Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра Информатики

Дисциплина «Программирование»

**ОТЧЕТ**

к лабораторной работе №6

на тему:

**«наследование»**

БГУИР 6-05-0612-02 35

|  |
| --- |
| Выполнил студент группы 353504  ГУСЕНЦОВА Екатерина Андреевна |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

**Задание 1. Вариант 15.** Предметная область: Фрукты или овощи . Выделить в предметной области 2-3 варианта сущности, отличающиеся несколькими полями и методами. Каждый класс имеет поля, свойства и методы. Спроектировать UML-диаграммы классов. Продемонстрировать работу всех объявленных методов.

# 2 Выполнение работы

В рамках заданной предметной области программа содержит несколько классов: родительский класс Vegetables\_fruits, производные классы Banana и Mango, и производный класс Orange, закрытый для наследования. Диаграмма данных классов представлена на рисунке 1.

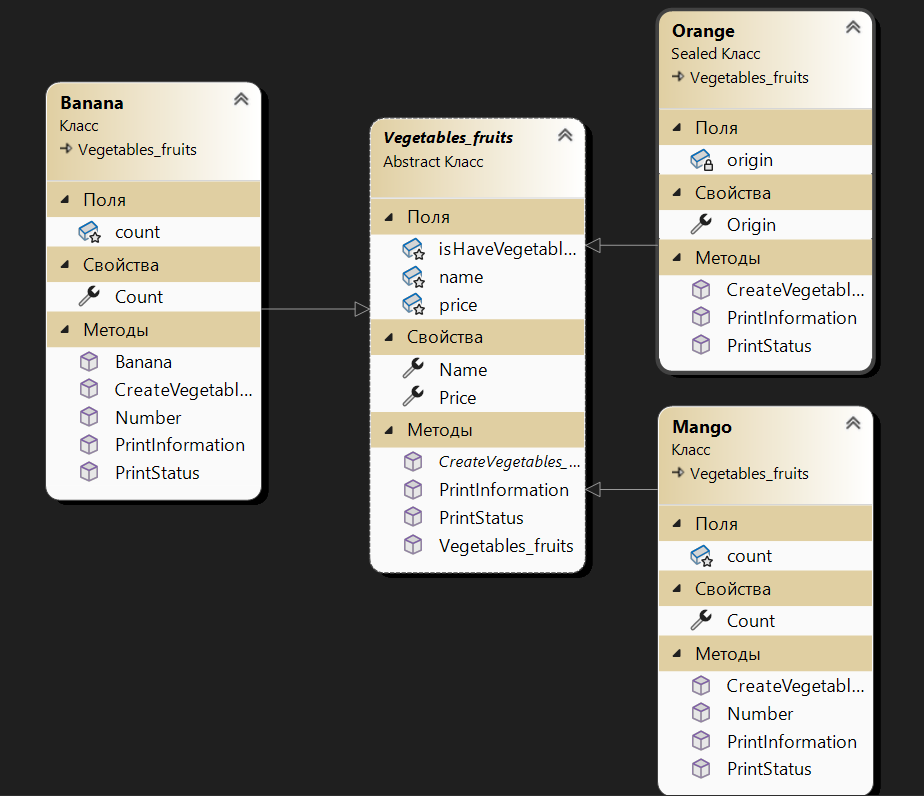


Рисунок 1 – Диаграмма классов

На диаграмме представлен класс Vegetables\_fruits, который является базовым абстрактным классом. Классы Banana и Mango наследуются от базового класса. Класс Orange объявляется с использованием модификатора sealed, что означает, что класс закрыт для наследования и не может использоваться в качестве базового.

Класс Vegetables\_fruits имеет защищенные поля, которые хранят наименование, цену, а также поле типа bool, которое отвечает за наличие созданного пользователем фрукта или овоща. Поля name и price имеют свойства для получения и изменения их значений. В классе содержится абстрактный метод для создания фрукта или овоща, публичный метод с реализацией для вывода информации о продукте, виртуальный метод, который выводит информацию о готовности продукта и конструктор класса. Ниже представлен листинг кода файла Vegetables\_fruits.cs.

using System;

namespace lab6

{

internal abstract class Vegetables\_fruits

{

protected string name;

protected double price;

protected bool isHaveVegetables\_fruits = false;

public string Name

{

get { return name; }

set { name = value; }

}

public double Price

{

get { return price; }

set { price = value; }

}

public abstract void CreateVegetables\_fruits(); // Abstract method

public virtual void PrintStatus()

{

Console.WriteLine("Товары можно забрать");

}

public virtual void PrintInformation()

{

Console.WriteLine("Товары не выбраны!");

}

public Vegetables\_fruits()

{

Console.WriteLine("\n\nКонструктор родительского класса вызван.\n");

}

}

}

Класс Banana содержит поле, которое хранит количество бананов, и соответствующее свойство для получения и изменения значения данного поля. В классе переопределяется метод PrintStatus() с помощью ключевого слова override для вывода сообщения о готовности продукта. Имеется метод Number(), который выводит информацию о количестве бананов. Другой публичный метод переопределяется от абстрактного метода и предназначен для создания бананов. Также имеется метод PrintInformation(), объявленный с помощью ключевого слова new, который скрывает базовый метод. Описан конструктор класса Banana для демонстрации вызова конструктора родительского класса при наследовании. Ниже представлен листинг кода файла Banana.cs.

