**Лабораторная работа №4**

**Тема: Работа с интерфейсами.**

**Выполнила студентка группы ИСП-223 Губарева Полина.**

**Вариант 4.**

**Вариант 4**

Задача:

Разработайте систему управления складом товаров с использованием интерфейсов в C#. Ваша программа должна позволять добавлять новые товары на склад, удалять товары, а также выполнять поиск товаров по различным критериям, таким как название, категория и количество.

Требования:

* Создайте интерфейс IProduct, который будет определять базовые свойства товара, такие как название, категория и количество. Включите в интерфейс метод для отображения информации о товаре.
* Реализуйте класс Product, который будет реализовывать интерфейс IProduct. Этот класс должен содержать поля для хранения информации о товаре, конструктор для инициализации полей и методы для доступа к ним.
* Создайте интерфейс IWarehouse, который будет определять методы для управления складом товаров, такие как добавление нового товара, удаление товара и поиск товаров по различным критериям.
* Реализуйте класс Warehouse, который будет реализовывать интерфейс IWarehouse. В этом классе должны быть методы для добавления новых товаров на склад, удаления товаров со склада, а также методы для поиска товаров по названию, категории или количеству.
* Напишите консольное приложение, которое будет использовать класс Warehouse для управления складом товаров. Приложение должно предоставлять пользователю возможность добавлять новые товары, удалять товары, а также выполнять поиск товаров по различным критериям.

using System;

using System.Collections.Generic;

// Интерфейс для товара

public interface IProduct

{

string Name { get; }

string Category { get; }

int Quantity { get; }

void DisplayInfo();

}

// Класс товара

public class Product : IProduct

{

public string Name { get; }

public string Category { get; }

public int Quantity { get; }

public Product(string name, string category, int quantity)

{

Name = name;

Category = category;

Quantity = quantity;

}

public void DisplayInfo()

{

Console.WriteLine($"Название: {Name}, Категория: {Category}, Количество: {Quantity}");

}

}

// Интерфейс для склада

public interface IWarehouse

{

void AddProduct(IProduct product);

void RemoveProduct(string name);

List<IProduct> FindProductsByName(string name);

List<IProduct> FindProductsByCategory(string category);

List<IProduct> FindProductsByQuantity(int quantity);

List<IProduct> GetAllProducts();

}

// Класс склада

public class Warehouse : IWarehouse

{

private List<IProduct> products = new List<IProduct>();

public void AddProduct(IProduct product)

{

if (products.Exists(p => p.Name.Equals(product.Name, StringComparison.OrdinalIgnoreCase)))

{

Console.WriteLine($"Товар с названием '{product.Name}' уже существует на складе. Пожалуйста, выберите другое название.");

return;

}

products.Add(product);

Console.WriteLine($"Товар '{product.Name}' добавлен на склад.");

}

public void RemoveProduct(string name)

{

var productToRemove = products.Find(p => p.Name.Equals(name, StringComparison.OrdinalIgnoreCase));

if (productToRemove != null)

{

products.Remove(productToRemove);

Console.WriteLine($"Товар '{name}' удален со склада.");

}

else

{

Console.WriteLine($"Товар '{name}' не найден на складе.");

}

}

public List<IProduct> FindProductsByName(string name)

{

return products.FindAll(p => p.Name.Equals(name, StringComparison.OrdinalIgnoreCase));

}

public List<IProduct> FindProductsByCategory(string category)

{

return products.FindAll(p => p.Category.Equals(category, StringComparison.OrdinalIgnoreCase));

}

public List<IProduct> FindProductsByQuantity(int quantity)

{

return products.FindAll(p => p.Quantity == quantity);

}

public List<IProduct> GetAllProducts()

{

return products;

}

}

class Program

{

static void Main(string[] args)

{

IWarehouse warehouse = new Warehouse();

while (true)

{

Console.WriteLine("\nВыберите действие:");

Console.WriteLine("1. Добавить товар");

Console.WriteLine("2. Удалить товар");

Console.WriteLine("3. Отобразить товары");

Console.WriteLine("4. Поиск товаров");

Console.WriteLine("5. Выход");

Console.Write("Введите номер действия: ");

int choice;

if (!int.TryParse(Console.ReadLine(), out choice))

{

Console.WriteLine("Ошибка ввода. Пожалуйста, введите номер.");

continue;

}

switch (choice)

{

case 1:

AddProduct(warehouse);

break;

case 2:

RemoveProduct(warehouse);

break;

case 3:

DisplayProducts(warehouse.GetAllProducts());

break;

case 4:

