

Министерство науки и высшего образования Российской Федерации  
Муромский институт (филиал)  
Федерального государственного бюджетного образовательного учреждения  
высшего образования  
«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет \_\_\_\_\_ ФИТР \_\_\_\_\_

Кафедра \_\_\_\_\_ ПИН \_\_\_\_\_

## ЛАБОРАТОРНАЯ РАБОТА №6

По \_\_\_\_\_ Объектно-ориентированному программированию \_\_\_\_\_

Тема \_\_\_\_\_ Чтение и запись текстовых файлов. StreamReader и StreamWriter \_\_\_\_\_

Руководитель

Привезенцев Д.Г.

\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Студент \_\_\_\_\_ ПИН - 121 \_\_\_\_\_

\_\_\_\_\_  
(группа)

Ермилов М.В.

\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Муром 2022

## Лабораторная работа №6

**Тема:** Чтение и запись текстовых файлов. StreamReader и StreamWriter

### Ход работы:

#### Задание:

создать текстовый файл с произвольной информацией.

Организовать просмотр содержимого файла. Организовать чтение и обработку данных из файла в соответствии с индивидуальным заданием. Сохранить полученные результаты в новый текстовый файл.

«Покупатель»:

фамилия; имя; отчество; пол; национальность; рост; вес; дата рождения (год, месяц число); номер телефона; домашний адрес (почтовый индекс, страна, область, район, город, улица, дом, квартира); номер кредитной карточки; банковского счета.  
Вывести данные о покупателях с города Муром.

#### Код основной программы:

```
class Program
{
    static Random random = new Random();
    static void Main(string[] arg)
    {
        Directory.CreateDirectory("D://Laba6/");
        for (int i = 0; i < 100; i++)
        {
            Buyer buyer = GeneratorBuyer();
            buyer.Save($"D://Laba6/{i}.txt");
            if (buyer.Address.City.Equals("Муром"))
                Console.WriteLine($"User %i%\n{buyer}");
        }
        static BuyerName GeneratorName()
        {
            string[] name = { "Михаил", "Фёдор", "Денис", "Альберт" };
            string[] lastname = { "Ермилов", "Тютчев", "Привезенцев", "Эйнштейн" };
            string[] patranomic = { "Владимирович", "Иванович", "Геннадьевич",
"Германович" };
            return new BuyerName(name[random.Next(0, 4)], lastname[random.Next(0, 4)],
patranomic[random.Next(0, 4)]);
        }
        static BuyerCash GeneratorCash()
        {
            return new BuyerCash($"{random.Next(0, 10000)}{random.Next(0,
10000)}{random.Next(0, 10000)}{random.Next(0, 10000)}");
        }
        static BuyerNumber GeneratorNumber(Data.ISO_3166_1 Country)
```

					МИВУ 09.02.03 – N.00n	Лист
Изм.	Лист	№ докум.	Подпись	Дата		2

```

        {
            return new BuyerNumber(Country, $"{random.Next(0, 1000000)}{random.Next(0, 1000000)}");
        }
        static BuyerAddress GeneratorAddress(Data.ISO_3166_1 Country)
        {
            Dictionary<Data.ISO_3166_1, List<Data.ISO_3166_2>> a = new
            Dictionary<Data.ISO_3166_1, List<Data.ISO_3166_2>>()
            {
                { Data.ISO_3166_1.RU, new
                List<Data.ISO_3166_2>(){Data.ISO_3166_2.RU_MOS, Data.ISO_3166_2.RU_VLA } },
                { Data.ISO_3166_1.IT, new
                List<Data.ISO_3166_2>(){Data.ISO_3166_2.IT_82, Data.ISO_3166_2.IT_34 } },
                { Data.ISO_3166_1.CA, new
                List<Data.ISO_3166_2>(){Data.ISO_3166_2.CA_BC, Data.ISO_3166_2.CA_AB } },
            };
            int b = random.Next(0, 2);
            int c = random.Next(0, 2);
            string[] d =
            {"Пушкинская", "Лазаревская", "Ленина", "Рождественская", "Московская", "Льва Толстого" };
            return new BuyerAddress( Country, a[Country][b], c, d[random.Next(0, 6)],
            random.Next(1, 100), random.Next(1, 20), (uint)random.Next(100000, int.MaxValue));
        }
        static Buyer GeneratorBuyer()
        {
            int[] c = {380,124,643 };
            Data.ISO_3166_1 C = (Data.ISO_3166_1)c[random.Next(0, 3)];

            return new Buyer(GeneratorName(),GeneratorAddress(C),GeneratorNumber(C),
            GeneratorCash());
        }
    }