using System;

namespace lab6

{

internal class Banana : Vegetables\_fruits

{

protected int count;

public int Count

{

get { return count; }

set { count = value; }

}

public override void PrintStatus()

{

Console.WriteLine("Бананы можно забрать");

}

public void Number()

{

Console.WriteLine("Количество бананов: " + count);

}

public override void CreateVegetables\_fruits()

{

isHaveVegetables\_fruits = true;

Console.Write("Введите страну производитель: ");

Name = Console.ReadLine();

Console.Write("Введите цену бананов: ");

Price = CheckingForInput.Double(Console.ReadLine());

Console.Write("Введите количество: ");

Count = CheckingForInput.Int(Console.ReadLine());

PrintStatus();

}

public override void PrintInformation()

{

if (isHaveVegetables\_fruits)

{

Console.WriteLine("Бананы (страна + цена): " + Name + ", " + Price + ". ");

Number();

}

else

{

base.PrintInformation();

}

}

public Banana()

{

Console.WriteLine("Конструктор производного класса Banana вызван!\n\n");

}

}

}

Класс Mango содержит поле, которое хранит количество манго, и соответствующее свойство для получения и изменения значения данного поля. В классе переопределяется метод PrintStatus() с помощью ключевого слова override для вывода сообщения о готовности продукта. Имеется метод Number(), который выводит информацию о количестве манго. Другой публичный метод переопределяется от абстрактного метода и предназначен для создания манго. Также имеется метод PrintInformation(), объявленный с помощью ключевого слова new, который скрывает базовый метод. Ниже представлен листинг кода файла Mango.cs.

using System;

namespace lab6

{

internal class Mango : Vegetables\_fruits

{

protected int count;

public int Count

{

get { return count; }

set { count = value; }

}

public override void PrintStatus()

{

Console.WriteLine("Манго можно забрать");

}

public void Number()

{

Console.WriteLine("Количество: " + count);

}

public override void CreateVegetables\_fruits()

{

isHaveVegetables\_fruits = true;

Console.Write("Введите страну производитель манго: ");

Name = Console.ReadLine();

Console.Write("Введите цену манго: ");

Price = CheckingForInput.Double(Console.ReadLine());

Console.Write("Введите количество: ");

Count = CheckingForInput.Int(Console.ReadLine());

PrintStatus();

}

public override void PrintInformation()

{

if (isHaveVegetables\_fruits)

{

Console.WriteLine("Манго (страна + цена): " + Name + ", " + Price + ". ");

Number();

}

else

{

base.PrintInformation();

}

}

}

}

Класс Orange содержит поле, которое хранит происхождение апельсинов, и соответствующее свойство для получения и изменения значения данного поля. В классе переопределяется метод PrintStatus() с помощью ключевого слова override для вывода сообщения о готовности продукта. Также имеется метод PrintInformation(), объявленный с помощью ключевого слова new, который скрывает базовый метод. Ниже представлен листинг кода файла Orange.cs.

using System;

namespace lab6

{

internal sealed class Orange : Vegetables\_fruits

{

private string origin;

public string Origin

{

get { return origin; }

set { origin = value; }

}

public override void PrintStatus()

{

Console.WriteLine("Апельсины можно забрать");

}

public override void PrintInformation()

{

if (isHaveVegetables\_fruits)

{

Console.WriteLine("Апельсины (страна + цена): " + Name + ", " + Price + ". Происхождение: " + origin);

}

else

{

base.PrintInformation();

}

}

public override void CreateVegetables\_fruits()

{

isHaveVegetables\_fruits = true;

Console.Write("Введите страну производитель апельсинов: ");

Name = Console.ReadLine();

Console.Write("Введите цену: ");

Price = CheckingForInput.Double(Console.ReadLine());

Console.Write("Введите вид апельсинов: ");

Origin = Console.ReadLine();

PrintStatus();

}

}

}

В главной функции у пользователя запрашиваются нужные данные для реализации программы. С помощью оператора многозадачного выбора switch-case производим демонстрацию работы всех функций программы. Ниже представлен листинг кода файла Program.cs.

using System;

namespace lab6

{

internal class Program

{

static void Main(string[] args)

{

Banana banana = new Banana();

Orange orange = new Orange();

Mango mango = new Mango();

Console.Clear();

Menu menuLab = new Menu("6", "1", "15", "Изучить механизмы наследования в C#. Индивидуальное задание: овощи и фрукты", "Гусенцова Екатерина", "353504");

menuLab.ShowInfo();

Console.WriteLine("Добро пожаловать в магазин овощей и фруктов!\n");

bool ans = true;

while (ans)

{

Menu.MainMenu();

Console.Write("\nВыбор: ");

int choice = CheckingForInput.Choice(Console.ReadLine());

switch (choice)

{

case 1:

banana.CreateVegetables\_fruits();

break;

case 2:

mango.CreateVegetables\_fruits();

break;

case 3:

orange.CreateVegetables\_fruits();

break;

case 4:

banana.PrintInformation();

break;

case 5:

mango.PrintInformation();

break;

case 6:

orange.PrintInformation();

break;

case 7:

Banana newBanana = new Banana();

newBanana.CreateVegetables\_fruits();

break;

case 8:

ans = false;

break;

default:

Console.WriteLine("Неверный выбор. Пожалуйста, попробуйте снова.");

break;

}

}

}

}

}

В программе также имеются классы для проверки на ввод входных данных CheckingForInput и класс, хранящий меню для описания заданий из лабораторной работы и главное меню для демонстрации работы функций программы.