SearchProducts(warehouse);

break;

case 5:

Console.WriteLine("Выход...");

return;

default:

Console.WriteLine("Неверный выбор. Пожалуйста, выберите действие из списка.");

break;

}

}

}

static void AddProduct(IWarehouse warehouse)

{

Console.Write("Введите название товара: ");

string name = Console.ReadLine();

if (warehouse.GetAllProducts().Exists(p => p.Name.Equals(name, StringComparison.OrdinalIgnoreCase)))

{

Console.WriteLine($"Товар с названием '{name}' уже существует на складе. Пожалуйста, выберите другое название.");

return;

}

Console.Write("Введите категорию товара: ");

string category = Console.ReadLine();

Console.Write("Введите количество товара: ");

int quantity;

while (!int.TryParse(Console.ReadLine(), out quantity))

{

Console.WriteLine("Ошибка ввода. Пожалуйста, введите корректное количество.");

Console.Write("Введите количество товара: ");

}

warehouse.AddProduct(new Product(name, category, quantity));

}

static void RemoveProduct(IWarehouse warehouse)

{

Console.Write("Введите название товара для удаления: ");

string name = Console.ReadLine();

warehouse.RemoveProduct(name);

}

static void SearchProducts(IWarehouse warehouse)

{

Console.WriteLine("\nПоиск товаров по:");

Console.WriteLine("1. Названию");

Console.WriteLine("2. Категории");

Console.WriteLine("3. Количеству");

Console.Write("Введите номер действия: ");

int choice;

if (!int.TryParse(Console.ReadLine(), out choice))

{

Console.WriteLine("Ошибка ввода. Пожалуйста, введите номер.");

return;

}

switch (choice)

{

case 1:

Console.Write("Введите название товара: ");

string name = Console.ReadLine();

DisplayProducts(warehouse.FindProductsByName(name));

break;

case 2:

Console.Write("Введите категорию товара: ");

string category = Console.ReadLine();

DisplayProducts(warehouse.FindProductsByCategory(category));

break;

case 3:

Console.Write("Введите количество товара: ");

int quantity;

while (!int.TryParse(Console.ReadLine(), out quantity))

{

Console.WriteLine("Ошибка ввода. Пожалуйста, введите корректное количество.");

Console.Write("Введите количество товара: ");

}

DisplayProducts(warehouse.FindProductsByQuantity(quantity));

break;

default:

Console.WriteLine("Неверный выбор. Пожалуйста, выберите действие из списка.");

break;

}

}

static void DisplayProducts(IEnumerable<IProduct> products)

{

Console.WriteLine("\nТовары на складе:");

foreach (var product in products)

