public static void TestDemo04()

{

// 这是继承的遗传特性，可以自动继承父类的非私有成员（字段、属性、方法）

Dog dog = new Dog();

dog.Food = "骨头";

dog.Eat(); // 从父类Animal中继承的Eat方法和Food字段

dog.MakeNoise();

Cat cat = new Cat();

cat.Age = 7; // 从父类Animal中继承的Age字段

cat.MakeNoise();// 从父类Animal中继承的MakeNoise方法

// 继承也会有变异特性，子类可以和父类行为不一样

// 变异的形式1：子类可以添加父类没有的成员（字段、属性、方法）

dog.Guard(); // 这是狗类所特有的，它的父类没有这个方法

cat.Climb(); // 这是猫类所特有的，它的父类没有这个方法

}

namespace NsDemo03

{

public class Dog

{

public int Age;

public string Food;

// 狗会叫，“旺旺"

public void MakeNoise()

{

Console.WriteLine("旺旺");

}

// 狗会吃东西

public void Eat()

{

Console.WriteLine("狗吃" + Food);

}

public void Guard()

{

Console.WriteLine("狗会看大门");

}

}

public class Cat

{

public int Age;

public string Food;

// 猫会叫，“喵喵"

public void MakeNoise()

{

Console.WriteLine("喵喵");

}

// 猫会吃东西

public void Eat()

{

Console.WriteLine("猫吃" + Food);

}

public void Climb()

{

Console.WriteLine("猫会爬树");

}

}

}

namespace NsDemo04

{

public class Animal

{

public int Age;

public string Food;

// 动物会叫

// 方法前面加virtual，表明这个方法子类可以重写

public virtual void MakeNoise()

{

Console.WriteLine("动物会叫");

}

// 动物会吃东西

public void Eat()

{

Console.WriteLine("动物会吃" + Food);

}

}

// 【public class 子类 : 父类】

public class Dog : Animal

{

public void Guard()

{

Console.WriteLine("狗会看大门");

}

/\*

\* override表明子类重写了父类的同名方法

\* 父类的这个方法要用virtual修饰

\*/

public override void MakeNoise()

{

Console.WriteLine("汪汪");

}

}

public class Cat : Animal

{

public void Climb()

{

Console.WriteLine("猫会爬树");

}

}

}