```

**Код конструктора класса для хранения данных покупателей, записью и чтением этих данных из файлов:**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

class Buyer
{
    public BuyerName Name;
    public BuyerAddress Address;
    public BuyerNumber Number;
    public BuyerCash Cash;

    //без параметров
    public Buyer() : this(new BuyerName(), new BuyerAddress(), new BuyerNumber(),
    new BuyerCash()) { }

    //1 параметр из 4
    public Buyer(BuyerName Name) : this(Name, new BuyerAddress(), new BuyerNumber(),
    new BuyerCash()) { }
    public Buyer(BuyerAddress Address) : this(new BuyerName(), Address, new
    BuyerNumber(), new BuyerCash()) { }
    public Buyer(BuyerNumber Number) : this(new BuyerName(), new BuyerAddress(),
    Number, new BuyerCash()) { }
    public Buyer(BuyerCash Cash) : this(new BuyerName(), new BuyerAddress(), new
    BuyerNumber(), Cash) { }

    //2 параметра из 4

```

					МИВУ 09.02.03 – N.00n	Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

```

        public Buyer(BuyerName Name, BuyerAddress Address) : this(Name, Address, new
BuyerNumber(), new BuyerCash()) { }
        public Buyer(BuyerName Name, BuyerNumber Number) : this(Name, new BuyerAddress(),
Number, new BuyerCash()) { }
        public Buyer(BuyerName Name, BuyerCash Cash) : this(Name, new BuyerAddress(),
new BuyerNumber(), Cash) { }
        public Buyer(BuyerAddress Address, BuyerNumber Number) : this(new BuyerName(),
Address, Number, new BuyerCash()) { }
        public Buyer(BuyerAddress Address, BuyerCash Cash) : this(new BuyerName(),
Address, new BuyerNumber(), Cash) { }
        public Buyer(BuyerNumber Number, BuyerCash Cash) : this(new BuyerName(), new
BuyerAddress(), Number, Cash) { }

        //3 параметра из 4
        public Buyer(BuyerName Name, BuyerAddress Address, BuyerNumber Number) :
this(Name, Address, Number, new BuyerCash()) { }
        public Buyer(BuyerName Name, BuyerAddress Address, BuyerCash Cash) : this(Name,
Address, new BuyerNumber(), Cash) { }
        public Buyer(BuyerName Name, BuyerNumber Number, BuyerCash Cash) : this(Name,
new BuyerAddress(), Number, Cash) { }
        public Buyer(BuyerAddress Address, BuyerNumber Number, BuyerCash Cash) :
this(new BuyerName(), Address, Number, Cash) { }

        //все параметры (основной конструктор)
        public Buyer(BuyerName Name, BuyerAddress Address, BuyerNumber Number, BuyerCash
Cash)
        {
            this.Name = Name;
            this.Address = Address;
            this.Number = Number;
            this.Cash = Cash;
        }

        public Buyer(string path)
        {
            Buyer a = Load(path);
            this.Name = a.Name;
            this.Address = a.Address;
            this.Number = a.Number;
            this.Cash = a.Cash;
        }

        public void Save(string path)
        {
            string number = Number.ToData();
            string cash = Cash.ToData();
            string address = Address.ToData();
            string name = Name.ToData();
            string text = $"{name}\n{address}\n{number}\n{cash}";
            try
            {
                using (StreamWriter sw = new StreamWriter(path, false,
System.Text.Encoding.Default))
                {
                    sw.WriteLine(text);
                }
            }
            catch (Exception e)
            {
                Console.WriteLine(e.Message);
            }
        }

        static public Buyer Load(string path)

```

					МИВУ 09.02.03 – N.00n	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