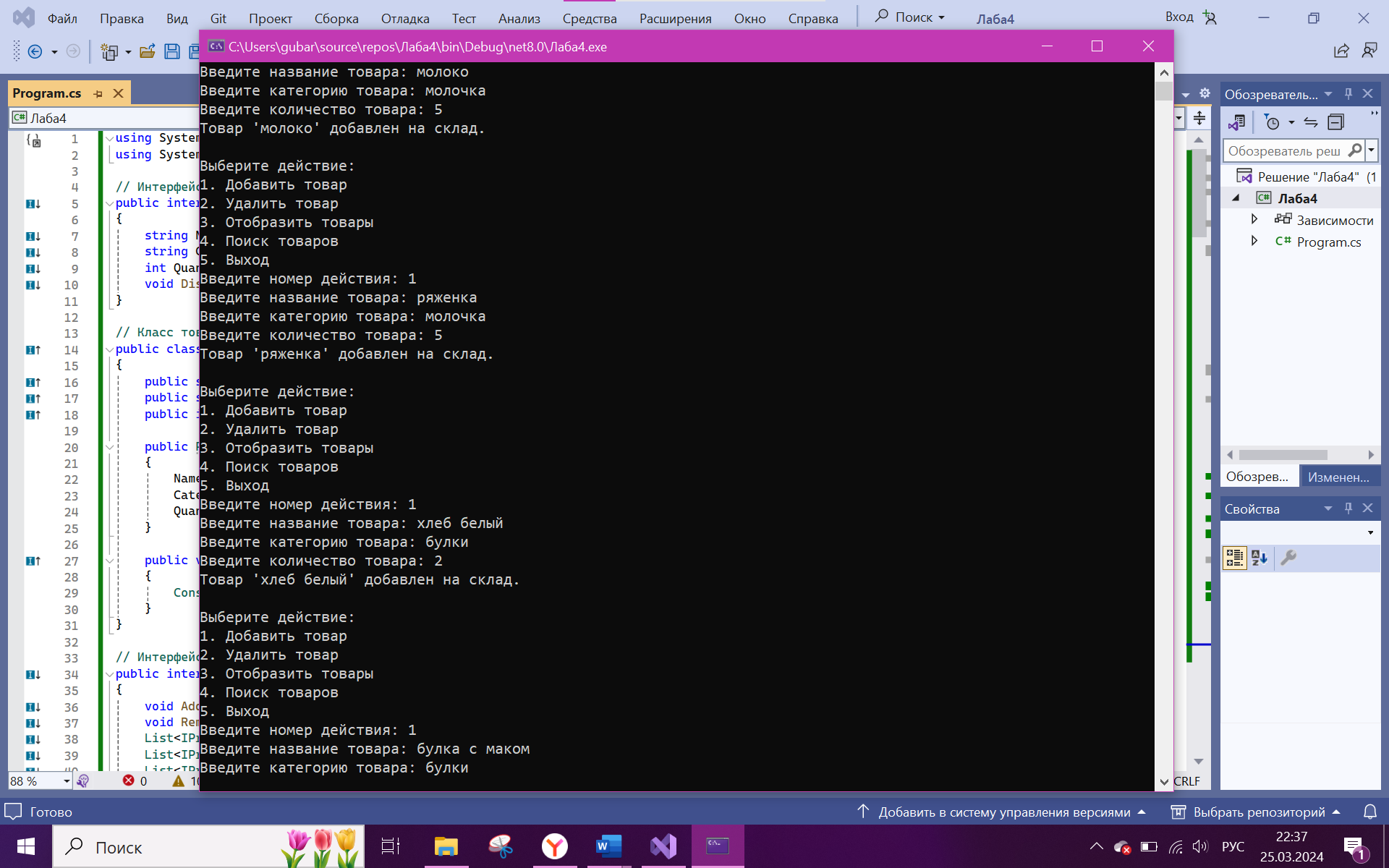
{

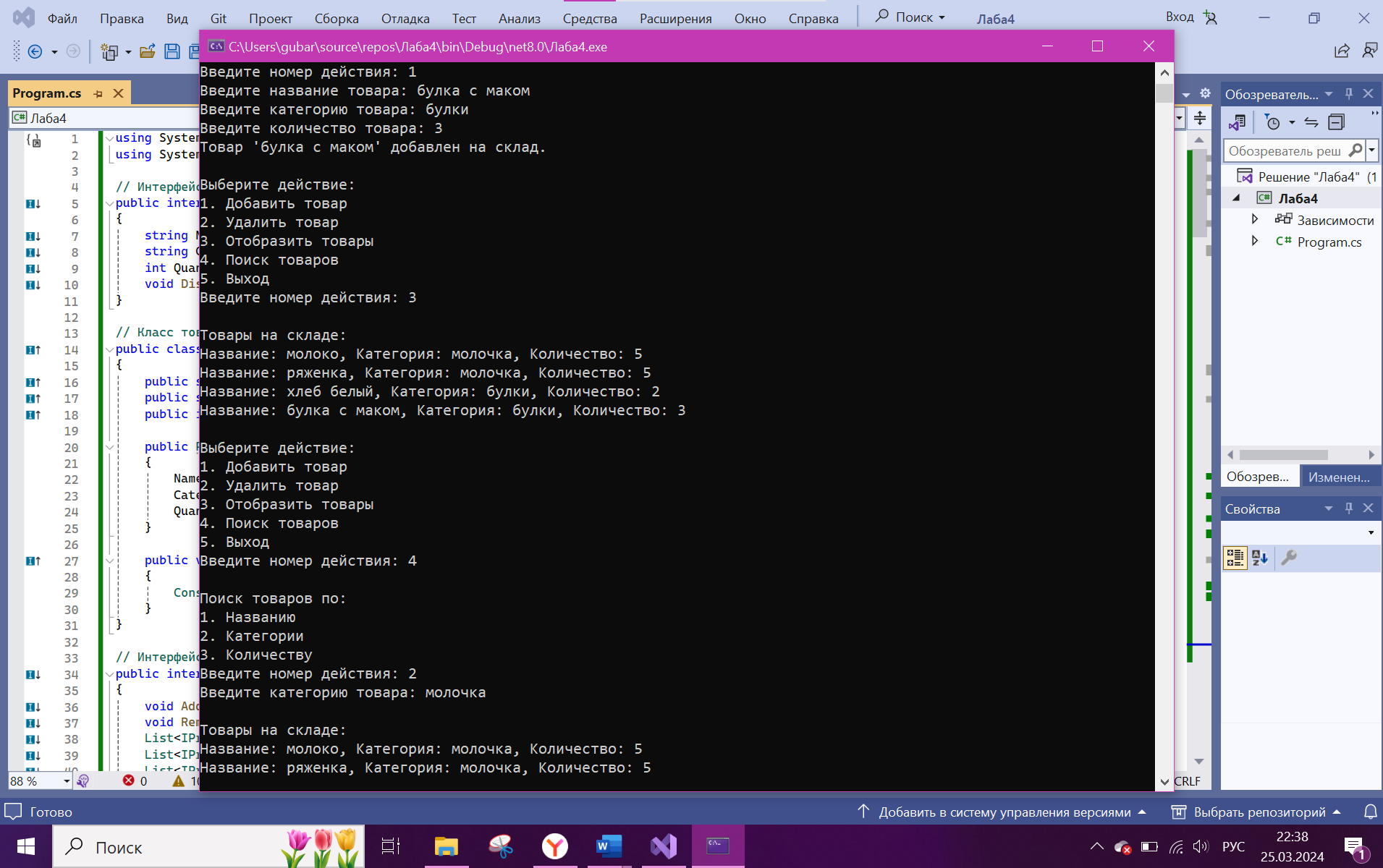
product.DisplayInfo();

