



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 Информатика и вычислительная техника

О Т Ч Е Т

по лабораторной работе №9

Название: Сериализация

Дисциплина: Разработка приложений на языке C#

Студент

ИУ6-73Б
(Группа)

(Подпись, дата)

К.А. Логачев
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

А.М. Минитаева
(И.О. Фамилия)

Цель: изучение механизма сериализации в языке C#.

Задание:

Требуется разработать программу, ведущую учёт заказов в магазине.

Классы:

- (Покупатель) с тремя атрибутами: имя (string), адрес (string), скидка (double)
- (Товар). Поля, соответствующие названию (string) и цене (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.

Решение:

Было разработано приложение по работе с сериализацией. Код программы представлен на листинге 1.

Листинг 1 –Код основной программы

```
using System;
using
System.Collections.Generic;
using System.Text.Json;
using System.IO;

namespace lab9
{
    public class Customer
    {
        public string fio { get; set; }
        public string address{ get;
set; } public double discount{
get; set; }
        public Customer(string fio0, string address0, double discount0)
        {
            fio = fio0;
            address = address0;
            discount =
discount0;
        }
    }

    public class Product
    {
        public string name { get; set;
} public decimal price{ get;
set; }
        public void SetParams(string name0, decimal price0)
        {
            name = name0;
            price =
price0;
        }
    }

    public class Base
    {
        public Dictionary<string, Product> products { get;
set; } public Base()
        {
            products = new Dictionary<string, Product>();
        }
        public void AddProduct(string id, string name, decimal price)
        {
            Product product = new
Product();
            product.SetParams(name,
price); products.Add(id,
product);
        }
        public void Show()
        {
            foreach (KeyValuePair<string, Product> product in products)
            {
                Console.WriteLine("{0} {1} {2}", product.Key,
product.Value.name, product.Value.price);
            }
        }
    }
}
```

```

public void SaveBase()
{
    var options = new JsonSerializerOptions{
        WriteIndented = true
    };
    string json = JsonSerializer.Serialize(this,
options); File.WriteAllText("base.json", json);
    Console.WriteLine(json);
}

public void LoadBase()
{
    var options = new JsonSerializerOptions
    {
        WriteIndented = true
    };
    string json = File.ReadAllText("base.json");
    Base? restoredBase = JsonSerializer.Deserialize<Base>(json,
options); products = restoredBase.products;
    Console.WriteLine(json);
}

}

public class OrderLine
{
    public string product_id { get;
set; } public int count { get;
set; }
    public OrderLine(string id, int count0)
    {
        product_id =
id; count =
count0;
    }
}

public class Order
{
    public int number { get; set; }
    public Customer customer { get;
set; } public Base shopBase;
    public double discount { get; set;
} public decimal sum_price { get;
set; }
    public List<OrderLine> orderLines { get; set; }

    public Order(int number0, Customer customer0, Base base0)
    {
        number = number0;
        customer =
customer0; shopBase
= base0; sum_price
= 0;
        orderLines = new List<OrderLine>();
    }

    public void AddOrderLine(string id, int count)
    {
        orderLines.Add(new OrderLine(id, count));
        sum_price = sum_price + shopBase.products[id].price * count;
        //Console.WriteLine(shopBase.products[id].price);
        //Console.WriteLine(count);
        //Console.WriteLine(sum_price);
        discount = (double)sum_price * customer.discount;
    }

    public void SaveOrder()
    {
        string json =

```

```

        JsonSerializer.Serialize(this);
        File.WriteAllText("order.json", json);
        Console.WriteLine(json);
    }
}
class Program
{
    static void Main(string[] args)
    {
        Base shopBase = new
        Base(); string input =
        "";
        string id;
        string name;
        decimal
        price;
        Console.ForegroundColor =
        ConsoleColor.Yellow;
        Console.WriteLine("Заполнение БД");
        Console.ForegroundColor =
        ConsoleColor.White; while (input != "end")
        {
            Console.WriteLine("ID
            товара"); input =
            Console.ReadLine();
            if (input ==
                "end") break;
            id = input;

            Console.WriteLine("Имя
            товара"); name =
            Console.ReadLine();
            if (name ==
                "end")
                break;

            Console.WriteLine("Цена
            товара"); input =
            Console.ReadLine();
            if (input ==
                "end") break;
            price = decimal.Parse(input);

            shopBase.AddProduct(id, name, price);
        }
        Console.ForegroundColor = ConsoleColor.Yellow;
        Console.WriteLine("Данные в БД");
        Console.ForegroundColor = ConsoleColor.White;
        shopBase.Show();

        Console.ForegroundColor = ConsoleColor.Yellow;
        Console.WriteLine("Создание пользователя");
        Console.ForegroundColor = ConsoleColor.White;
        Console.WriteLine("ФИО пользователя");
        string fio = Console.ReadLine();
        Console.WriteLine("Адрес пользователя");
        string address = Console.ReadLine();
        Console.WriteLine("Скидка пользователя в
        %");
        double discount = double.Parse(Console.ReadLine()) / 100;
        Customer customer = new Customer(fio, address, discount);

        Console.ForegroundColor = ConsoleColor.Yellow;
        Console.WriteLine("Сбор заказа");
        Console.ForegroundColor = ConsoleColor.White;

        Order order = new Order(0, customer,
        shopBase); input = "";
        int count;
        while (input != "end")
    }
}

