**Абабий Илья**

**ПРАКТИЧЕСКАЯ РАБОТА №7**

**Тема:Структуры в С#**

Цель работы: научиться разрабатывать структуры, их основные компоненты: поля, методы, конструкторы, создавать объекты класса, обращаться к открытым полям и методам структуры.

1 вариант: Разработать структуру склад. Определить 3 поля и 2 метода. Создать 3 объекта, записать в лист и организовать возможности явного и неявного объявления.

Ход работы:

static void Main(string[] args)

{

///явное

// Создание 3 объектов:

Warehouse warehouse1 = new Warehouse()

{

Name = "Product 1",

Quantity = 10,

Price = 100

};

Warehouse warehouse2 = new Warehouse()

{

Name = "Product 2",

Quantity = 20,

Price = 200

};

Warehouse warehouse3 = new Warehouse()

{

Name = "Product 3",

Quantity = 30,

Price = 300

};

// Добавление объектов в лист:

List<Warehouse> warehouses = new List<Warehouse>() { warehouse1, warehouse2, warehouse3 };

// Работа с объектами:

Console.WriteLine($"Склад 1: {warehouse1.Name}, количество: {warehouse1.Quantity}, цена: {warehouse1.Price}");

warehouse1.Add(5);

Console.WriteLine($"Склад 1 после добавления: количество: {warehouse1.Quantity}");

warehouse1.Remove(12);

Console.WriteLine($"Склад 1 после удаления: количество: {warehouse1.Quantity}");

///неявное

var warehouse4 = new Warehouse

{

Name = "Product 4",

Quantity = 40,

Price = 400

};

// Добавление объекта в лист:

warehouses.Add(warehouse4);

// Работа с объектом:

Console.WriteLine($"Склад 4: {warehouse4.Name}, количество: {warehouse4.Quantity}, цена: {warehouse4.Price}");

Console.ReadKey();

}

public struct Warehouse

{

// 3 поля:

public string Name; // Наименование

public int Quantity; // Количество

public decimal Price; // Цена

// 2 метода:

public void Add(int quantity) // Добавить

{

Quantity += quantity;

}

public void Remove(int quantity) // Удалить

{

if (Quantity >= quantity)

{

Quantity -= quantity;

}

else

{

throw new Exception("Недостаточно товара на складе!");

}

}

}

Вывод:Я научился разрабатывать структуры, их основные компоненты: поля, методы, конструкторы, создавать объекты класса, обращаться к открытым полям и методам структуры.