**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).

class Program

{

static void Main(string[] args)

{

Vector[] vectors = new Vector[5];

// заполняем массив объектами базового и производных классов

vectors[0] = new Vector("Вектор 1");

vectors[1] = new TwoDimensionalVector("(первый)", 3.0, 4.0);

vectors[2] = new ThreeDimensionalVector("(первый)", 1.0, 2.0, 3.0);

vectors[3] = new TwoDimensionalVector("(второй)", -1.0, 5.0);

vectors[4] = new ThreeDimensionalVector("(Второй)", 4.0, -2.0, 0.0);

// вызываем виртуальную функцию для каждого объекта в массиве

foreach (Vector vector in vectors)

{

Console.WriteLine(vector);

}

}

}

class Vector

{

protected string \_name;

public Vector(string name)

{

\_name = name;

}

public override string ToString()

{

return $"Имя: {\_name}.";

}

public virtual double GetLength()

{

return 0.0;

}

}

class TwoDimensionalVector : Vector

{

private double \_x;

private double \_y;

public TwoDimensionalVector(string name, double x, double y) : base(name)

{

\_x = x;

\_y = y;

}

public override string ToString()

{

return base.ToString() + $" Длина: {GetLength()}";

}

public override double GetLength()

{

return Math.Sqrt(\_x \* \_x + \_y \* \_y);

}

}

class ThreeDimensionalVector : Vector

{

private double \_x;

private double \_y;

private double \_z;

public ThreeDimensionalVector(string name, double x, double y, double z) : base(name)

{

\_x = x;

\_y = y;

\_z = z;

}

public override string ToString()

{

return base.ToString() + $" Длина: {GetLength()}";

}

public override double GetLength()

{

return Math.Sqrt(\_x \* \_x + \_y \* \_y + \_z \* \_z);

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Имя: Вектор 1.  Имя: (первый). Длина: 5  Имя: (первый). Длина: 3,7416573867739413  Имя: (второй). Длина: 5,0990195135927845  Имя: (Второй). Длина: 4,47213595499958 |

Анализ результатов:

