**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 – вектор (поле – название). Потомки: двумерный вектор (поля: – компоненты вектора), трехмерный вектор (поле – размерность). Полиморфные методы: длина вектора.

Листинг программы:

public class Vector

{

public String Name { get; set; }

public Vector()

{

Name = "Vector";

}

public virtual void Print()

{

Console.WriteLine(Name);

}

public virtual double VectorLength()

{

return 0;

}

}

public class XYVector : Vector

{

public double X { get; set; }

public double Y { get; set; }

public XYVector(double x, double y)

{

this.X = x;

this.Y = y;

}

public override void Print()

{

Console.WriteLine(Name);

Console.WriteLine("Длинна: " + VectorLength());

Console.WriteLine("");

}

public override double VectorLength()

{

return Math.Sqrt(Math.Pow(X, 2) + Math.Pow(Y, 2));

}

}

public class XYZVector : Vector

{

public double X { get; set; }

public double Y { get; set; }

public double Z { get; set; }

public XYZVector(double x, double y, double z)

{

this.X = x;

this.Y = y;

this.Z = z;

}

public override void Print()

{

Console.WriteLine(Name);

Console.WriteLine("Площадь: " + VectorLength());

Console.WriteLine("");

}

public override double VectorLength()

{

return Math.Sqrt(Math.Pow(X, 2) + Math.Pow(Y, 2) + Math.Pow(Z, 2));

}

}

class Program

{

public static void Main()

{

double z, x, y; string k; int R;

Console.WriteLine("Введите число векторов");

R = Convert.ToInt32(Console.ReadLine());

Vector[] Vectors = new Vector[R];

for (int i = 0; i < R; i++)

{

Console.WriteLine("Тип вектора(2/3) : ");

k = Console.ReadLine();

if (k == "2")

{

Console.WriteLine("Введите х");

x = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите у");

y = Convert.ToInt32(Console.ReadLine());

Vectors[i] = new XYVector(x, y);

Console.WriteLine("Длинна" + " " + Vectors[i].VectorLength());

}

else if (k == "3")

{

Console.WriteLine("Введите х");

x = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите у");

y = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите z");

z = Convert.ToInt32(Console.ReadLine());

Vectors[i] = new XYZVector(x, y, z);

Console.WriteLine("Площадь" + " " + Vectors[i].VectorLength());

}

else if (k != "1" || k != "2")

{

Console.WriteLine("Неизвстный тип.");

}

}

Таблица 11.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2, 3, 5, 8, 9, 2, 5, 1 | 13.03840, 5.09901 |

Источник: собственная разработка

Анализ результатов:

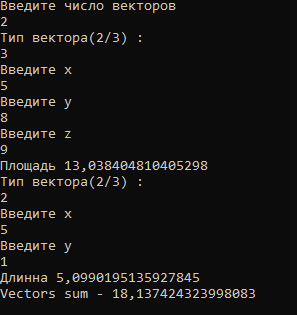


Рисунок 11.1 – Результат работы программы

Источник: собственная разработка