```

{
    List<string> a = new List<string>();
    try
    {
        using (StreamReader sr = new StreamReader(path,
System.Text.Encoding.Default))
        {
            string line;
            while ((line = sr.ReadLine()) != null)
            {
                a.Add(line);
            }
        }
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
    }
    return new Buyer(BuyerName.GetData(a[0]), BuyerAddress.GetData(a[1]),
BuyerNumber.GetData(a[2]), BuyerCash.GetData(a[3]));
}

public override string ToString()
{
    return $"{Name}\n{Address}\n{Number}\n{Cash}";
}
}

```

### Код конструктора класса для хранения имени:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

class BuyerName
{
    public string Name;
    public string LastName;
    public string Patronymic;
    public BuyerName(string Name, string LastName, string Patronymic)
    {
        this.Name = Name;
        this.LastName = LastName;
        this.Patronymic = Patronymic;
    }
    public BuyerName(string Name, string LastName) : this(Name, LastName, "") { }
    public BuyerName(string Name) : this(Name, "", "") { }
    public BuyerName() : this("NoName", "", "") { }

    public override string ToString()
    {
        string re = "";
        if (!LastName.Equals("")) re += $"Фамилия: {LastName}; ";
        if (!Name.Equals("")) re += $"Имя: {Name}; ";
        if (!Patronymic.Equals("")) re += $"Отчество: {Patronymic}; ";
        return re.Trim();
    }
    public string ToData()
    {
        return $"{Name}:{LastName}:{Patronymic}";
    }
    static public BuyerName GetData(string data)
    {
        string[] a = data.Split(":");
        return new BuyerName(a[0], a[1], a[2]);
    }
}

```

					МИВУ 09.02.03 – N.00n	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

```

    }

}

Код конструктора класса для хранения номера телефона:

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

class BuyerNumber
{
    private long _number = 0;
    private Data.ISO_3166_1 _code = 0;
    private const long MaxNumber = 10000000000;

    public BuyerNumber() { }
    public BuyerNumber(Data.ISO_3166_1 Country, long Number)
    {
        _code = Country;
        _number = Number % MaxNumber;
    }
    public BuyerNumber(Data.ISO_3166_1 Country, string Number) : this (Country,
long.Parse(Number)) { }

    public override string ToString()
    {
        int code = _code != 0 ? Data.NumberCode[_code] : 0;
        long a = _number;
        int b = (int)(a / (MaxNumber / 1000));
        a %= MaxNumber / 1000;
        int c = (int)(a / 10000);
        a %= 10000;
        int d = (int)(a / 100);
        a %= 100;

        return $"{code} {(b,3)} {(c,3)}-{(d,2)}-{((int)a,2)}";
    }
    private string _(int num, int n)
    {
        if(n == 3)
        {
            if (num > 99) return $"{num}";
            if (num > 9) return $"0{num}";
            return $"00{num}";
        }
        else if(n == 2)
        {
            if (num > 9) return $"{num}";
            return $"0{num}";
        }
        return $"{num}";
    }

    public string ToData()
    {
        return $"{(int)_code}:{_number}";
    }
    public static BuyerNumber GetData(string data)
    {
        string[] a = data.Split(":");

```

					МИВУ 09.02.03 – N.00n	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

```

        Data.ISO_3166_1 code = (Data.ISO_3166_1)int.Parse(a[0]);
        return new BuyerNumber(code, a[1]);
    }
}

```

### Код конструктора класса для хранения номера карты:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
/* Примечание:
 * если использовать хранение сразу в string, то места будет занято в несколько раз
 * больше
 * long = 8 байт
 * char = 2 байта
 * string = length * char, в нашем случае от 16 до 19 (зависит есть ли пробелы)
 * string = 2*16 : 2*19 = 32 байт : 38 байт
 * память экономится в 4 : 4.75 раз
 */
class BuyerCash
{
    long _number = 0;
    const long _Max = 1000000000000000000;

    public BuyerCash() { }
    public BuyerCash(long Number) { _number = Number % _Max; }
    public BuyerCash(string Number) : this(long.Parse(Number.Replace(" ", ""))) { }

