內部類：類的第五個成員

1. 定義： Java中允許將一個類A聲明在另一個類B中，則類A就是內部類，類B稱為外部類
2. 內部類的分類：

成員內部類(類裡面的)（靜態、非靜態） vs 局部內部類(方法內、代碼塊內、構造器內)

1. 成員內部類的理解：

一方面，作為外部類的成員：

>調用外部類的結構

>可以被static修飾:外部類不行,內部類可以。

>可以被4種不同的權限修飾

另一方面，作為一個類：

> 類內可以定義屬性、方法、構造器等

> 可以被final修飾，表示此類不能被繼承。言外之意，不使用final，就可以被繼承

> 可以被abstract修飾

1. 成員內部類

4.1如何創建成員內部類的對象?(靜態的、非靜態的)

//創建靜態的Dog內部類的實例(靜態的成員內部類):

Person.Dog dog = **new** Person.Dog();

//創建非靜態的Bird內部類的實例(非靜態的成員內部類):

// Person.Bird bird = new Person.Bird();//錯誤的

Person p = **new** Person();

Person.Bird bird = p.**new** Bird();

4.2如何在成員內部類中調用外部類的結構?

**class** Person{

String name = "小明";

**public** **void** eat(){

}

//非靜態成員內部類

**class** Bird{

String name = "杜鵑";

**public** **void** display(String name){

//方法的形參

System.***out***.println(name);

//內部類的屬性:一般都省略this，除了當跟形參同名時才加this

System.***out***.println(**this**.name);

//外部類的屬性

System.***out***.println(Person.**this**.name);

//調用外部類方法

Person.**this**.eat();//通常都省略Person.**this**.除非同名時。

}

}

}

1. 局部內部類的使用

//返回一個實現了Comparable接口的類的對象

**public** Comparable getComparable(){

//創建一個實現了Comparable接口的類:局部內部類

//方式一：

// class MyComparable implements Comparable{

//

// @Override

// public int compareTo(Object o) {

// return 0;

// }

//

// }

//

// return new MyComparable();

//方式二：創建Comparable的匿名實踐類的匿名對象

**return** **new** Comparable(){

@Override

**public** **int** compareTo(Object o) {

**return** 0;

}

};

}

注意點:

在局部內部類的方法中（比如：show）如果調用局部內部類所聲明的方法(比如：method)中的局部變量(比如：num)的話,要求此局部變量聲明為final的。

\* jdk 7及之前版本：要求此局部變量顯式的聲明為final的

\* jdk 8及之後的版本：可以省略final的聲明

總結:

成員內部類和局部類,在編譯以後,都會生成字節碼文件。

格式: 成員內部類:外部類$內部類名.class

局部內部類:外部類$數字 內部類名.class