# 编程要求：

编写C#控制台应用程序，建立鸟类的定义，并定义企鹅、老鹰、鸡三种动物，判断它们是否属于鸟类，并输出判断结果

## 劣质代码实现与分析

using System;

namespace BirdsCheck

{

class Program

{

//主函数 【1】这种把名称翻译一遍的注释是毫无意义的

static void Main(string[] args)

{

Bird x = new Bird();

penguin a = new penguin();

eagle b = new eagle();

chicken c = new chicken();

【2】这种a/b/c命名写起来方便，但没有任何含义，不要使用，除非是约定俗称的一些字母，比如循环语句中用于计数的i j k

if(a.CanFly == x.CanFly)

{

Console.WriteLine("penguin is bird");

}

else

{

Console.WriteLine("penguin is not bird");

}

if (b.CanFly == x.CanFly)

{

Console.WriteLine("eagle is bird");

}

else

{

Console.WriteLine("eagle is not bird");

}

if (c.CanFly == x.CanFly)

{

Console.WriteLine("chicken is bird");

}

else

{

Console.WriteLine("chicken is not bird");

}

【3】上面这三段代码除了名称以外一模一样，看着让人非常不爽

Console.ReadKey();

}

}

//鸟

class Bird

{

public bool CanFly = true;

}

//企鹅

【4】类名应该是大写开头

class penguin

{

public bool CanFly = false;

}

//老鹰

class eagle

{

public bool CanFly = true;

}

//鸡

class chicken

{

public bool CanFly = false;

}

【5】在这个例子中，企鹅、老鹰和鸡三种东西拥有共同的属性，完全可以放在一个类中

}

【6】所有的类都放在一个文件中，导致单个文件过长，且不独立

## 对上述【1】-【6】问题的修改

【6】将不同的类放在不同的文件中

//-------------------------File 1-------------------------//

using System;

namespace BirdsCheck

{

class Program

{

static void Main(string[] args)

{

Animal penguin = new Animal("penguin", true, true, true, true, false);

Animal eagle = new Animal("eagle", true, true, true, true, true);

Animal chicken = new Animal("chicken", true, true, true, true, false);

【2】有意义的命名

Bird.CheckIfAnimalIsBird(penguin);

Bird.CheckIfAnimalIsBird(eagle);

Bird.CheckIfAnimalIsBird(chicken);

【1】有意义的注释

//黑框显示出来后，输入一个字符才会关闭，便于查看输出结果

Console.ReadKey(false);

}

}

}

//-------------------------File 2-------------------------//

namespace BirdsCheck

{

/// <summary>

/// 定义一切动物的类

/// </summary>

【4】类名应该是大写开头

【5】将属性相同的类合并为一个类

public class Animal

{

public string Name;

public bool IsOviparity;

public bool IsHomothermal;

public bool HasWing;

public bool HasFeather;

public bool CanFly;

public Animal(){ }

/// <summary>

/// 给出动物属性的构造函数

/// </summary>

public Animal(string name, bool isOviparity, bool isHomothermal, bool hasWing, bool hasFeather, bool canFly)

{

Name = name;

IsOviparity = isOviparity;

IsHomothermal = isHomothermal;

HasWing = hasWing;

HasFeather = hasFeather;

CanFly = canFly;

}

}

}

//-------------------------File 3-------------------------//

using System;

namespace BirdsCheck

{

/// <summary>

/// 定义所有鸟类的类

/// </summary>

public class Bird : Animal

{

/// <summary>

/// 检查一个动物是否是鸟类

/// </summary>

【3】将重复的代码抽取成一个方法

static public bool CheckIfAnimalIsBird(Animal animal)

{

if (animal.IsOviparity == true && animal.IsHomothermal == true && animal.HasWing == true && animal.HasFeather == true)

{

Console.WriteLine(animal.Name + " is bird");

return true;

}

else

{

Console.WriteLine(animal.Name + " is not bird");

return false;

}

}

}

}

## 以上程序还有其他问题，同学们可以进一步发掘