**Лабораторная работа № 54-55**

**«Работа с файловыми потоками»**

**Цель работы:** получение навыков работы с файловыми потоками в С#.

Задания для лабораторной работы **ОБЩИЕ**:

1. Создайте файл numbers.txt и запишите в него натуральные числа от 1 до 256 через запятую.  
 2. Дан массив строк: "Toyota", "Ford", "Chevrolet", "Honda", "Nissan", "BMW", "Mercedes-Benz", "Audi", "Volkswagen", "Hyundai", "Kia". Запишите в файл элементы массива построчно (каждый элемент в новой строке).

3. Возьмите любой текстовый файл, и найдите в нем размер самой длинной строки.

4.Дан массив размера N , заполненный случайными величинами (использовать new Random().Next(1,1000). Вывести в консоли вначале его элементы с четными индексами, а затем - с нечетными и записать результат в файл.

class Program

{

static void Main(string[] args)

{

using (StreamWriter writer = new StreamWriter("numbers.txt"))

{

for (int i = 1; i <= 256; i++)

{

writer.Write(i);

if (i < 256) writer.Write(",");

}

}

string[] carBrands = { "Toyota", "Ford", "Chevrolet", "Honda", "Nissan", "BMW", "Mercedes-Benz", "Audi", "Volkswagen", "Hyundai", "Kia" };

using (StreamWriter writer = new StreamWriter("car\_brands.txt"))

{

foreach (var brand in carBrands)

{

writer.WriteLine(brand);

}

}

string filename = "car\_brands.txt";

int maxLength = 0;

using (StreamReader reader = new StreamReader(filename))

{

string line;

while ((line = reader.ReadLine()) != null)

{

maxLength = Math.Max(maxLength, line.Length);

}

}

Console.WriteLine("Длина самой длинной строки: " + maxLength);

Random random = new Random();

int N = 20;

int[] numbers = new int[N];

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

{

numbers[i] = random.Next(1, 1000);

}

using (StreamWriter writer = new StreamWriter("output.txt"))

{

for (int i = 0; i < N; i += 2)

{

writer.Write(numbers[i] + (i + 2 < N ? ", " : ""));

}

writer.WriteLine();

for (int i = 1; i < N; i += 2)

{

writer.Write(numbers[i] + (i + 2 < N ? ", " : ""));

}

}

Console.WriteLine("Результаты записаны в файл output.txt");

}

}



Задания для лабораторной работы ИНДИВИДУАЛЬНЫЕ (**по вариантам**)

5.Вариант

* 1. читает бинарный файл изображения;
  2. применяет простой алгоритм шифрования (например, xor с ключом) к каждому байту файла;
  3. сохраняет зашифрованные данные в новый файл;
  4. реализуйте дешифрование (обратный процесс).

class Program

{

static void Main(string[] args)

{

string inputFilePath = "image.bmp";

string encryptedFilePath = "encrypted\_image.dat";

string decryptedFilePath = "decrypted\_image.bmp";

byte key = 123;

EncryptFile(inputFilePath, encryptedFilePath, key);

Console.WriteLine("Файл зашифрован");

DecryptFile(encryptedFilePath, decryptedFilePath, key);

Console.WriteLine("Файл расшифрован");

}

static void EncryptFile(string inputFilePath, string outputFilePath, byte key)

{

using (FileStream fileStream = new FileStream(inputFilePath, FileMode.Open, FileAccess.Read))

using (FileStream outputStream = new FileStream(outputFilePath, FileMode.Create, FileAccess.Write))

{

int byteRead;

while ((byteRead = fileStream.ReadByte()) != -1)

{

byte encryptedByte = (byte)(byteRead ^ key);

outputStream.WriteByte(encryptedByte);

}

}

}

static void DecryptFile(string inputFilePath, string outputFilePath, byte key)

{

using (FileStream fileStream = new FileStream(inputFilePath, FileMode.Open, FileAccess.Read))

using (FileStream outputStream = new FileStream(outputFilePath, FileMode.Create, FileAccess.Write))

{

int byteRead;

while ((byteRead = fileStream.ReadByte()) != -1)

{

byte decryptedByte = (byte)(byteRead ^ key);

outputStream.WriteByte(decryptedByte);

}

}

}

}

****

.

**Контрольные вопросы:**

1. **Что такое файл?**

это набор данных, который хранится на внешнем запоминающем устройстве (например на жестком диске).

1. **Как создать файл?**

static void Main(string[] args)

{  
 File.Create("D:\\new\_file.txt");

}

1. **Как удалить файл?**

static void Main(string[] args)  
{  
   File.Delete("d:\\test.txt"); //удаление файла   
}

1. **Что такое поток?**

это абстрактное представление данных (в байтах), которое облегчает работу с ними.

1. Чтение из файла и запись в файл

**FileStream** - представляет поток, который позволяет выполнять операции чтения/записи в файл.

Для чтения текстовых данных используется класс **StreamReader**.

Для записи данных в поток используется класс **StreamWriter**.

1. **Как создать/ удалить папку?**

Как создать папку  
 С помощью статического метода **CreateDirectory()** класса **Directory**:

static void Main(string[] args)  
{   
  Directory.CreateDirectory("d:\\new\_folder");  
}  
**Как удалить папку**Для удаления папок используется метод **Delete()**:  
static void Main(string[] args)  
{   
 Directory.Delete("d:\\new\_folder"); //удаление пустой папки   
}  
 **Если папка не пустая, необходимо указать параметр рекурсивного удаления** - true:

static void Main(string[] args)

{   
 Directory.Delete("d:\\new\_folder", true); //удаление папки, и всего, что внутри  
}

**Вывод:** получил навыки работы с файловыми потоками в С#.