    public override string ToString()
    {
        int a = (int)(_number / (_Max / 10000));
        int b = (int)((_number / (_Max / 100000000)) % 10000);
        int c = (int)((_number / (_Max / 1000000000000)) % 10000);
        int d = (int)(_number % 10000);
        return $"{_(a)} {_(b)} {_(c)} {_(d)}";
    }
    private string _(int num)
    {
        if (num > 999) return $"{num}";
        else if (num > 99) return $"0{num}";
        else if (num > 9) return $"00{num}";
        return $"000{num}";
    }
    public string ToData()
    {
        return $"{_number}";
    }
    static public BuyerCash GetData(string data)
    {
        return new BuyerCash(data);
    }
}

```

### Код конструктора класса для хранения адреса:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

class BuyerAddress
{
    Data.ISO_3166_1 _country = 0;
    Data.ISO_3166_2 _district = 0;
}

```

					МИВУ 09.02.03 – N.00n	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

```

    int _city = -1;

    public Data.ISO_3166_1 CountryCode
    {
        get { return _country; }
        set
        {
            if( _country != value)
            {
                _district = 0;
                _city = -1;
            }
            _country = value;
        }
    }
    public Data.ISO_3166_2 DistrictCode
    {
        get { return _district; }
        set
        {
            if (_district != value && Data.District[_country].ContainsKey(value))
            {
                _city = -1;
                _district = value;
            }
            else if(!Data.District[_country].ContainsKey(value))
            {
                _country = 0;
                _city = -1;
                _district = 0;
            }
        }
    }
    public int CityCode
    {
        get { return _city; }
        set { _city = value; }
    }

    public string Street = "";
    public int Home = 0;
    public int Apartment = 0;
    public uint MailIndex = 0;

    public string Country { get { return Data.Country.ContainsKey(_country) ?
Data.Country[_country] : "None"; } }
    public string District { get { return Data.District.ContainsKey(_country) ?
(Data.District[_country].ContainsKey(_district) ? Data.District[_country][_district] :
"None") : "None"; } }
    public string City { get {
        return Data.City.ContainsKey(_country) ?
            (Data.City[_country].ContainsKey(_district) ?
                (_city != 0 && Data.City[_country][_district].Count > _city ?
Data.City[_country][_district][_city] : "None")
                : "None")
            : "None"; } }

    public BuyerAddress() { }
    public BuyerAddress(Data.ISO_3166_1 Country, Data.ISO_3166_2 District, int City,
string Street, int Home, int Apartment, uint MailIndex)
    {
        CountryCode = Country;
        DistrictCode = District;
    }

```

					МИВУ 09.02.03 – N.00n	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		



```

        CityCode = City;
        this.Street = Street;
        this.Home = Home;
        this.Apartment = Apartment;
        this.MailIndex = MailIndex;
    }

    public string ToData()
    {
        return
        $"{(int)_country}:{(int)_district}:{_city}:{Street}:{Home}:{Apartment}:{MailIndex}";
    }

    static public BuyerAddress GetData(string data)
    {
        string[] a = data.Split(":");
        return new BuyerAddress(
            (Data.ISO_3166_1)int.Parse(a[0]),
            (Data.ISO_3166_2)int.Parse(a[1]),
            int.Parse(a[2]),
            a[3],
            int.Parse(a[4]),
            int.Parse(a[5]),
            uint.Parse(a[6])
        );
    }

    public override string ToString()
    {
        return $"Страна: {Country}; Область: {District}; Город: {City}; Улица: {Street};
        Дом: {Home}; Квартира: {Apartment}; Почтовый индекс: {MailIndex}";
    }
}

```

**Код статичного класса библиотеки, для хранения данных городов, стран:**

### 1 часть класса:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
/* Примечание:
 * Данный класс представляет из себя библиотеку с городами, областями и странами
 * ISO – мировой стандарт
 * 3166 – стандарт обозначения государств и зависимых территорий, а также основных
административных образований внутри государств.
 * 3166-1 – стандарт обозначения государств
 * 3166-2 – стандарт обозначения областей N государства
 *
 * Данный класс не является полной библиотекой со всеми странами, регионами и городами.
 * Тут показана лишь вариация того, как можно систематизировать большие объемы данных в
код, с дальнейшей работой.
 *
 * Так-же при желании можно доработать данный класс, добавив переводы на разные языки
мира, улицы, дома.
 * Но это будет требовать огромных ресурсозатрат, которых у обычного студента попросту
нет.
 */
static partial class Data
{
    static public Dictionary<ISO_3166_1, string> Country
    {
        get { return _country; }
    }
    static public Dictionary<ISO_3166_1, Dictionary<ISO_3166_2, string>> District

```

					МИВУ 09.02.03 – N.00n	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