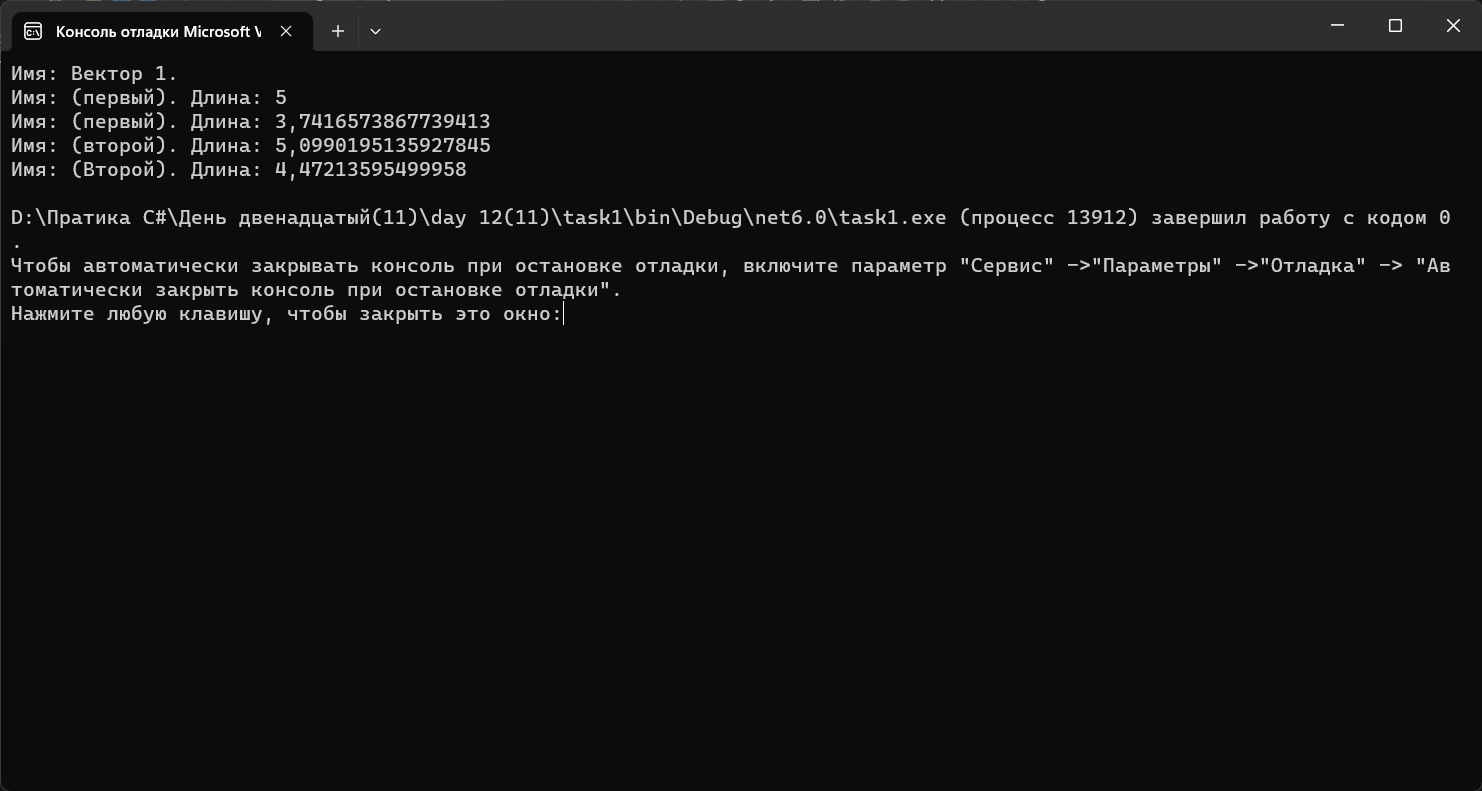


Рисунок 1.1 – Результат работы программы

Задание 2. Составить программу с абстрактным родительским классом и двумя объектами - потомками. Для этого модифицировать задание 2. 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).

internal class Program

{

static void Main(string[] args)

{

Vector[] vectors = new Vector[]

{

new TwoDimensionalVector("первый", 4, 3),

new ThreeDimensionalVector("второй",1, 2, 3),

new TwoDimensionalVector("третий", 4, 7),

new ThreeDimensionalVector("четвёртый", 3, 8, 3)

};

double length = 0;

foreach (var vector in vectors)

{

Console.WriteLine(vector);

length += vector.GetLength();

}

Console.WriteLine($"Сумма длин: {length:f2}");

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace task2

{

abstract class Vector

{

protected string \_name;

public Vector(string name)

{

\_name = name;

}

public override string ToString()

{

return $"Имя: {\_name}.";

}

public abstract double GetLength();

}

class TwoDimensionalVector : Vector

{

private double \_x;

private double \_y;

public TwoDimensionalVector(string name, double x, double y) : base(name)

{

\_x = x;

\_y = y;

}

public override string ToString()

{

return base.ToString() + $" Длина: {GetLength():f2}";

}

public override double GetLength()

{

return Math.Sqrt(\_x \* \_x + \_y \* \_y);

}

}

class ThreeDimensionalVector : Vector

{

private double \_x;

private double \_y;

private double \_z;

public ThreeDimensionalVector(string name, double x, double y, double z) : base(name)

{

\_x = x;

\_y = y;

\_z = z;

}

public override string ToString()

{

return base.ToString() + $" Длина: {GetLength():f2}";

}

public override double GetLength()

{

return Math.Sqrt(\_x \* \_x + \_y \* \_y + \_z \* \_z);

}

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Имя: первый. Длина: 5,00  Имя: второй. Длина: 3,74  Имя: третий. Длина: 8,06  Имя: четвёртый. Длина: 9,06  Сумма длин: 25,86 |

