|  |  |
| --- | --- |
| 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 №10**

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

Тема: Типизированные файлы

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

Санкт-Петербург

2023

**Цель работы.**

Освоить методы работы с типизированными файлами.

**Дисклеймер.**

BinaryFormatter использоваться не будет, поскольку начиная с версии .NET 8 его использование считается небезопасным. Вместо этого будет использоваться BinaryReader и BinaryWriter.

**Выполнение работы.**

**Задание 1.**

namespace Practice;

using System.IO;

using System.Text;

using static Console;

[Serializable]

public struct Address

{

public string City;

public string District;

public string Street;

public string ZipCode;

public string Phone;

}

class Program

{

static void Main()

{

string file =

"C:\\Users\\konnl\\OneDrive\\Документы\\Колледж\\Программирование\\Практическая работа 10\\addresses.dat";

Address[] addresses = new Address[5];

addresses[0].City = "Москва";

addresses[1].City = "Санкт-Петербург";

addresses[2].City = "Калининград";

addresses[3].City = "Екатеринбург";

addresses[4].City = "Саратов";

var output = File.OpenWrite(file);

BinaryWriter bw = new(output, Encoding.UTF8, false);

foreach (var elem in addresses)

{

bw.Write(elem.City);

}

output.Close();

var input = File.OpenRead(file);

BinaryReader br = new(input, Encoding.UTF8, false);

Address[] loadedAddresses = new Address[addresses.Length];

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

{

loadedAddresses[i].City = br.ReadString();

}

input.Close();

foreach (var elem in loadedAddresses)

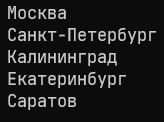
{

WriteLine(elem.City);

}

}

}



**Задание 2.**

using var output = File.OpenWrite(file);

{

using BinaryWriter bw = new(output, Encoding.UTF8, false);

{

foreach (var elem in addresses)

{

bw.Write(elem.City);

}

}

}

Address[] loadedAddresses = new Address[addresses.Length];

using var input = File.OpenRead(file);

{

using BinaryReader br = new(input, Encoding.UTF8, false);

{

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

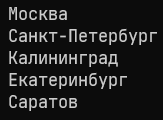
{

loadedAddresses[i].City = br.ReadString();

}

}

}

****

**Задание 3.**

foreach (var elem in loadedAddresses)

{

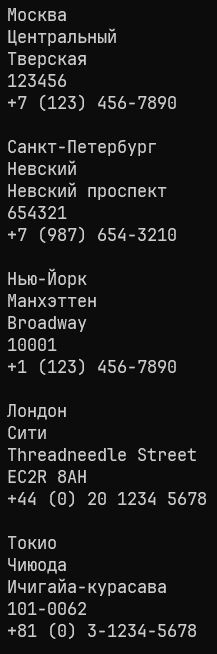
if (elem.City[0] == 'С')

WriteLine(elem.City);

}



**Задание 4.**



**Вопросы.**

1. Типизированный файл – файл, в котором представлены типизированные данные одного типа.
2. Для работы с типизированными файлами можно использовать классы BinaryReader (для чтения) и BinaryWriter (для записи). Существуют так же и другие классы для различных форматов сериализации (например, XML).