```

        {
            get { return _district; }
        }
        static public Dictionary<ISO_3166_1, Dictionary<ISO_3166_2, List<string>>> City
        {
            get { return _city; }
        }
        static public Dictionary<ISO_3166_1, int> NumberCode
        {
            get { return _numbercode; }
        }
    }
}

```

## 2 часть класса:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

static partial class Data
{
    public enum ISO_3166_1
    {
        //alpha 2
        IT = 380,
        CA = 124,
        RU = 643,
        //alpha 3
        RUS = RU,
        ITA = IT,
        CAN = CA,
    }
    public enum ISO_3166_2
    {
        RU_MOS = 1,
        RU_VLA = 2,
        IT_34 = 3,
        IT_82 = 4,
        CA_AB = 5,
        CA_BC = 6,
    }
}
}

```

## 3 часть класса:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

static partial class Data
{
    private static Dictionary<ISO_3166_1, int> _numbercode = new
Dictionary<ISO_3166_1, int>()
    {
        {ISO_3166_1.CA, 1},
        {ISO_3166_1.RU, 7},
        {ISO_3166_1.IT, 39},
    };
}
}

```

					МИВУ 09.02.03 – N.00n	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

#### 4 часть класса:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

static partial class Data
{
    static private Dictionary<ISO_3166_1, string> _country = new
Dictionary<ISO_3166_1, string>(){
        {ISO_3166_1.RU, "Россия"},
        {ISO_3166_1.IT, "Италия"},
        {ISO_3166_1.CA, "Канада"},
    };
}
```

#### 5 часть класса:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

static partial class Data
{
    static private Dictionary<ISO_3166_1, Dictionary<ISO_3166_2, string>> _district
= new Dictionary<ISO_3166_1, Dictionary<ISO_3166_2, string>>(){
    {
        ISO_3166_1.RU, new Dictionary<ISO_3166_2, string>(){
            { ISO_3166_2.RU_MOS, "Московская Область"},
            { ISO_3166_2.RU_VLA, "Владимирская Область"},
        },
        {
            ISO_3166_1.IT, new Dictionary<ISO_3166_2, string>(){
                { ISO_3166_2.IT_34, "Венеция"},
                { ISO_3166_2.IT_82, "Сицилия"},
            },
            {
                ISO_3166_1.CA, new Dictionary<ISO_3166_2, string>(){
                    { ISO_3166_2.CA_AB, "Альберта"},
                    { ISO_3166_2.CA_BC, "Британская Колумбия"},
                },
            },
        },
    };
}
```

#### 6 часть класса:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

static partial class Data
{
```

					МИВУ 09.02.03 – N.00n	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

```

static private Dictionary<ISO_3166_1, Dictionary<ISO_3166_2, List<string>>>
_city = new Dictionary<ISO_3166_1, Dictionary<ISO_3166_2, List<string>>>()
{
    {
        ISO_3166_1.RU, new Dictionary<ISO_3166_2, List<string>>()
        {
            { ISO_3166_2.RU_MOS, new List<string>() { "Москва",
"Внуково" } },
            { ISO_3166_2.RU_VLA, new List<string>() { "Владимир",
"Муром" } }
        }
    },
    {
        ISO_3166_1.IT, new Dictionary<ISO_3166_2, List<string>>()
        {
            { ISO_3166_2.IT_34, new List<string>() { "Венеция",
"Беллуно" } },
            { ISO_3166_2.IT_82, new List<string>() { "Палермо",
"Рагуза" } }
        }
    },
    {
        ISO_3166_1.CA, new Dictionary<ISO_3166_2, List<string>>()
        {
            { ISO_3166_2.CA_AB, new List<string>() { "Эдмонтон",
"Калгари" } },
            { ISO_3166_2.CA_BC, new List<string>() { "Виктория",
"Ванкувер" } }
        }
    }
};
}