Анализ результатов:

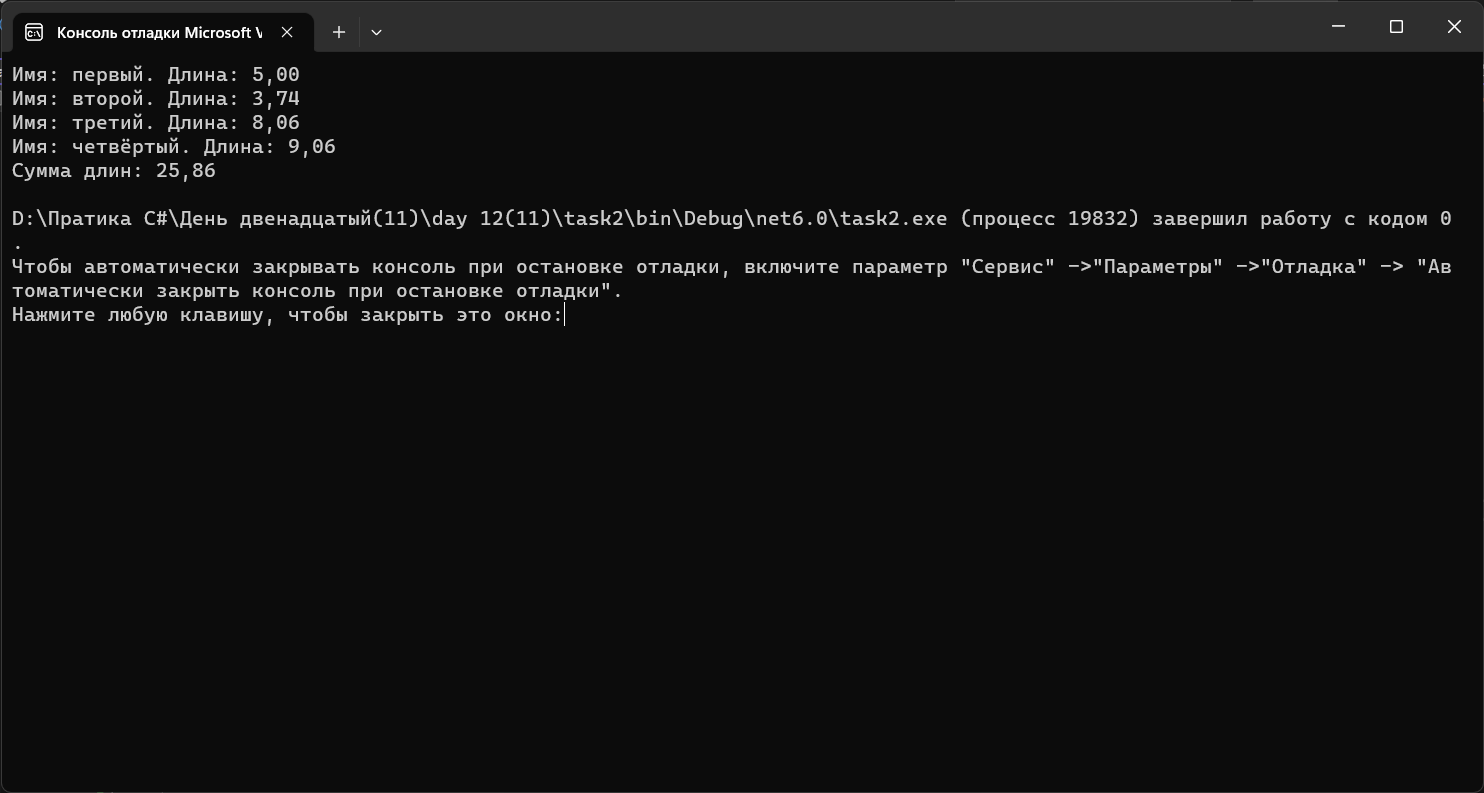
**

Рисунок 1.2 – Результат работы программы

работы программы