МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники (ФИТР)

Кафедра программного обеспечения информационных систем и технологий

**О Т Ч Ё Т**

**ПО ПРОДЕЛАННОЙ РАБОТЕ**

**по курсу «Конструирование программного обеспечения»**

|  |  |
| --- | --- |
| Выполнили студенты гр. 10701221: | Шишкунова А. Д.  Прокофьева А. В. |
|  |  |
| Проверил: | Станкевич С.Н. |

Минск, 2023

**Задание:**

**1 вариант. Цветочница или магазин цветов (Flower Shop).** В магазине цветов можно собрать букет из соответствующих цветов. Необходимо определить вес букета, его стоимость и самый дорогой/недорогой цветок (или цветы, если они одинаковы по стоимости).

**UML диаграмма классов:**

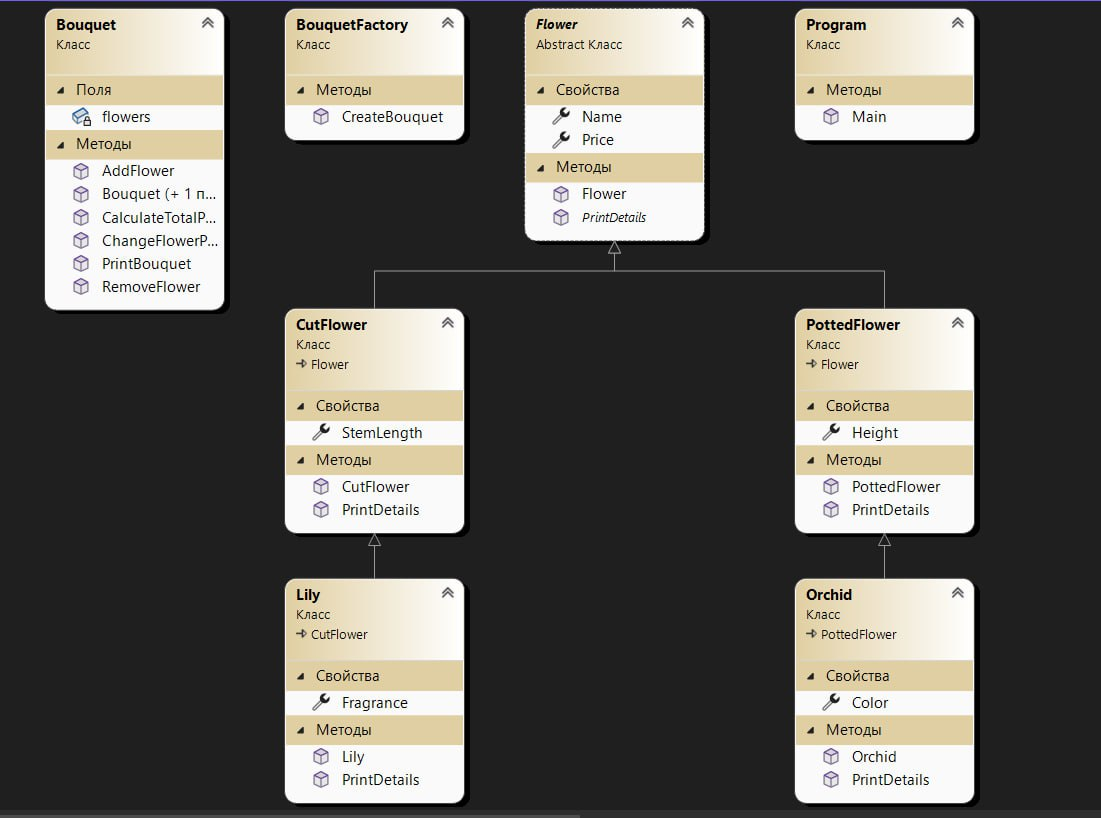


Рис. 1 – Диаграмма классов, созданная в среде Visual Studio

**Выполнение программы:**

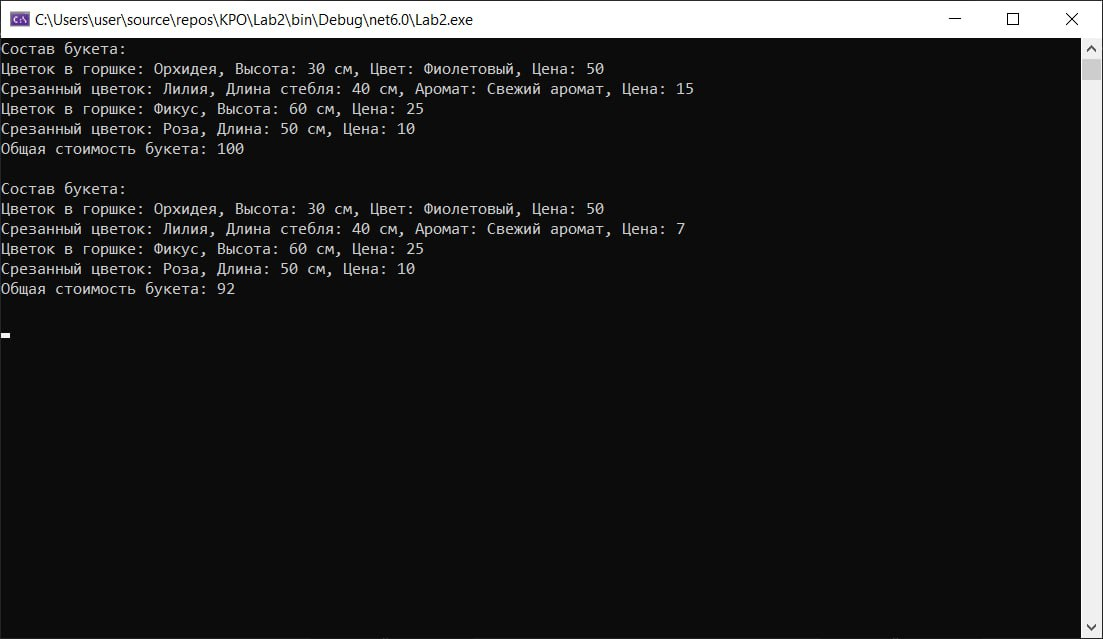


Рис. 2 – Результат выполнения программы

**UNIT-тестирование:**

using KPOLab2;

using Lab2;

using System.Diagnostics;

using System.Xml.Linq;

namespace FlowerShopTest

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void AddFlower\_ShouldIncreaseFlowerCount()

{

// Arrange

Bouquet bouquet = new Bouquet();

CutFlower flower = new CutFlower("Rose", 10.0, 40);

// Act

bouquet.AddFlower(flower);

// Assert

Assert.AreEqual(1, bouquet.Flowers.Count);

}

[TestMethod]

public void RemoveFlower\_ShouldDecreaseFlowerCount()

{

// Arrange

Flower flower1 = new CutFlower("Rose", 10.0, 32);

Flower flower2 = new CutFlower("Tulip", 5.0, 50);

List<Flower> initialFlowers = new List<Flower> { flower1, flower2 };

Bouquet bouquet = new Bouquet(initialFlowers);

// Act

