Министерство образования Республики Беларусь

Учреждение образования «Брестский государственный технический университет»

Кафедра ИИТ

Отчёт по лабораторной работе №3

Выполнил:

студент 3 курса

Группы ПО-5

Поздняков Д.А.

Проверил:

Крощенко А.А.

Брест 2021

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

**Вариант 5**

**Задание 1.** 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.

**Задание 2.** 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

• Расширение

• 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):**

Lab3.cs

using System;

namespace lab3.\_1.\_5

{

class lab3

{

static void Main(string[] args)

{

int[] arr1 = new int[] { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };

int[] arr2 = new int[] { 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 };

SetOfIntegers Set1 = new SetOfIntegers(arr1);

SetOfIntegers Set2 = new SetOfIntegers(arr2);

SetOfIntegers Set3 = new SetOfIntegers();

Set3.UnionOfSets(Set1, Set2);

Set3.Print();

Set3.BelongsToTheSet(10);

Set3.BelongsToTheSet(21);

Set3.BelongsToTheSet(111111111);

Set3.BelongsToTheSet(7);

Set2.DeleteElementOfSet(52);

Set2.Print();

Set1.AddElementOfSet(17);

Set1.Print();

Console.WriteLine(Set1.Equals(Set2));

Console.WriteLine(Set3.ToString());

}

}

}

SetOfIntegers.cs

using System;

namespace lab3.\_1.\_5

{

class SetOfIntegers

{

private static int N = 10;

int[] Set = new int[N];

public int[] GetSet()

{

return Set;

}

public void SetSet(int[] set)

{

Set = set;

}

public override bool Equals(object obj)

{

if (obj == null)

return false;

SetOfIntegers set = obj as SetOfIntegers; // возвращает null если объект нельзя привести к типу Set

if ((set as SetOfIntegers) == null)

return false;

return set.Set == Set;

}

public override string ToString()

{

string str = "";

if (Set == null)

return base.ToString();

else

{

for (int i = 0; i < Set.Length; i++)

{

str += Set[i] + " ";

}

}

return str;

}

public SetOfIntegers()

{

}

public SetOfIntegers(int[] arr)

{

if (arr.Length != 10)

{

Console.WriteLine("Неверный размер массива!");

}

for (int i = 0; i < N; i++)

{

Set[i] = arr[i];

}

}

public void Print()

{

for (int i = 0; i < this.Set.Length; i++)

{

Console.Write($"{this.Set[i]} ");

}

Console.WriteLine();

}

public int[] UnionOfSets(SetOfIntegers set1, SetOfIntegers set2)

{

int[] Res = new int[set1.GetSet().Length + set2.GetSet().Length];

for (int i = 0; i < set1.GetSet().Length; i++)

{

Res[i] = set1.GetSet()[i];

}

for (int i = set1.GetSet().Length, j = 0; i < Res.Length; i++, j++)

{

Res[i] = set2.GetSet()[j];

}

SetSet(Res);

return Res;

}

public void BelongsToTheSet(int element)

{

for (int i = 0; i < GetSet().Length; i++)

{

if (GetSet()[i] == element)

{

Console.WriteLine($"{element} принадлежит множеству");

return;

}

}

Console.WriteLine($"{element} не принадлежит множеству");

}

public void DeleteElementOfSet(int NumElement)

{

NumElement--;

if (NumElement > GetSet().Length || NumElement < 1)

{

Console.WriteLine("Ошибка! Введено слишком большое или слишком маленькое значение!");

return;

}

int[] Res = new int[GetSet().Length - 1];

for (int i = 0, j = 0; i < GetSet().Length; i++)

{

if (i != NumElement)

{

Res[j] = GetSet()[i];

j++;

}

}

SetSet(Res);

}

public void AddElementOfSet(int NumElement)

{

int[] Res = new int[GetSet().Length + 1];

for (int i = 0; i < GetSet().Length; i++)

{

Res[i] = GetSet()[i];

}

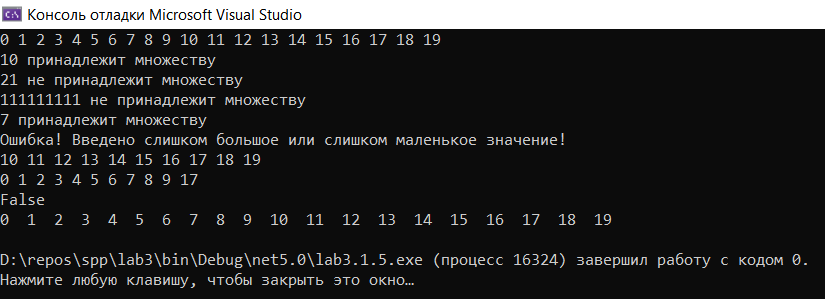
Res[Res.Length - 1] = NumElement;

SetSet(Res);

}

}

}



**Код программы (задание 2):**

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

namespace lab3.\_2.\_5

{

class Program

{

static void GenerateFileSystem(List<File> file, int[] freeSpace, int numFiles)

{

for (int i = 0; i < numFiles; i++)

{

File file1 = new File(i + 1, freeSpace);

if (!file1.Empty)

{

file.Add(file1);

}

}

}

static void DeleteFile(List<File> file, int[] freeSpace, string fileName)

