|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 **Прикладная информатика**

**Отчет**

**по лабораторной работе № 9**

**Дисциплина:** Разработка приложений на языке C#

**Название работы:** Сериализация



Студент гр. ИУ6-72Б **\_\_**01.10.2022**\_\_\_\_\_\_\_ \_**И.С. Марчук\_

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** А.М. Минитаева

(Подпись, дата) (И.О. Фамилия)

Москва, 2022

**Цель работы:** изучение принципов сериализации и десериализации

**Задание:**

Требуется разработать программу, ведущую учёт заказов в магазине.

Классы:

* покупатель, с атрибутами: имя (string), адрес (string), скидка (double);
* товар: название (string) и цена (decimal);
* база данных товаров, хранящий ассоциативный массив («словарь») с информацией о товарах;
* orderLine с полями количество (int) и продукт (Product);
* order с полями номер заказа (int), клиент (Customer), скидка (decimal),

общая стоимость (decimal) и строки заказа (List<OrderLIne>).

Реализовать следующую логику основной программы:

1. Создаётся и заполняется база данных товаров (ассоциативный массив).
2. В консоли вводятся данные по конкретному покупателю, создаётся соответствующий объект.
3. Создаётся заказ для введённого ранее покупателя. Устанавливается скидка на заказ в соответствии со скидкой покупателя.
4. В цикле формируются необходимое количество строк заказа: вводятся коды товаров и количества их единиц.
5. Полная информация о заказе сохраняется в файле с заданным именем.

Создать методы, которые осуществляют сериализацию/ десериализацию объекта типа База данных товаров. Формат выбрать самостоятельно.

Листинг программы:

Program.cs:

using System;

namespace lab9

{

class Program

{

static void Main(string[] args)

{

// заполняем базу

Base b = new Base();

// сохраняем в файл

b.write();

}

}

}

Base.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Text.Json;

using System.IO;

using System.Diagnostics;

using System.Xml.Linq;

using System.Xml.Serialization;

using System.Runtime.Serialization;

namespace lab9

{

internal class Base

{

List<Order> orders = new List<Order>();

List<Product> products = new List<Product>();

public Base()

{

Console.WriteLine("Создание базы товаров: ");

products.Add(new Product("Банан", 1));

products.Add(new Product("Арбуз", 5));

products.Add(new Product("Баклажан", 12));

products.Add(new Product("Белый хлеб", 11));

products.Add(new Product("Икра белая", 10));

products.Add(new Product("Виноград", 6));

Init();

}

public void Init()

{

int id = 0;

Random random = new Random();

int count = random.Next(2, 4);

Console.WriteLine("Создание базы клиентов ("+ count+"шт.)");

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

{

Client cl = new Client();

var or = new Order(++id, cl);

for (int j = 0; j < random.Next(1, 3); j++)

{

OrderLine ol = new OrderLine(random.Next(1, 10), products[random.Next(products.Count)]);

or.AddOrderLine(ol);

}

orders.Add(or);

}

}

// сохранение в файл

public void write()

{

Console.WriteLine();

// сериализация в джсон

Console.WriteLine("Сохраняю в файл lab9\\bin\\Debug\\net6.0\\data\_json.txt");

File.WriteAllText("data\_json.txt", JsonSerializer.Serialize(orders));

Console.WriteLine("Сохранено");

}

}

}

Client.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab9

{

internal class Client

{

public string name { get; set; }

public string address { get; set; }

public int discount { get; set; }

public Client()

{

Console.WriteLine(" Создание нового клиента: ");

Console.WriteLine(" Введите имя: ");

this.name = Console.ReadLine();

Console.WriteLine(" Введите Адрес: ");

this.address = Console.ReadLine();

Console.Write(" Введите Скидку (0-99):");

int balance = 0;

while (!int.TryParse(Console.ReadLine(), out balance) || balance < 0|| balance > 99)

{

Console.WriteLine("Введено не корректное число!");

}

this.discount = balance;

Console.WriteLine(" Новый клиент создан: " + name

+ " Адрес: " + address + " Скидка: " + balance);

}

}

}

Order.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab9

{

internal class Order

{

public int id { get; set; }

public Client client { get; set; }

public decimal discond => client.discount;

public decimal price => getPrice();

public List<OrderLine> lines { get; }

private decimal getPrice()

{

decimal price = 0;

foreach (OrderLine line in lines)

{

price += line.product.price \* line.count;

}

return (price \* (100 - discond)) / 100;

}

public Order(int id, Client client)

{

Console.WriteLine(" Создание заказа id= "+ id+": ");

this.id = id;

this.client = client;

lines = new List<OrderLine>();

}

public void AddOrderLine(OrderLine ol)

{

lines.Add(ol);

}

}

}

OrderLine.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab9

{

internal class OrderLine

{

public int count { get; set; }

public Product product { get; set; }

public OrderLine(int count, Product product)

{

Console.WriteLine(" Пункт заказа: кол-во= " + count + " Товар=" + product.name + ": ");

this.count = count;

this.product = product;

}

}

}

Product.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab9

{

internal class Product

{

public string name { get; }

public decimal price { get; }

public Product(string name, decimal price)

{

Console.WriteLine(" Создание товара: " + name + " \tПо цене:" + price);

this.name = name;

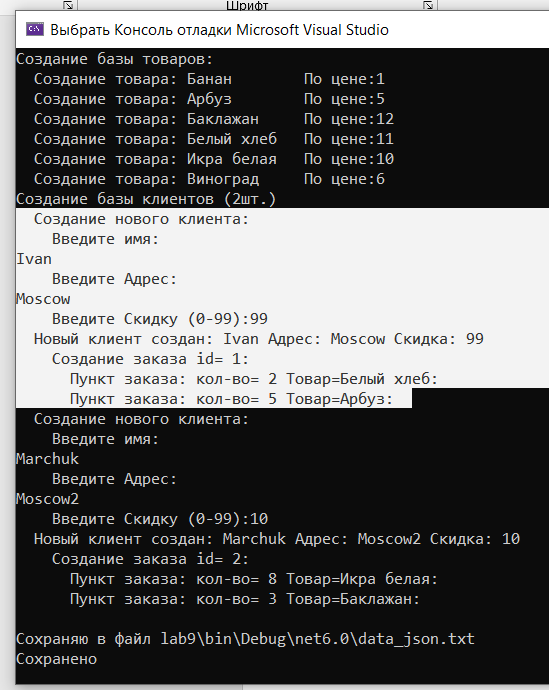
this.price = price;

}

}

}

Работа программы показана на рисунке 1.



*Рисунок 1 – Работа программы*

**Вывод:** в процессе выполнения лабораторной работы были изучены

средства сериализации и десериализации классов в C#.