```

```

{
    Console.WriteLine("ID
товара"); input =
    Console.ReadLine();
    if (input ==
        "end") break;
    id = input;

    Console.WriteLine("Количество единиц
товара"); input = Console.ReadLine();
    if (input == "end")
        break;
    count = int.Parse(input);

    order.AddOrderLine(id, count);
}

Console.ForegroundColor =
ConsoleColor.Yellow;
Console.WriteLine("Данные заказа");
Console.ForegroundColor =
ConsoleColor.White; order.SaveOrder();
Console.ForegroundColor = ConsoleColor.Yellow;
Console.WriteLine("Данные в базе");
Console.ForegroundColor = ConsoleColor.White;
shopBase.SaveBase();
Console.ForegroundColor =
ConsoleColor.Yellow;
Console.WriteLine("Восстановленные данные в
базе"); Console.ForegroundColor =
ConsoleColor.White; shopBase.LoadBase();
}
}
}

```

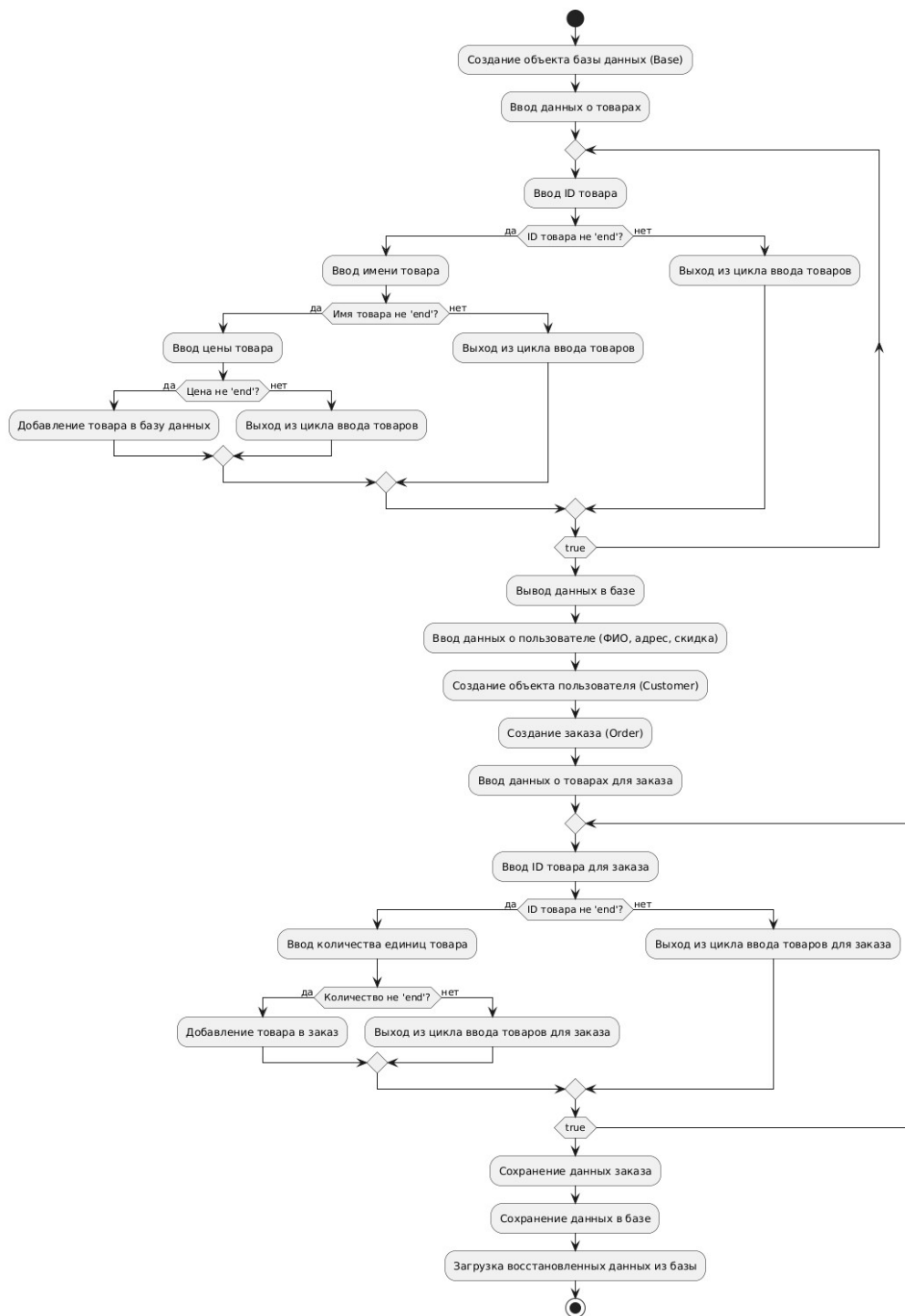


Рисунок 1 – Схема алгоритма

```

Заполнение БД
ID товара
1
Имя товара
lol
Цена товара
100
ID товара
2
Имя товара
kek
Цена товара
200
ID товара
end
Данные в БД
1 lol 100
2 kek 200
Создание пользователя
ФИО пользователя
LFA
Адрес пользователя
BMSTU
Скидка пользователя в %
10
Сбор заказа
ID товара
1
Количество единиц товара
4
ID товара
2
Количество единиц товара
3
ID товара
end
Данные заказа
{"number":0,"customer":{"fio":"LFA","address":"BMSTU","discount":0.1},"discount":100,"sum_price":1000,"orderLines":[{"product_id":"1","count":4},{"product_id":"2","count":3}]}
Данные в базе
{
  "products": {
    "1": {
      "name": "lol",
      "price": 100
    },
    "2": {
      "name": "kek",
      "price": 200
    }
  }
}
Восстановленные данные в базе
{
  "products": {
    "1": {
      "name": "lol",
      "price": 100
    },
    "2": {
      "name": "kek",
      "price": 200
    }
  }
}

```

Рисунок 2 – Результат работы программы

Вывод: в ходе выполнения лабораторной работы было разработано приложение по работе с сериализацией. Приложение работает корректно. Изучена работа с сериализацией в приложениях на языке C#.