Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

(Университет ИТМО)

Факультет **Прикладная информатика**

Образовательная программа **Мобильные и сетевые технологии**

Направление подготовки(специальность) **09.03.03 Прикладная информатика**

Лабораторная работа №1

По дисциплине «Объектно-ориентированное программирование»

Тема: Лабораторная работа C#.

Выполнил Каприн Семён Евгеньевич к3140.
Дата 25.09.2025

Санкт-Петербург 2025

Цель работы

Реализовать консольное приложение на C#, которое будет эмулировать вендинговый автомат, позволяющее пользователю:

- Посмотреть список доступных товаров с их ценами и количеством.
- Вставить монеты разных номиналов.
- Выбрать товар и получить его, если внесённой суммы достаточно.
- Получить сдачу (если нужно) и вернуть неиспользованные монеты при отмене операции.
- Администраторский режим для пополнения ассортимента и сбора собранных средств.

Ход работы

1) Основной цикл реализовал через while (true) и обработку строки через switch case:

```
while (true)
{
    var input = Console.ReadLine();
    if (string.IsNullOrEmpty(input)) continue;

    var parts = input.Trim().Split();
    var cmd = parts[0].ToLower();
    var arg = parts.Length > 1 ? parts[1].Trim() : string.Empty;
    var additionalArg1 = parts.Length > 2 ? parts[2].Trim() : string.Empty;
    var additionalArg2 = parts.Length > 3 ? parts[3].Trim() : string.Empty;

    switch (cmd)
```

2) Реализовал Баланс пользователя через int, Баланс машины через SortedDictionary для обработки разных номиналов.

```
private static int UserMoney { get; set; } = 0;
public static SortedDictionary<int, int> money = new SortedDictionary<int, int>();
```

3) Список продуктов – это List из экземпляров Product, которые имеют название, стоимость и количество

```
public class Product
    {
        public int productId { get; set; }
        public string productName { get; set; }
        public int productPrice { get; set; }
        public int productQuantity { get; set; }
    }
   }
```

```
public static List<Product> productsList { get; set; }
```

4) Все продукты находятся по имени в списке.

```
var existing = AtmDatabase.productsList.FirstOrDefault(p =>
   string.Equals(p.productName.ToLower(), name.ToLower()));
```

5) Если необходимо вывести деньги из машины, то вычитание происходит от большего номинала к меньшему, с помощью SortedDictionary и Reverse()

```
foreach (var x in AtmDatabase.money.Reverse())
{
   if (UserMoney <= 0)
      break;
   while (x.Key <= UserMoney && AtmDatabase.money[x.Key] > 0)
   {
      UserMoney -= x.Key;
      AtmDatabase.money[x.Key]--;
      returned += x.Key;
   }
}
```

6) Сохранение идет в файл Json через обертку

```
[System.Serializable]
class SaveWrapper
{
   public List<Product> ProductsSaveList { get; set; }
   public SortedDictionary<int, int> AtmSaveBank { get; set; }
}
```

```
public static void Save()
{
   var wrapper = new SaveWrapper()
   {
      ProductsSaveList = productsList,
      AtmSaveBank = money
   };
   var json = JsonConvert.SerializeObject(wrapper, Formatting.Indented);
   File.WriteAllText(SavePath, json);
}
```

```
public static void LoadProducts()
{
    if (!File.Exists(SavePath))
    {
        // Создам новый список продуктов, если нет текущего
        productsList = new List<Product>
        {
            new Product { productId = 1, productName = "Чипсы", productPrice = 123,
productQuantity = 2 },
            new Product { productId = 2, productName = "Вода", productPrice = 43,
productQuantity = 1 },
            new Product { productId = 3, productName = "Кола", productPrice = 87,
productQuantity = 1 }
        };
        money = new SortedDictionary<int, int>
        {
            { 1, 12 },
            { 2, 12 },
            { 5, 9 },
            { 10, 6 },
            { 25, 4 },
            { 50, 3 },
            { 100, 2 },
            { 500, 1 },
            { 1000, 1 }
        };
        Save();
        return;
    }

    string json = File.ReadAllText(SavePath);
    productsList =
JsonConvert.DeserializeObject<SaveWrapper>(json).ProductsSaveList;
    money = JsonConvert.DeserializeObject<SaveWrapper>(json).AtmSaveBank;
}
```

Ссылка на GitHub:

https://github.com/SUPER-PAU/VendingMachine-ConsoleApplication

Вывод

В результате проделанной работы мной были изучены основы ООП языка C#, а также было создано консольное приложение с базовыми операциями.