Результат выполнения программы представлен на рисунке 2.

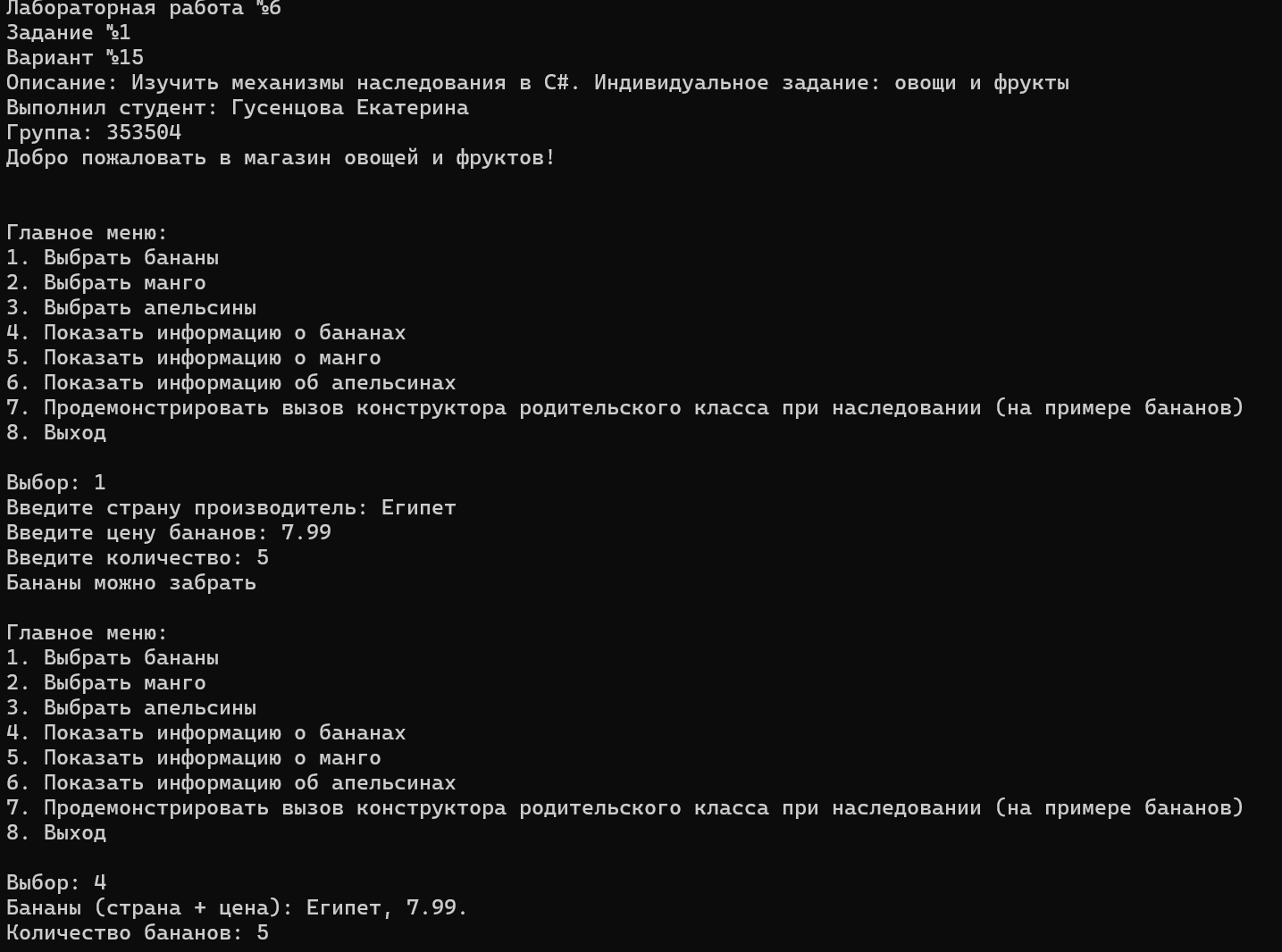


Рисунок 2 – Результат выполнения программы

# Вывод

В ходе лабораторной работы были изучены основные принципы, связанные с использованием наследования классов, были изучены возможности создания иерархии классов с общими свойствами и методами, освоены все возможности классов с модификаторами abstract и sealed, а также приобретены навыки использования абстрактных и виртуальных методов.