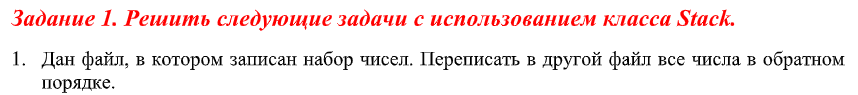
**//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////**



string path = @"D:\3 Курс\C#\Laba9\Laba19\Laba19\_1.txt";

string path2 = @"D:\3 Курс\C#\Laba9\Laba19\Laba19\_2.txt";

Stack stack = new Stack();

string str;

using (StreamReader reader = new StreamReader(path))

{

str = await reader.ReadToEndAsync();

Console.WriteLine(str);

}

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

stack.Push(str[i]);

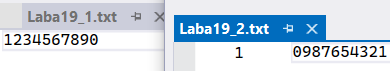
using (StreamWriter writer = new StreamWriter(path2, false))

{

for (int i = stack.Count; i > 0; i--)

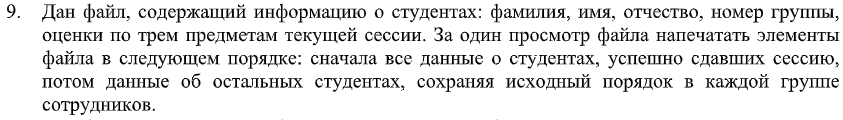
await writer.WriteAsync((char)stack.Pop());

}



**//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////**





string path = @"D\Laba19.xml";

List<Student> colection = new List<Student>();

Student predmet1 = new Student("Vova", "Lolkin", "Otecovich", "t-118", 1, 2, 3);

Student predmet2 = new Student("Roman", "Abobkin", "Otecovich", "t-118", 4, 2, 9);

Student predmet3 = new Student("Matvey", "Lolkin", "Otecovich", "t-118", 2, 2, 3);

Student predmet4 = new Student("Sasha", "Lolkin", "Otecovich", "t-118", 9, 9, 9);

Student predmet5 = new Student("Eugenya", "Lolkin", "Otecovich", "t-118", 6, 6, 7);

colection.Add(predmet1);

colection.Add(predmet2);

colection.Add(predmet3);

colection.Add(predmet4);

colection.Add(predmet5);

XmlSerializer xml = new XmlSerializer(typeof(List<Student>));

using (FileStream fs = new FileStream(path, FileMode.OpenOrCreate))

{

xml.Serialize(fs, colection);

WriteLine("Объект был сериализован\n");

}

List<Student>? colection2;

using (FileStream fs = new FileStream(path, FileMode.OpenOrCreate))

{

colection2 = xml.Deserialize(fs) as List<Student>;

}

Queue<Student>? students = new Queue<Student>();

foreach (var i in colection2)

{

if (i.ocenki[0] >= 3 && i.ocenki[1] >= 3 && i.ocenki[2] >= 3)

students.Enqueue(i);

}

foreach (var i in colection2)

{

if (i.ocenki[0] < 3 || i.ocenki[1] < 3 || i.ocenki[2] < 3)

students.Enqueue(i);

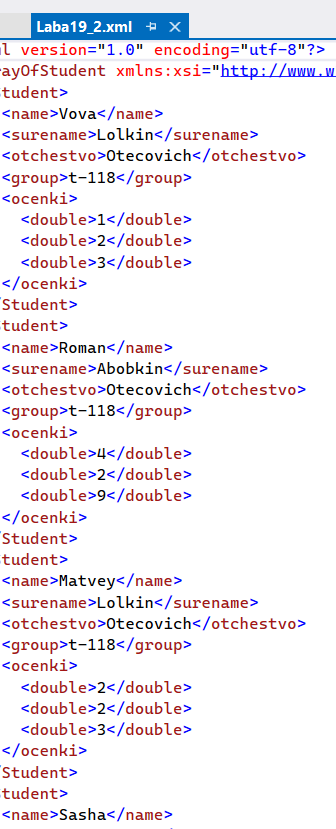
}

for (int i = 0, j = students.Count; i < j; i++)

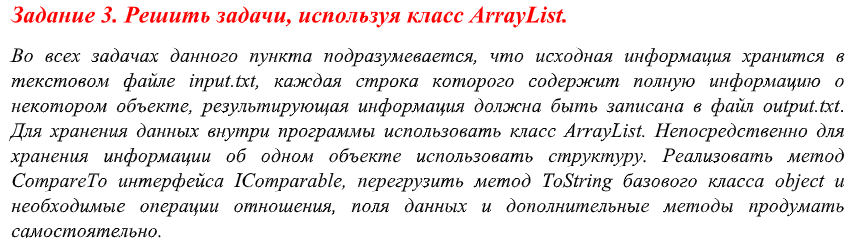
{

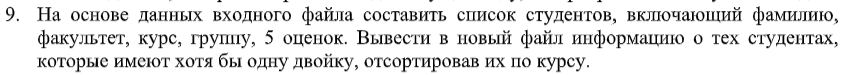
WriteLine(students.Dequeue());

}



**//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////**





string path = @"D:\3 Курс\C#\Laba9\Laba19\Laba19\_3.xml";