```

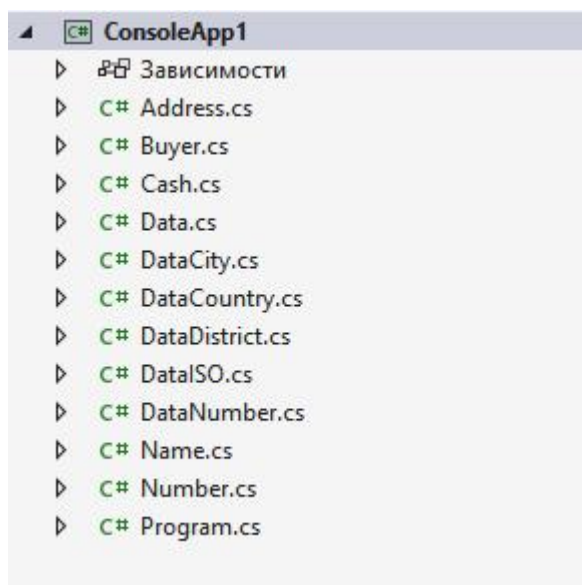


Рис. 1 - структура файлов программы

User №6  
 Фамилия: Ермилов; Имя: Денис; Отчество: Германович;  
 Страна: Россия; Область: Владимирская Область; Город: Муром; Улица: Московская; Дом: 89; Квартира: 5; Почтовый индекс: 148550289;  
 +7 957 897-11-52  
 2053 7594 5148 7534

User №9  
 Фамилия: Привезенцев; Имя: Михаил; Отчество: Геннадьевич;  
 Страна: Россия; Область: Владимирская Область; Город: Муром; Улица: Льва Толстого; Дом: 3; Квартира: 17; Почтовый индекс: 1047344225;  
 +7 087 913-84-09  
 4914 1601 3240 7744

User №14  
 Фамилия: Тютчев; Имя: Михаил; Отчество: Геннадьевич;  
 Страна: Россия; Область: Владимирская Область; Город: Муром; Улица: Московская; Дом: 76; Квартира: 3; Почтовый индекс: 1089757662;  
 +7 227 871-04-58  
 9835 1142 7224 1458

User №32  
 Фамилия: Ермилов; Имя: Фёдор; Отчество: Иванович;  
 Страна: Россия; Область: Владимирская Область; Город: Муром; Улица: Лазаревская; Дом: 55; Квартира: 18; Почтовый индекс: 1548322572;  
 +7 721 448-42-96  
 0816 0286 1508 1345

User №46  
 Фамилия: Тютчев; Имя: Альберт; Отчество: Иванович;  
 Страна: Россия; Область: Владимирская Область; Город: Муром; Улица: Рождественская; Дом: 43; Квартира: 6; Почтовый индекс: 158653202;  
 +7 002 991-11-16  
 0542 7435 1924 8369

User №52  
 Фамилия: Привезенцев; Имя: Альберт; Отчество: Геннадьевич;  
 Страна: Россия; Область: Владимирская Область; Город: Муром; Улица: Лазаревская; Дом: 30; Квартира: 9; Почтовый индекс: 1088953897;  
 +7 136 758-97-63  
 0058 9597 8469 7887

User №70  
 Фамилия: Тютчев; Имя: Денис; Отчество: Геннадьевич;  
 Страна: Россия; Область: Владимирская Область; Город: Муром; Улица: Ленина; Дом: 44; Квартира: 16; Почтовый индекс: 658167828;  
 +7 619 925-54-46  
 0045 6440 0942 5845

User №74  
 Фамилия: Ермилов; Имя: Альберт; Отчество: Иванович;  
 Страна: Россия; Область: Владимирская Область; Город: Муром; Улица: Ленина; Дом: 5; Квартира: 8; Почтовый индекс: 1769735384;  
 +7 347 011-16-06  
 9800 7085 6269 3574

