)
        {
            Disk disk = new Disk();
            disk.Add(new File(100000, "asd", "txt"));
            disk.Add(new File(100000, "zxc", "txt"));
            disk.Add(new File(500000, "xcv", "txt"));
            disk.Add(new File(200000, "cvb", "txt"));
            disk.Show();
            disk.Delete("asd");
            disk.Delete("xcv");
            disk.Add(new File(400000, "xvcb", "txt"));
            disk.Show();
        }
    }
    public class Disk
    {
        List<File> Files = new List<File>();
        const int Size = 1440000;

        public Disk() { }
        public void Delete(string _name)

```

```

{
    for(int i = 0; i < Files.Count; i++)
    {
        if (Files[i].name == _name) Files.RemoveAt(i);
    }
}
public void Show()
{
    foreach(File file in Files)
    {
        Console.WriteLine($" Имя Файла {file.name}| РАЗмер {file.size}|
Расширение {file.exp}| НАчало и Конец {file.Psize} {file.Nsize}");
    }
}
public void Add(File file)
{
    int size = file.size;
    int temp = 0;
    if(Files.Count == 0)
    {
        file.Psize = 0;
        file.Nsize = file.Psize + file.size;
        Files.Add(file);
        Console.WriteLine("Файл успешно создан");
        return;
    }
    temp = Files[0].Psize;
    for (int i = 0; i < Files.Count; i++)
    {
        if (i == Files.Count - 1)
        {
            if (Files[i].Psize - temp > size)
            {
                file.Psize = temp;
                file.Nsize = temp + file.size;

                Files.Insert(i, file);

                Console.WriteLine("Файл успешно создан");
                return;
            }
            int TempSize = Size - Files[i].Nsize;
            if (file.size < TempSize)
            {
                file.Psize = Files[i].Nsize + 1;
                file.Nsize = file.Psize + file.size;
                Files.Add(file);
                Console.WriteLine("Файл успешно создан");

            }
            else Console.WriteLine("Нет места для файла");
            return;
        }
        if (Files[i].Psize == temp)
        {
            temp = Files[i].Nsize + 1;
            continue;
        }
        else
        {
            if (Files[i].Psize - temp > size)
            {
                //file.Psize = Files[i].Nsize;
                file.Psize = temp;
            }
        }
    }
}

```

