

11 Полиморфизм

Задание 1. Составить программу с одним родительским классом и двумя потомками. Потомки должны содержать виртуальные функции. Создать виртуальную функцию выдачи результатов расчета методов на экран монитора с указанием названий и полей, и их значений соответствующего объекта. Составить тестирующую программу с выдачей протокола на экран монитора. При этом создать объекты базового и производных типов, используя полиморфный контейнер - массив ссылок базового класса на объекты базового и производных классов (количество объектов ≥ 5). Родительский класс – Мебель(поле – название), потомки – Шкаф(поле объём V в см), Диван(поле площадь S в см). Методы стоимость мебели: Шкаф($C = \sqrt{V} * 0,75$), Диван($C = \frac{S^2}{3} + 5000$).

Листинг программы:

```
namespace Space
{
    class Furniture
    {
        public string Name { get; set; }

        public virtual double GetCost()
        {
            return 0;
        }

        public virtual void PrintInfo()
        {
            Console.WriteLine();
            Console.WriteLine($"Название: {Name}");
        }
    }

    class Wardrobe : Furniture
    {
        public double Volume { get; set; }

        public override double GetCost()
        {
```

					УП 2-40 01 01.37ТП.227.23.11							
Изм.	Лист	№ док	Подпись	Дата	Полиморфизм					Лит	Лист	Листов
Разраб.		Сорокина Е.А.										
Проверил.		Новик А.И.									51	
										Гродненский ГКТТид		
Н.контр.												
Утвердил.												

```

        return Math.Sqrt(Volume) * 0.75;
    }

    public override void PrintInfo()
    {
        Console.WriteLine();
        Console.WriteLine($"Название: {Name}\nОбъём: {Volume}
см^3\nЦена: {Math.Round(GetCost())} руб.");
    }
}

class Sofa : Furniture
{
    public double Area { get; set; }

    public override double GetCost()
    {
        return (Math.Pow(Area, 2) / 3) + 5000;
    }

    public override void PrintInfo()
    {
        Console.WriteLine();
        Console.WriteLine($"Название: {Name}\nПлощадь: {Area}
см^2\nЦена: {Math.Round(GetCost())} руб.");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Furniture[] furnitureArray = new Furniture[5];
        furnitureArray[0] = new Furniture { Name = "Стол Крик" };
        furnitureArray[1] = new Wardrobe { Name = "Шкаф Нарния", Volume =
5000 };
        furnitureArray[2] = new Sofa { Name = "Диван Победа", Area = 3000 };
        furnitureArray[3] = new Wardrobe { Name = "Шкаф Крыша", Volume =
7000 };
        furnitureArray[4] = new Sofa { Name = "Диван Ландыши", Area = 4000
};

        Console.WriteLine("Мебельный каталог:");
        foreach (Furniture f in furnitureArray)

```

```

        {
            f.PrintInfo();
        }

        Console.ReadKey();
    }
}

```

Таблица 11.1 – Входные и выходные данные

Входные данные	Выходные данные
"Стол Крик"	Стол крик
"Шкаф Нарния", 5000	53
"Диван Победа", 3000	3005000
"Шкаф Крыша", 7000	63
"Диван Ландыши", 4000	5338333

Анализ результатов:

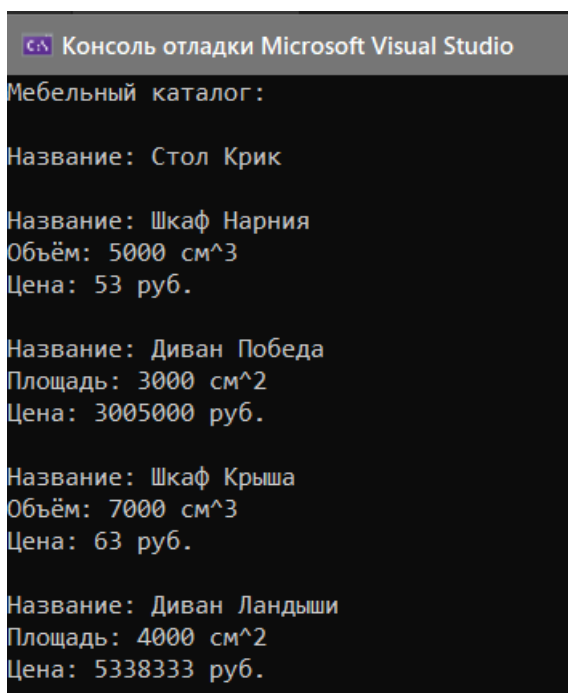


Рисунок 11.1 – Результат работы программы

Задание 2. Составить программу с абстрактным родительским классом и двумя объектами - потомками. Составить тестирующую программу с выдачей протокола на экран монитора. В ней нужно реализовать циклический вывод параметров объектов, используя полиморфный контейнер - массив объектов базового класса (количество объектов ≥ 5). Организовать вычисление суммарной длины векторов.

Листинг программы:

```
namespace Space
```

```
{
```

```
    public abstract class Vector
```

```
    {
```

```
        public abstract void Print();
```

```
        public abstract double Length();
```

```
    }
```

```
    public class Vector2D : Vector
```

```
    {
```

```
        private double x;
```

```
        private double y;
```

```
        public Vector2D(double x, double y)
```

```
        {
```

```
            this.x = x;
```

```
            this.y = y;
```

```
        }
```

```
        public override void Print()
```

```
        {
```

```
            Console.WriteLine("Двумерный вектор: ({0}, {1})", x, y);
```

```
        }
```

```
        public override double Length()
```

```
        {
```

```
            return Math.Sqrt(x * x + y * y);
```

```
        }
```

```
    }
```

```
    public class Vector3D : Vector
```

```
    {
```

```
        private double x;
```

```
        private double y;
```

```
        private double z;
```

```
        public Vector3D(double x, double y, double z)
```

```
        {
```

```
            this.x = x;
```

```
            this.y = y;
```

```
            this.z = z;
```

```
        }
```

Изм.	Лист	№ док	Подпись	Дата

```

    public override void Print()
    {
        Console.WriteLine("Трёхмерный вектор: ({0}, {1}, {2})", x, y, z);
    }

    public override double Length()
    {
        return Math.Sqrt(x * x + y * y + z * z);
    }
}

class Program
{
    static void Main(string[] args)
    {
        List<Vector> vectors = new List<Vector>();

        vectors.Add(new Vector2D(1, 2));
        vectors.Add(new Vector3D(3, 4, 5));
        vectors.Add(new Vector2D(-1, 3));
        vectors.Add(new Vector3D(-2, 6, -3));
        vectors.Add(new Vector2D(4, -2));

        double totalLength = 0;

        foreach (Vector v in vectors)
        {
            v.Print();
            totalLength += v.Length();
        }

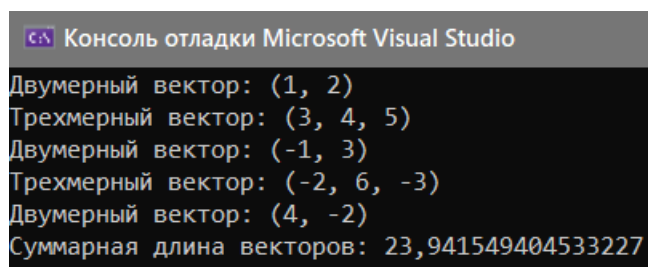
        Console.WriteLine("Суммарная длина векторов: {0}", totalLength);
    }
}

```

Таблица 11.2 – Входные и выходные данные

Входные данные	Выходные данные
(1, 2) (3, 4, 5) (-1, 3) (-2, 6, -3) (4, -2)	23,941549404533227

Анализ результатов:



```

Консоль отладки Microsoft Visual Studio
Двумерный вектор: (1, 2)
Трёхмерный вектор: (3, 4, 5)
Двумерный вектор: (-1, 3)
Трёхмерный вектор: (-2, 6, -3)
Двумерный вектор: (4, -2)
Суммарная длина векторов: 23,941549404533227
    
```

Рисунок 11.2 – Результат работы программы