

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»
(Университет ИТМО)

Факультет **Инфокоммуникационных технологий**

Образовательная программа **Мобильные и облачные технологии**

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

Лабораторная работа №1

По дисциплине «Объекто-ориентированное программирование»

Тема: Разработка консольного приложения

Выполнил	Езиева М.Д. К3239
Проверил	Слюсаренко С.В.
Дата	<u>07.10.2025</u>

Санкт-Петербург 2025

Цель работы

Нужно разработать консольное приложение, 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

Создается класс `product`, который нужен для представления товара в автомате. В нем объявлены `public` 3 свойства: название, цена и оставшееся количество.

```
class Product
{
    public string Name { get; set; }
    public decimal Price { get; set; }
    public int Quantity { get; set; }

    public Product(string name, decimal price, int quantity)
    {
        Name = name;
        Price = price;
        Quantity = quantity;
    }

    public override string ToString()
    {
        return $"{Name} – {Price} руб (осталось {Quantity})";
    }
}
```

Создание класса `VendingMachine`, который описывает автомат внутри. В нем определены 4 `private` поля. Список товаров, словарь в котором хранятся данные о номинале монеты и их количестве, переменная которая содержит сумму внесенную пользователем, переменная в которой хранится сумма выручки.

```

class VendingMachine
{
    private List<Product> products;
    private Dictionary<int, int> coins;
    private decimal insertedAmount = 0;
    private decimal totalEarnings = 0;

    public VendingMachine()
    {

        products = new List<Product>
        {
            new Product("Кофе", 120, 10),
            new Product("Чай", 100, 10),
            new Product("Сникерс", 150, 10),
            new Product("Вода", 80, 10)
        };

        coins = new Dictionary<int, int>
        {
            { 10, 1000 },
            { 20, 1000 },
            { 50, 1000 },
            { 100, 1000 }
        };
    }
}

```

Метод DisplayProducts, который отвечает за отображение списка товаров. Выводит в консоль название, цену и количество.

```

public void DisplayProducts()
{
    Console.WriteLine("\nСписок товаров:");
    for (int i = 0; i < products.Count; i++)
    {
        Console.WriteLine($"{i + 1}. {products[i]}");
    }
}

```

Метод InsertCoin, который отвечает за внесение монет в автомат. Проверяет номинал монеты, добавляет ее в общую сумму и выводит в консоль сколько пользователь внес средств.

```

public void InsertCoin(int val)
{
    if (coins.ContainsKey(val))
    {
        coins[val]++;
        insertedAmount += val;
        Console.WriteLine($"Внесено: {val} руб | Всего: {insertedAmount}
руб");
    }
    else
    {
        Console.WriteLine("Такой монеты автомат не принимает");
    }
}

```

Метод SelectProduct, который реализует процесс покупки товара. Проверяет правильность ввода пользователя, количество товара, количество внесенных средств, списывает деньги, добавляет цену товара в выручку, отдает сдачу, выводит в консоль полученный товар.

```

public void SelectProduct(int index)
{
    if (index < 1 || index > products.Count)
    {
        Console.WriteLine("Неверный номер товара");
        return;
    }

    Product selected = products[index - 1];

    if (selected.Quantity <= 0)
    {
        Console.WriteLine("Товар закончился");
        return;
    }

    if (insertedAmount >= selected.Price)
    {
        selected.Quantity--;
        insertedAmount -= selected.Price;
        totalEarnings += selected.Price;
        Console.WriteLine($"Вы получили: {selected.Name}");
        GiveChange(insertedAmount);
        insertedAmount = 0;
    }
    else
    {
        Console.WriteLine("Недостаточно средств");
    }
}

```

```
    }  
}
```

Метод GiveChange для выдачи сдачи

```
public void GiveChange(decimal change)  
{  
    if (change > 0)  
        Console.WriteLine($"Ваша сдача: {change} руб");  
}
```

Метод ReturnCoins, который реализует отмену операции. Проверяет есть ли внесенные пользователем деньги, если да, то возвращает, если нет, то выводит сообщение что возвращать нечего

```
public void ReturnCoins()  
{  
    if (insertedAmount > 0)  
    {  
        Console.WriteLine($"Возврат: {insertedAmount} руб");  
        insertedAmount = 0;  
    }  
    else  
    {  
        Console.WriteLine("Вы ещё не вносили деньги");  
    }  
}
```

Метод adminMode, который реализует меню администратора. Сначала программа запрашивает пароль (1234). Потом открывает меню администратора, где можно пополнить запасы товара или забрать выручку.

```
public void adminMode()  
{  
    Console.WriteLine("\nВведите пароль администратора: ");  
    string password = Console.ReadLine();  
  
    if (password != "1234")  
    {  
        Console.WriteLine("Неверный пароль");  
        return;  
    }  
  
    int choice;  
    do  
    {
```

```

        Console.WriteLine("\n--- Админ-меню ---");
        Console.WriteLine("1. Пополнить товары");
        Console.WriteLine("2. Собрать выручку");
        Console.WriteLine("0. Выход");
        Console.Write("Выбор: ");
        choice = int.Parse(Console.ReadLine());

        switch (choice)
        {
            case 1:
                RefillProducts();
                break;
            case 2:
                result();
                break;
        }
    } while (choice != 0);
}

```

Метод RefillProducts, который отвечает за пополнение товара в автомате. Считывается введенное администратором число и прибавляется к текущему количеству товара.

```

private void RefillProducts()
{
    foreach (var product in products)
    {
        Console.Write($"Добавить товар {product.Name}: ");
        int qty = int.Parse(Console.ReadLine());
        product.Quantity += qty;
    }
    Console.WriteLine("Пополнение завершено");
}

```

Метод result, который нужен для получения и обнуление выручки.

```

private void result()
{
    Console.WriteLine($"Получено: {totalEarnings} руб");
    totalEarnings = 0;
}

```

Класс program, главная программа, которая управляет всей логикой автомата. Создается автомат, выводится меню, вызываются методы.

```

class Program
{
    static void Main()
    {
        VendingMachine machine = new VendingMachine();
        int choice;

        do
        {
            Console.WriteLine("\n--- ВЕНДИНГОВЫЙ АВТОМАТ ---");
            Console.WriteLine("1. Посмотреть товары");
            Console.WriteLine("2. Внести монеты");
            Console.WriteLine("3. Выбрать товар");
            Console.WriteLine("4. Отменить и вернуть деньги");
            Console.WriteLine("5. Войти как администратор");
            Console.WriteLine("0. Выход");
            Console.Write("Выбор: ");
            choice = int.Parse(Console.ReadLine());

            switch (choice)
            {
                case 1:
                    machine.DisplayProducts();
                    break;
                case 2:
                    Console.Write("Введите номинал монеты (10, 20, 50, 100):");

                    int coin = int.Parse(Console.ReadLine());
                    machine.InsertCoin(coin);
                    break;
                case 3:
                    machine.DisplayProducts();
                    Console.Write("Введите номер товара: ");
                    int productIndex = int.Parse(Console.ReadLine());
                    machine.SelectProduct(productIndex);
                    break;
                case 4:
                    machine.ReturnCoins();
                    break;
                case 5:
                    machine.adminMode();
                    break;
            }

            } while (choice != 0);

            Console.WriteLine("Спасибо за использование автомата!");
        }
    }
}

```

Вывод

В ходе лабораторной работы реализовано консольное приложение, эмулирующее функциональность вендингового автомата. Программа поддерживает операции покупки, возврата, выдачи сдачи и меню администратора. Полученные знания закрепили понимание принципов ООП (инкапсуляция, работа со списками и словарями, взаимодействие между объектами).