bouquet.RemoveFlower(flower1);

// Assert

Assert.AreEqual(1, bouquet.Flowers.Count);

}

[TestMethod]

public void Flower\_Initialization\_Success()

{

// Arrange

string name = "Rose";

double price = 10.99;

// Act

Flower flower = new CutFlower(name, price, 44);

// Assert

Assert.AreEqual(name, flower.Name);

Assert.AreEqual(price, flower.Price);

}

[TestMethod]

public void Bouquet\_CalculateTotalPrice()

{

// Arrange

Flower flower1 = new CutFlower("Rose", 10.0, 32);

Flower flower2 = new CutFlower("Tulip", 5.0, 50);

Bouquet bouquet = new Bouquet();

bouquet.AddFlower(flower1);

bouquet.AddFlower(flower2);

double totalPrice = 15.0;

// Act

double currentPrice = bouquet.CalculateTotalPrice();

// Assert

Assert.AreEqual(totalPrice, currentPrice);

}

[TestMethod]

public void Boquet\_ChangeFlowerPrice()

{

//Arrange

string name = "Rose";

double price = 10.0;

double changedPrice = 15.0;

Flower flower1 = new CutFlower(name, price, 32);

Bouquet bouquet = new Bouquet();

bouquet.AddFlower(flower1);

//Act

bouquet.ChangeFlowerPrice(name, changedPrice);

// Assert

Assert.AreEqual(changedPrice, flower1.Price);

}

[TestMethod]

public void CreateBouquet\_Test()

{

// Arrange

List<Flower> initialFlowers = new List<Flower>

{

new CutFlower("Rose", 15.0, 40),

new CutFlower("Tulip", 25.0, 30),

new CutFlower("Lily", 35.0, 40)

};

// Act

Bouquet bouquet = BouquetFactory.CreateBouquet(initialFlowers);

// Assert

Assert.AreEqual(initialFlowers.Count, bouquet.Flowers.Count);

CollectionAssert.AreEqual(initialFlowers, bouquet.Flowers);

}

//[TestMethod]

//public void PrintDetails\_Test()