XmlSerializer xml = new XmlSerializer(typeof(ArrayList));

ArrayList students = new ArrayList

{

new Student("Vova", "Lolkin", "Otecovich", "t-118", 1, 2, 3, 4, 5),

new Student("Roman", "Abobkin", "Otecovich", "t-118", 4, 2, 9, 3, 3),

new Student("Matvey", "Lolkin", "Otecovich", "t-119", 2, 2, 3, 9, 9),

new Student("Sasha", "Lolkin", "Otecovich", "t-118", 9, 9, 9, 8, 7),

new Student("Eugenya", "Lolkin", "Otecovich", "t-118", 6, 6, 7, 9, 9)

};

ArrayList arrayList = new ArrayList();

foreach (Student student in students)

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

if (student.ocenki[i] == 2)

{

arrayList.Add(student);

break;

}

// ArrayList Невозможно сериализовать

/\*using (FileStream fs = new FileStream(path, FileMode.OpenOrCreate))

{

xml.Serialize(fs, arrayList);

WriteLine("Объект был сериализован\n");

}\*/

WriteLine("//////////////////////////////////////////////////////////////////");

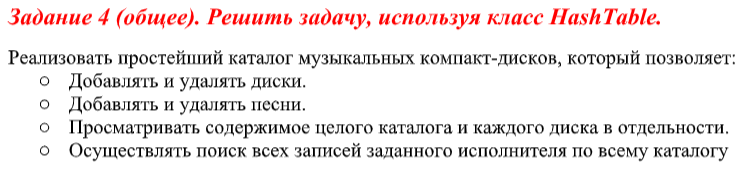
IComparer comparer = new Student();

students.Sort(comparer);

foreach (Student student in students)

Console.WriteLine(student);

**//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////**



bool menu = true;

ArrayList arrayList = new ArrayList();

while (menu)

{

Write("\n1 - Добавить диск\n2 - Удалить диск\n 3 - Добавить песню\n 4 - Удалить песню\n5 - Просмотреть содержимое\n6 - Поиск песни\n0 - Выход\n>>");

int sw = ToInt32(ReadLine());

WriteLine();

switch (sw)

{

case 0:

{

menu = false;

break;

}

case 1:

{

try

{

arrayList.Add(new Hashtable());

WriteLine($"Диск был успелшно добавлен!\nВсего дисков: {arrayList.Count}");

}

catch (Exception ex)

{

WriteLine(ex.Message);

}

break;

}

case 2:

{

try

{

WriteLine("Всего дисков: " + arrayList.Count);

Write("Какой диск вы желаете удалить?\n>>"); int n = ToInt32(ReadLine());

arrayList.RemoveAt(0);

WriteLine($"Диск был успелшно удалён!\nВсего дисков: {arrayList.Count}");

}

catch (Exception ex)

{

WriteLine(ex.Message);

}

break;

}

case 3:

{

try

{

Write("В какой диск вы желаете добавить песню?\n>>"); int n = ToInt32(ReadLine());

Write("Введите название песни:"); string name = ReadLine();

Write("Введите текст песни:"); string str = ReadLine();

((Hashtable)arrayList[n - 1]).Add(name, str);

}

catch (Exception ex)

{

WriteLine(ex.Message);

}

break;

}

case 4:

{

try

{

Write("Введите номер диска песни:"); int n = ToInt32(ReadLine());

Write("Введите название песни для удаления...\n>>"); string name = ReadLine();

((Hashtable)arrayList[n - 1]).Remove(name);

}

catch (Exception ex)

{

WriteLine(ex.Message);

}

break;

}

case 5:

{

try

{

Write("Какой диск вы желаете посмотреть?\nВведите \"-1\" чтобы просмотреть всё содержимое каталога\n>>"); int n = ToInt32(ReadLine());

if (n == -1)

{

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

{

WriteLine($"-------Диск {i + 1}-------");

ICollection с = (arrayList[i] as Hashtable).Keys;

foreach (string str in с)

WriteLine($"Песня \"{str}\"\nТекст песни:{(arrayList[i] as Hashtable)[str]}\n");

WriteLine();

}

}

else

{

WriteLine($"-------Диск {n}-------");

ICollection с = (arrayList[n - 1] as Hashtable).Keys;

foreach (string str in с)

WriteLine($"Песня \"{str}\"\nТекст песни:{(arrayList[n - 1] as Hashtable)[str]}\n");

WriteLine();

}

}

catch (Exception ex)

{

WriteLine(ex.Message);

}

break;

}

case 6:

{

try

{

Write("Введите название песни для для поиска...\n>>"); string name = ReadLine();

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

{

ICollection с = (arrayList[i] as Hashtable).Keys;

foreach (string str in с)

{

if(object.Equals(name, str))

{

WriteLine($"Песня находится в диске Номер: {i + 1}\nПесня \"{str}\"\nТекст песни:{(arrayList[i] as Hashtable)[str]}\n");

WriteLine();

}

}

}

}

catch (Exception ex)

{

WriteLine(ex.Message);

}

break;

}

}

WriteLine();

WriteLine();

}