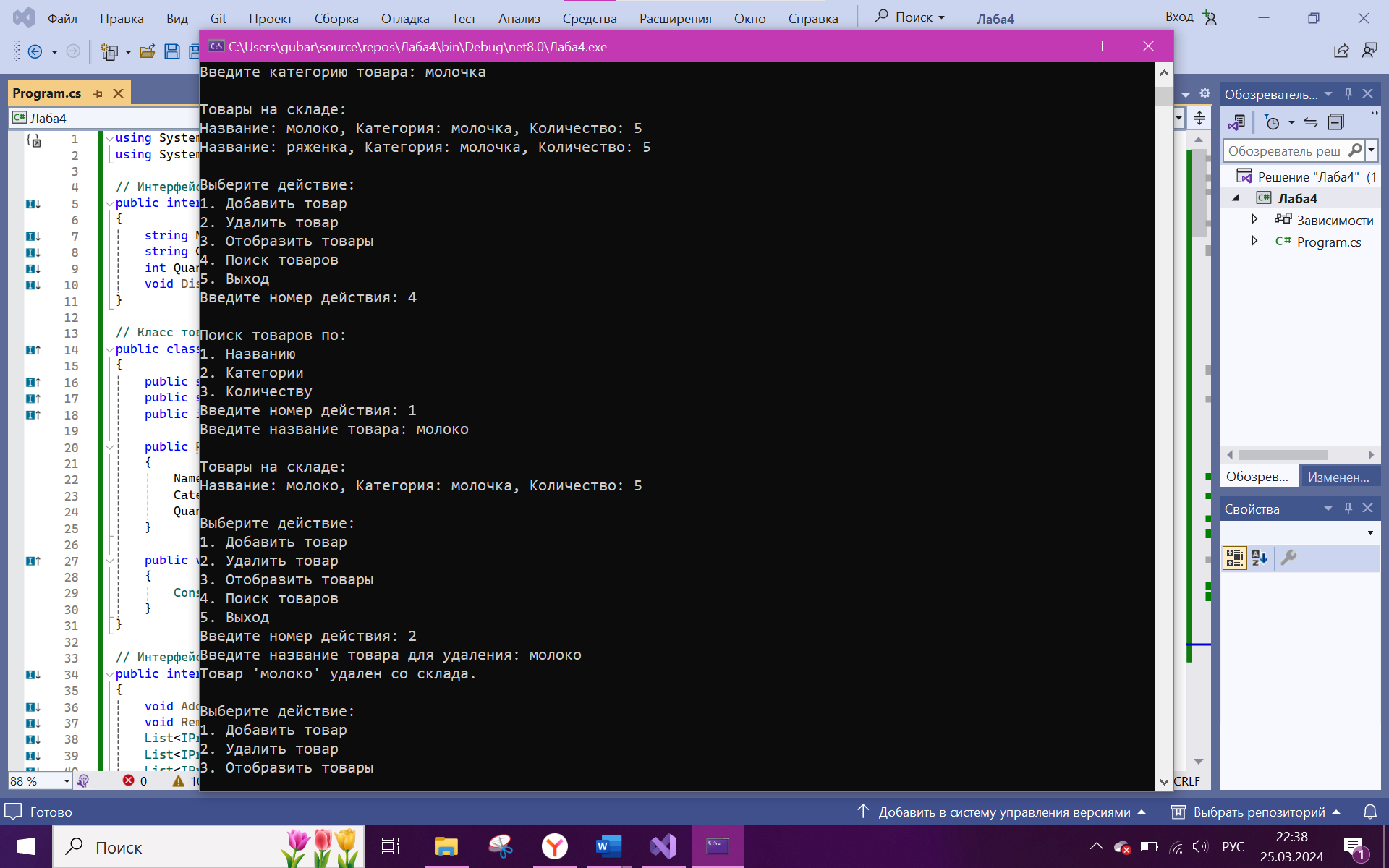
}

