**接口的声明**

接口的声明语法格式如下：

[可见度] interface 接口名称 [extends 其他的接口名] {

// 声明变量

// 抽象方法

}

Interface关键字用来声明一个接口。下面是接口声明的一个简单例子。

NameOfInterface.java 文件代码：

/\* 文件名 : NameOfInterface.java \*/

import java.lang.\*;

//引入包

public interface NameOfInterface

{

//任何类型 final, static 字段

//抽象方法

}

接口有以下特性：

接口是隐式抽象的，当声明一个接口的时候，不必使用abstract关键字。

接口中每一个方法也是隐式抽象的，声明时同样不需要abstract关键字。

接口中的方法都是公有的。

实例

Animal.java 文件代码：

/\* 文件名 : Animal.java \*/

interface Animal {

public void eat();

public void travel();

}

接口的实现

当类实现接口的时候，类要实现接口中所有的方法。否则，类必须声明为抽象的类。

类使用implements关键字实现接口。在类声明中，Implements关键字放在class声明后面。

实现一个接口的语法，可以使用这个公式：

接口语法：

...implements 接口名称[, 其他接口名称, 其他接口名称..., ...] ...

实例

MammalInt.java 文件代码：

/\* 文件名 : MammalInt.java \*/

public class MammalInt implements Animal{

public void eat(){

System.out.println("Mammal eats");

}

public void travel(){

System.out.println("Mammal travels");

}

public int noOfLegs(){

return 0;

}

public static void main(String args[]){

MammalInt m = new MammalInt();

m.eat();

m.travel();

}

}

以上实例编译运行结果如下:

Mammal eats

Mammal travels

重写接口中声明的方法时，需要注意以下规则：

类在实现接口的方法时，不能抛出强制性异常，只能在接口中，或者继承接口的抽象类中抛出该强制性异常。

类在重写方法时要保持一致的方法名，并且应该保持相同或者相兼容的返回值类型。

如果实现接口的类是抽象类，那么就没必要实现该接口的方法。

在实现接口的时候，也要注意一些规则：

一个类可以同时实现多个接口。

一个类只能继承一个类，但是能实现多个接口。

一个接口能继承另一个接口，这和类之间的继承比较相似。