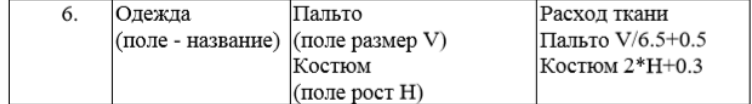
**ПРАКТИЧЕСКОЕ ЗАДАНИЕ 11**

**МЕХАНИЗМ НАСЛЕДОВАНИЯ. 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).



Листинг программы

try

{

var s = new List<Clothes>(){new Clothes("Unidentified Clothes"),

new Coat("Coat",1), new Suit("Suit",1),

new Coat("Coat",2), new Suit("Suit",2),

new Coat("Coat",3), new Suit("Suit",3)};

foreach (Clothes p in s)

{

p.Calculate();

Console.WriteLine();

}

Console.ReadLine();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

class Clothes

{

private string \_Cloth { get; set; }

public Clothes(string cloth)

{

this.\_Cloth = cloth;

}

public virtual void Calculate() //virtual - можем переписывать этот метод (метод Calculate())

{

Console.WriteLine("Calculating the {0} / Object type is: {1}", this.\_Cloth, this.GetType());

}

}

class Coat : Clothes

{

private double boxsizeV;

public Coat(string cloth, double boxsizeV) : base(cloth)

{

this.boxsizeV = boxsizeV;

}

public override void Calculate()

{

base.Calculate(); //base - вызывает этот метод в родительском классе (метод Calculate())

Console.WriteLine($"Wingspan = {boxsizeV}");

Console.WriteLine($"Result = {(double)(boxsizeV / 6.5 + 0.5)}");

}

}

class Suit : Clothes

{

private double fieldgrowthH;

public Suit(string cloth, double fieldgrowthH) : base(cloth)

{

this.fieldgrowthH = fieldgrowthH;

}

public override void Calculate() // override - используется, чтобы переписать метод (метод Calculate() родительского класса)

{

base.Calculate();

Console.WriteLine($"height = {fieldgrowthH}");

Console.WriteLine($"Result = {(double)(2 \* fieldgrowthH + 0.3)}");

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Данные об одежде | Информация об одежда, расход ткани |

Анализ результатов:

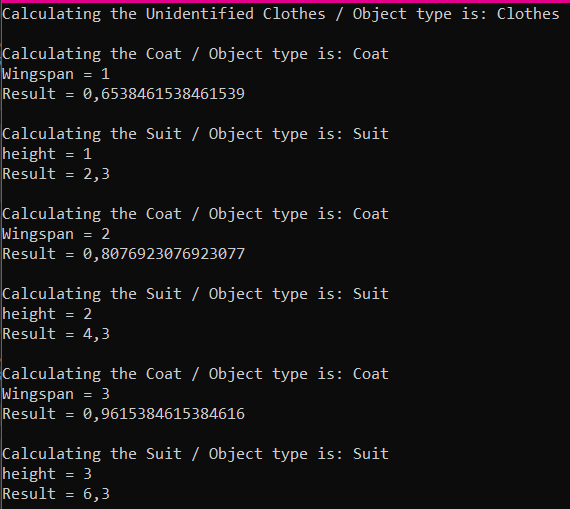


Рисунок 1.1 – Результат работы программы

Задание 2. Организовать вычисление суммарного расхода ткани.

Листинг программы

try

{

var s = new List<Clothes>() { new Coat("Coat", 1), new Suit("Crow", 1), new Coat("Coat", 2), new Suit("Crow", 2), new Coat("Coat", 3), new Suit("Crow", 3) };

foreach (Clothes p in s)

{

p.Calculate();

Console.WriteLine();

}

Console.ReadLine();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

abstract class Clothes

{

public abstract string Cloth { get; set; }

public abstract void Calculate();

}

class Coat : Clothes

{

public override string Cloth { get; set; }

private double boxsizeV;

public Coat(string cloth, double boxsizeV)

{

this.Cloth = cloth;

this.boxsizeV = boxsizeV;

}

public override void Calculate()

{

Console.WriteLine("Calculating the {0} / Object type is: {1}", this.Cloth, this.GetType());

Console.WriteLine($"Wingspan = {boxsizeV}");

Console.WriteLine($"Result = {(double)(boxsizeV \* 1 / 2000)}");

}

}

class Suit : Clothes

{

public override string Cloth { get; set; }

private double fieldgrowthH;

public Suit(string cloth, double fieldgrowthH)

{

this.Cloth = cloth;

this.fieldgrowthH = fieldgrowthH;

}

public override void Calculate() // override - используется, чтобы переписать метод (метод Calculate() родительского класса)

{

Console.WriteLine("Calculating the {0} / Object type is: {1}", this.Cloth, this.GetType());

Console.WriteLine($"height = {fieldgrowthH}");

Console.WriteLine($"Result = {(double)(0.8 \* fieldgrowthH)}");

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Данные об одежде | Информация об одежда, расход ткани |

Анализ результатов:

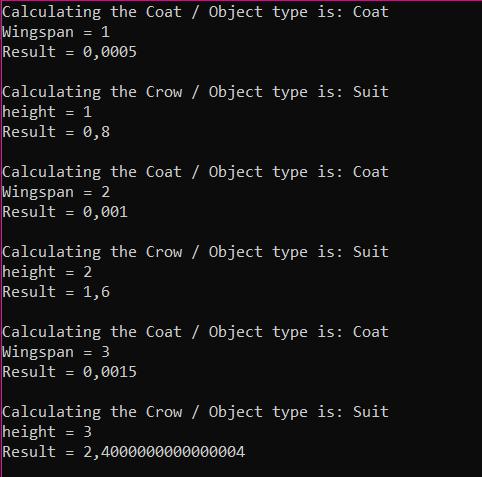


Рисунок 1.2 – Результат работы программы