//{

// // Arrange

// List<Flower> expectedFlowers = new List<Flower>

// {

// new CutFlower("Rose", 15.1, 40),

// new CutFlower("Tulip", 25.1, 30),

// new CutFlower("Lily", 35.0, 40)

// };

// Bouquet bouquet = new Bouquet(expectedFlowers);

// using (var sw = new StringWriter())

// {

// Console.SetOut(sw);

// // Act

// bouquet.PrintBouquet();

// var output = sw.ToString().Trim();

// //Assert

// var expectedOutput = $"Состав букета:\n"+

// "Срезанный цветок: Rose, Длина: 40 см, Цена: 15,1\n"+

// "Срезанный цветок: Tulip, Длина: 30 см, Цена: 25,1\n"+

// "Срезанный цветок: Lily, Длина: 40 см, Цена: 35,0";

// Assert.AreEqual(expectedOutput, output);

// }

//}

}

}

**Листинг программы всех лабораторных работ:**

*Program.cs*

public class Program

{

public static void Main()

{

List<Flower> initialFlowers = new List<Flower>

{

new Orchid("Орхидея", 50.0, 30, "Фиолетовый"),

new Lily("Лилия", 15.0, 40, "Свежий аромат"),

new PottedFlower("Фикус", 25.0, 60),

new CutFlower("Роза", 10.0, 50)

};

Bouquet bouquet = BouquetFactory.CreateBouquet(initialFlowers);

bouquet.PrintBouquet();

Console.WriteLine("Общая стоимость букета: " + bouquet.CalculateTotalPrice() + "\n");

bouquet.ChangeFlowerPrice("Лилия", 7.0);

bouquet.PrintBouquet();

Console.WriteLine("Общая стоимость букета: " + bouquet.CalculateTotalPrice() + "\n");

Console.ReadLine();

}

}

*Flower.cs*

public abstract class Flower

{

public string Name { get; set; }

public double Price { get; set; }

public Flower(string name, double price)

{

Name = name;

Price = price;

}

public abstract void PrintDetails();

}

*Bouquet.cs*

public class Bouquet

{

private List<Flower> flowers;

public Bouquet()

{

flowers = new List<Flower>();

}

public Bouquet(List<Flower> initialFlowers)

{

flowers = new List<Flower>(initialFlowers);

}

public void AddFlower(Flower flower)

{

flowers.Add(flower);

}

public void RemoveFlower(Flower flower)

{

flowers.Remove(flower);

}

public void PrintBouquet()

{

Console.WriteLine("Состав букета:");

foreach (Flower flower in flowers)

{

flower.PrintDetails();

}

}

public double CalculateTotalPrice()

{

double totalPrice = 0.0;

foreach (Flower flower in flowers)

{

totalPrice += flower.Price;

}

return totalPrice;

}

public void ChangeFlowerPrice(string flowerName, double newPrice)

{

foreach (Flower flower in flowers)

{

if (flower.Name == flowerName)

{

flower.Price = newPrice;

break;

}

}

}

*BouquetFactory.cs*

public class BouquetFactory

{

public static Bouquet CreateBouquet(List<Flower> initialFlowers)

{

Bouquet bouquet = new Bouquet();

foreach (Flower flower in initialFlowers)

{

bouquet.AddFlower(flower);

}

return bouquet;

}

}

*PottedFlower.cs*

public class PottedFlower : Flower

{

public int Height { get; set; }

public PottedFlower(string name, double price, int height) : base(name, price)

{

Height = height;

}

public override void PrintDetails()

{

Console.WriteLine($"Цветок в горшке: {Name}, Высота: {Height} см, Цена: {Price}");

}

}

public class Orchid : PottedFlower

{

public string Color { get; set; }

public Orchid(string name, double price, int height, string color) : base(name, price, height)

{

Color = color;

}

public override void PrintDetails()

{

Console.WriteLine($"Цветок в горшке: {Name}, Высота: {Height} см, Цвет: {Color}, Цена: {Price}");

}

}

*CutFlower.cs*

public class CutFlower : Flower

{

public int StemLength { get; set; }

public CutFlower(string name, double price, int stemLength) : base(name, price)

{

StemLength = stemLength;

}

public override void PrintDetails()

{

Console.WriteLine($"Срезанный цветок: {Name}, Длина: {StemLength} см, Цена: {Price}");

}

}

public class Lily : CutFlower

{

public string Fragrance { get; set; }

public Lily(string name, double price, int stemLength, string fragrance) : base(name, price, stemLength)

{

Fragrance = fragrance;

}

public override void PrintDetails()

{

Console.WriteLine($"Срезанный цветок: {Name}, Длина стебля: {StemLength} см, Аромат: {Fragrance}, Цена: {Price}");

}

}