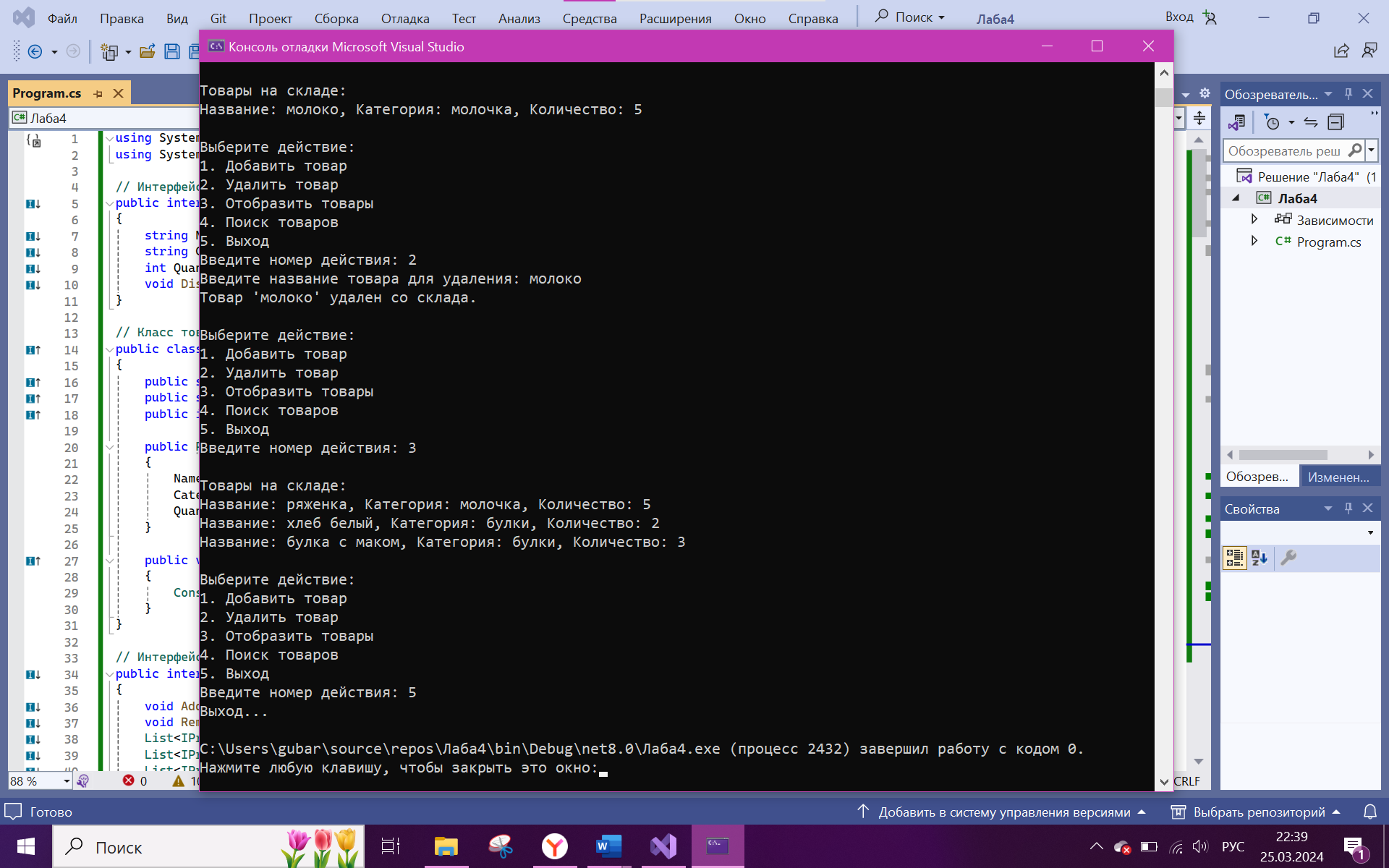
}

}

****

****

****

****