**ПРИЛОЖЕНИЕ 7**

Практическая работа №8

Задание 2

2.1

using System;   
using System.Collections.Generic;   
  
namespace Abituruen   
{   
class Program   
{   
  
static void Main(string[] args)   
{   
Abituruen[] Abiturueni = new Abituruen[]   
{   
new Abituruen("Сергеев ", "Минск ", 4, 5, 5),   
new Abituruen("Белов ", "Москва ", 3, 5, 5),   
new Abituruen("Аваков ", "Мариуполь ", 5, 5,5),   
new Abituruen("Козлов ", "Минск ", 5, 4, 5),   
new Abituruen("Канатов ", "Минск ", 3, 3, 4)   
};   
List<string> st = new List<string>();   
int count = 0;   
  
for (int i = 0; i < Abiturueni.Length; i++)   
if (Abiturueni[i].SrednBal >= 4.5 && Abiturueni[i].Adr == "Минск ")   
{ st.Add(Abiturueni[i].Fio); count++; }   
  
Console.WriteLine("Количество студентов проживающих в Минске, и прошедшие экзамены = {0}", count);   
Console.WriteLine("Отсортированный список студентов по алфавитному порядку:");   
  
st.Sort();   
foreach (string q in st)   
Console.WriteLine(q);   
Console.ReadKey();   
}   
}   
  
class Abituruen   
{   
string FIO;   
string Adress;   
int ocen\_mathem;   
int ocen\_fiz;   
int ocen\_yazik;   
  
public Abituruen(string FIO, string Adress, int ocen\_mathem, int ocen\_fiz, int ocen\_yazik)   
{   
this.FIO = FIO;   
this.Adress = Adress;   
this.ocen\_mathem = ocen\_mathem;   
this.ocen\_fiz = ocen\_fiz;   
this.ocen\_yazik = ocen\_yazik;   
Console.WriteLine("Фамилия: " + FIO);   
Console.WriteLine("Адрес: " + Adress);   
Console.WriteLine("Оценка по математике: " + ocen\_mathem);   
Console.WriteLine("Оценка по физике: " + ocen\_fiz);   
Console.WriteLine("Оценка по русскоум: " + ocen\_yazik);   
Console.WriteLine("--------------------------------------");   
}   
float Sredn\_bal()   
{   
return (float)(ocen\_fiz + ocen\_mathem + ocen\_yazik) / 3;   
}   
public string Adr   
{   
get { return Adress; }   
}   
public float SrednBal   
{   
get { return Sredn\_bal(); }   
}   
public string Fio   
{   
get { return FIO; }   
}   
}   
}

2.2

using System;

namespace ConsoleApplication1

{

class DataBase : IComparable<object>

{

public string FIO, Adres;

public int Nomer, Vrema\_vnytrennix\_razgovorov;

public int Vrema\_mezdynarodnyh\_razgovorov;

byte sort;

public DataBase() { }

public DataBase(string FIO, string Adres, int Nomer, int Vrema\_vnytrennix\_razgovorov1, int Vrema\_mezdynarodnyh\_razgovorov1)

{

this.FIO = FIO;

this.Adres = Adres;

this.Nomer = Nomer;

this.Vrema\_vnytrennix\_razgovorov = Vrema\_vnytrennix\_razgovorov1;

this.Vrema\_mezdynarodnyh\_razgovorov = Vrema\_mezdynarodnyh\_razgovorov1;

}

public int CompareTo(object other)

{

DataBase db = other as DataBase;

if (db != null)

{

switch (db.sort)

{

case 1:

{

if (String.Compare(this.Adres, db.Adres) > 0)

return 1;

if (String.Compare(this.Adres, db.Adres) < 0)

return -1;

else

return 0;

}

case 2:

{

if (String.Compare(this.FIO, db.FIO) > 0)

return 1;

if (String.Compare(this.FIO, db.FIO) < 0)

return -1;

else

return 0;

}

case 3:

{

if (String.Compare(Convert.ToString(this.Nomer), Convert.ToString(db.Nomer)) > 0)

return 1;

if (String.Compare(Convert.ToString(this.Nomer), Convert.ToString(db.Nomer)) < 0)

return -1;

else

return 0;

}

case 4:

{

if (String.Compare(Convert.ToString(this.Vrema\_vnytrennix\_razgovorov), Convert.ToString(db.Vrema\_vnytrennix\_razgovorov)) > 0)

return 1;

if (String.Compare(Convert.ToString(this.Vrema\_vnytrennix\_razgovorov), Convert.ToString(db.Vrema\_vnytrennix\_razgovorov)) > 0)

return -1;

else

return 0;

}

default: throw new FormatException("Такой сортировки не существует");

}

}

else

throw new ArgumentException("Данный параметр не относится к базе данных");

}

public void Sort(ref DataBase[] db\_object)

{

try

{

Array.Sort(db\_object);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

foreach (DataBase d in db\_object)

Console.WriteLine("{0}\t{1}\t{2} \t{3} \t{4}", d.FIO, d.Adres, d.Nomer, d.Vrema\_vnytrennix\_razgovorov, d.Vrema\_mezdynarodnyh\_razgovorov);

}

public void NumberSort(byte j, ref DataBase[] arr)

{

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

arr[i].sort = j;

}

}

class Program

{

static void Main()

{

DataBase[] db\_arr = new DataBase[5];

db\_arr[0] = new DataBase("Фалевич ", "д.жупраны ", 655666, 3, 6);

db\_arr[1] = new DataBase("Еваровский ", "ст.ошмяны ", 655666, 3, 8);

db\_arr[2] = new DataBase("Савицкий ", "ошмяны ", 655888, 1, 10);

db\_arr[3] = new DataBase("Богданович ", "ст.ошмяны ", 5353532, 5, 11);

db\_arr[4] = new DataBase("Шиманский ", "ст.ошмяны ", 8989565, 16, 19);

Console.WriteLine("Исходная таблица: \n-------------------------------------------------------------------------------------------------------\n\n" +

"ФИО \tАдрес \tНомер \t Время внутренних разговоров \t Время междугородних разговоров\n");

foreach (DataBase d in db\_arr)

Console.WriteLine("{0}\t{1}\t{2} \t{3} \t{4}", d.FIO, d.Adres, d.Nomer, d.Vrema\_vnytrennix\_razgovorov, d.Vrema\_mezdynarodnyh\_razgovorov);

Console.WriteLine("\nОтсортированная таблица БД по ФИО: " +

"\n-------------------------------------------------------------------------------------------------------\n" +

"ФИО \tАдрес \tНомер \t Время внутренних разговоров \t Время междугородних разговоров\n");

DataBase sr1 = new DataBase();

sr1.NumberSort(1, ref db\_arr);

sr1.Sort(ref db\_arr);

Console.WriteLine("\nОтсортированная таблица БД по Адрему : " +

"\n-------------------------------------------------------------------------------------------------------\n" +

"ФИО \tАдрес \tНомер \t Время внутренних разговоров \t Время междугородних разговоров\n");

sr1.NumberSort(2, ref db\_arr);

sr1.Sort(ref db\_arr);

Console.WriteLine("\nОтсортированная таблица БД по Номеру: " +

"\n-------------------------------------------------------------------------------------------------------\n" +

"ФИО \tАдрес \tНомер \t Время внутренних разговоров \t Время междугородних разговоров\n");

sr1.NumberSort(3, ref db\_arr);

sr1.Sort(ref db\_arr);

Console.WriteLine("\nОтсортированная таблица БД по времени внутригородских разговоров: " +

"\n-------------------------------------------------------------------------------------------------------\n" +

"ФИО \tАдрес \tНомер \t Время внутренних разговоров \t Время междугородних разговоров\n");

sr1.NumberSort(4, ref db\_arr);

sr1.Sort(ref db\_arr);

Console.ReadLine();

}

}

}