{

for(int i = 0; i < file.Count; i++)

{

if(file[i].FN == fileName)

{

int temp1 = Array.IndexOf(freeSpace, file[i].ID);

int temp2 = Array.IndexOf(freeSpace, file[i].ID) + file[i].FS;

file.RemoveAt(i);

for(int j = temp1; j < temp2; j++)

{

freeSpace[j] = 0;

}

i--;

Console.WriteLine("Файл удален!");

//break; //Если нужно удалять только один файл

}

}

}

static void AddFile(List<File> file, int[] freeSpace)

{

int[] tempArr = freeSpace;

IEnumerable<int> nums = tempArr.Distinct();

tempArr = nums.ToArray();

int j = 0;

for(; j < tempArr.Length + 1; j++)

{

if (!tempArr.Contains(j))

{

break;

}

}

File file1 = new File(j, freeSpace);

if (!file1.Empty)

{

file.Add(file1);

}

}

static void Main(string[] args)

{

int[] FreeSpace = new int[1440];

int NumFiles = 10;

List<File> FileSystem = new List<File>();

GenerateFileSystem(FileSystem, FreeSpace, NumFiles);

for(int i = 0; i < FileSystem.Count; i++)

{

FileSystem[i].Print();

}

DeleteFile(FileSystem, FreeSpace, "da");

AddFile(FileSystem, FreeSpace);

for (int i = 0; i < FileSystem.Count; i++)

{

FileSystem[i].Print();

}

for (int i = 0; i < FreeSpace.Length; i++)

{

Console.WriteLine($"{i + 1}) {FreeSpace[i]}");

}

}

}

}

File.cs

using System;

namespace lab3.\_2.\_5

{

class File

{

public static int FileSystemSize = 1440;

private int FileSize;

private string FileExtension;

private string FileName;

public int ID;

public int NumOfFiles;

public bool Empty;

public int FS

{

get

{

return FileSize;

}

set

{

FileSize = value;

}

}

public string FE

{

get

{

return FileExtension;

}

set

{

FileExtension = value;

}

}

public string FN

{

get

{

return FileName;

}

set

{

FileName = value;

}

}

static string randomString(int lenght)

{

Random rand = new Random();

Console.OutputEncoding = System.Text.Encoding.UTF8;

string a = "";

for (int i = 0; i < lenght; i++)

{

a += (char)rand.Next(0x0061, 0x007A);

}

return a;

}

public void Print()

{

Console.WriteLine("Название файла: {0}\t Расширение: {1}\t Размер: {2}", FN, FE, FS);

}

public void CheckFreeSpace()

{

for(int i = 0; i < FileSystemSize; i++)

{

}

}

public void FillingFreeSpace(int[] FreeSpace)

{

if (ID == 1)

{

for (int i = 0; i < FS; i++)

{

FreeSpace[i] = ID;

}

}

else

{

int index = Array.IndexOf(FreeSpace, 0);

int total = 0;

for (int i = index; i < FreeSpace.Length - 1; i++)

{

if (FreeSpace[i] == 0)

{

total++;

if (total == FS)

{

for (int j = index; j < FS + index; j++)

{

FreeSpace[j] = ID;

}

break;

}

}

else

{

total = 0;

index = Array.IndexOf(FreeSpace, 0, i);

}

}

if (total < FS)

{

Console.WriteLine("Не хватает места для файла!");

Empty = true;

}

else Empty = false;

}

}

public File(int id, int[] FreeSpace)

{

Random temp = new Random();

string[] fe = new string[] { ".txt", ".rar", ".bat", ".dox", ".exe" };

string[] fn = new string[] { "tre", "da", "net", "ladno" };

ID = id;

FS = temp.Next(150, 250);

FE = fe[temp.Next(0, 5)];

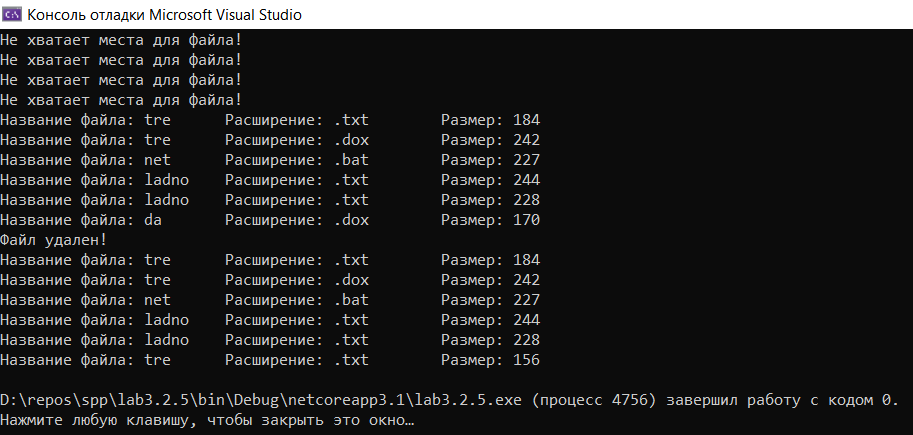
FN = fn[temp.Next(0, 4)];//randomString(temp.Next(3, 8));

FillingFreeSpace(FreeSpace);

}

}

}



Вывод: в ходе выполнения лабораторной работы научиться создавать и использовать классы в программах на языке программирования C#.