Рис. 2 – пример работы программы

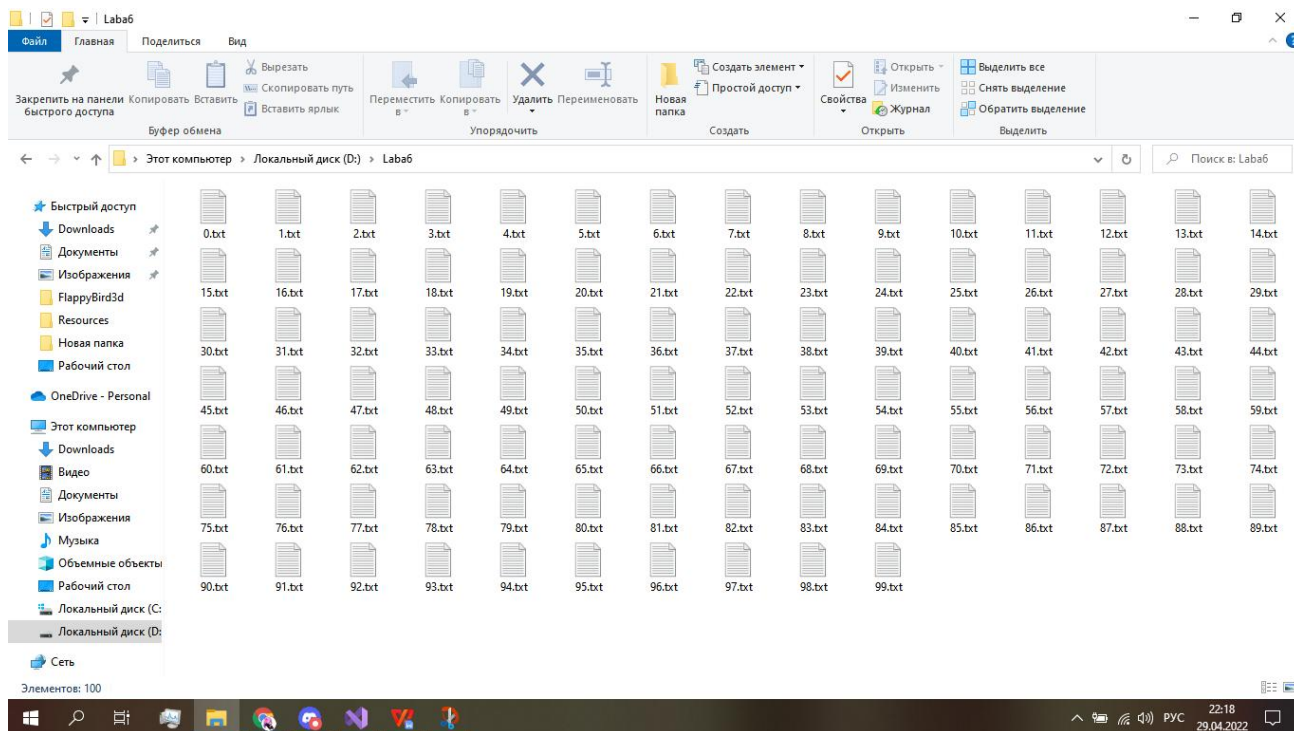


Рис. 3 - все сохраненные файлы с программы

					МИВУ 09.02.03 – N.00n	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

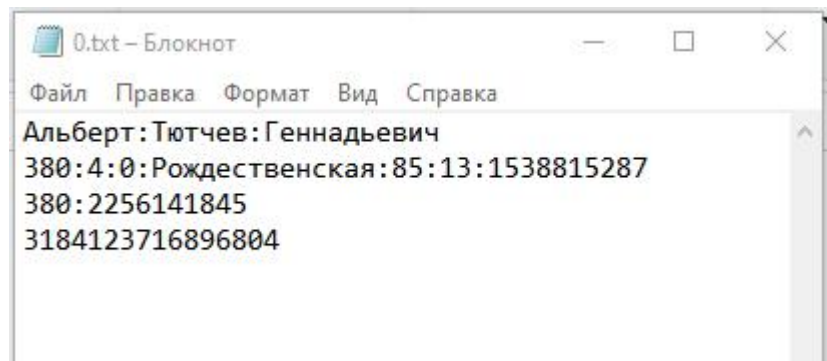


Рис. 4 - пример хранения класса с информацией о покупателе