|  |  |
| --- | --- |
| http://www.pl130.ru/doc/index/%D0%9B%D0%BE%D0%B3%D0%BE%D1%82%D0%B8%D0%BF22.jpg | Санкт-Петербургское государственное бюджетное  профессиональное образовательное учреждение  "Колледж электроники и приборостроения" |

отчет

**по практической работе №9**

**по дисциплине «Основы алгоритмизации и программирования»**

Тема: Файлы последовательного доступа

|  |  |  |
| --- | --- | --- |
| Студент гр. 13 ИТ |  | Липинский К.С. |
| Преподаватель |  | Счастливцев А.К. |

Санкт-Петербург

2023

**Цель работы.**

Создать приложение для чтения текстовых файлов с помощью последовательного доступа.

**Выполнение работы (вариант 2).**

**Задание 1.**

namespace Practice;

using System.IO;

using static Console;

class Program

{

static void Main()

{

int count = 0; string line;

Write("Enter the symbol you want to find: ");

char sym = Convert.ToChar(ReadLine()!);

using StreamReader fileIn = new

(@"C:\Users\konnl\OneDrive\Документы\Колледж\Программирование\Практическая работа 9\1.txt");

{

while (fileIn.EndOfStream == false)

{

if ((char)fileIn.Read() == sym)

{

count++;

}

}

WriteLine(count);

}

}

}

****

**Задание 2.**

namespace Practice;

using System.IO;

using static Console;

class Program

{

static void Main()

{

int count = 0; string i;

using StreamReader stream = new

("C:\\Users\\konnl\\OneDrive\\Документы\\Колледж\\Программирование\\Практическая работа 8\\2.txt");

{

while ((i = stream.ReadLine()!) != null)

{

if (i[0] == i[^1])

{

count++;

}

}

}

WriteLine(count);

}

}

****

**Задание 3.**

namespace Practice;

using System.IO;

using static Console;

class Program

{

static void Main()

{

string file =

@"C:\Users\konnl\OneDrive\Документы\Колледж\Программирование\Практическая работа 9\3.txt";

string pnumfile =

@"C:\Users\konnl\OneDrive\Документы\Колледж\Программирование\Практическая работа 9\pnum.txt";

string nnumfile =

@"C:\Users\konnl\OneDrive\Документы\Колледж\Программирование\Практическая работа 9\nnum.txt";

GenNums(file);

using StreamWriter pfile = new(pnumfile);

using StreamWriter nfile = new(nnumfile);

using StreamReader stream = new(file);

{

string output = stream.ReadToEnd();

foreach (string line in output.Split(" "))

{

try

{

if (Convert.ToInt32(line) > 0)

{

pfile.WriteLine(line);

}

else if (Convert.ToInt32(line) < 0)

{

nfile.WriteLine(line);

}

}

catch

{

}

}

}

}

static void GenNums(string file)

{

Random random = new();

using StreamWriter numfile = new(file);

{

for (int i = 0; i < 50; i++)

{

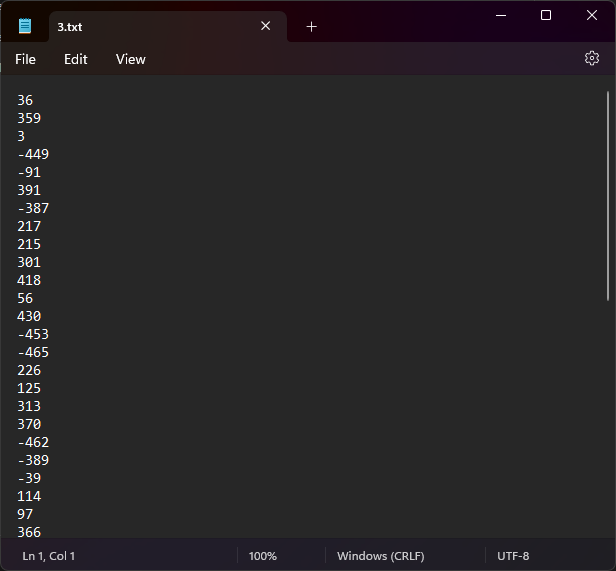
numfile.Write($"{random.Next(-500, 500)} ");

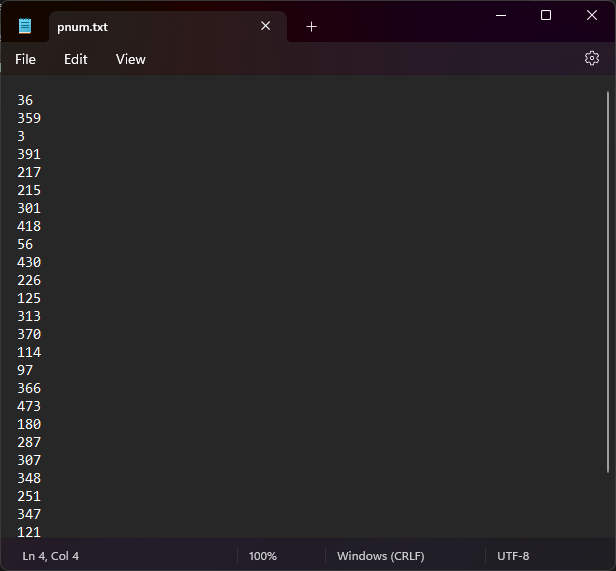
}

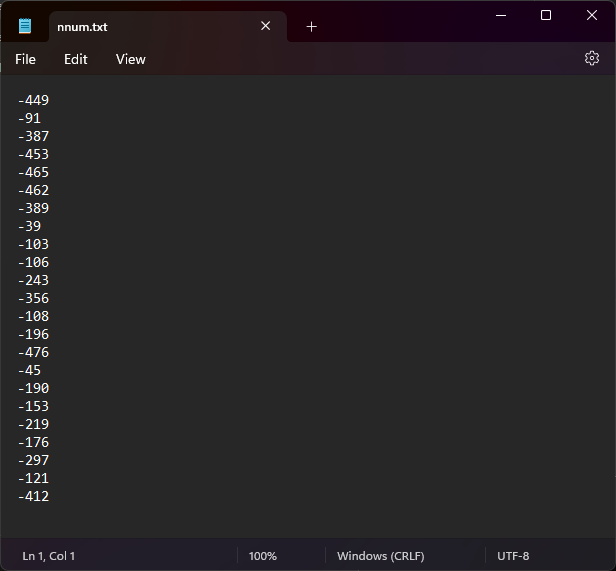
}

}

}



****

****

**Вопросы.**

1. Отличия файлов последовательного доступа от файлов прямого доступа:

* **Порядок доступа:** Файлы последовательного доступа обеспечивают последовательный доступ к данным, тогда как файлы прямого доступа позволяют обращаться к данным по произвольному порядку.
* **Перемещение:** В файлах последовательного доступа чтение или запись данных происходит последовательно, без возможности прямого перемещения к определенной позиции. В файлах прямого доступа можно перемещаться по файлу и читать или записывать данные в произвольных местах.
* **Эффективность:** Файлы последовательного доступа могут быть более эффективными при работе с большими объемами данных, так как не требуют постоянного перемещения позиции файла. Файлы прямого доступа могут быть более эффективными для случаев, когда требуется частый и произвольный доступ к данным.

1. Запись в файл последовательного доступа осуществляется с помощью классов StreamWriter, а чтение – StreamReader.
2. Для открытия файлов используется директива using, для закрытия файлов при этом необязательно использовать метод, но он есть – close().