

一，在java中，允许一个类的定义位于另一个类的内部，前者称为内部类，后者称为外部类

相当于我们可以在类的内部定义另一个类

二，分类

- 1，和类的属性以及方法进行并列的内部类称为成员内部类
- 2，放在方法中的类称为局部内部类

成员内部类：

- ①是一个外部类的成员
 - a，可以有修饰符（4个）；
 - b，可以用static final abstract修饰
 - c，可以调用外部类的属性
- ②同时也是一个类
 - a，可以用abstract修饰
 - b，还可以在内部类中定义属性、方法、构造器

局部内部类：在方法听内定义的类称为局部内部类

关于内部类：需要知道：①如何创建成员内部类的对象（创建静态成员内部类的----直接写外部类 名

=new 外部类.内部类（）；创建静态成员

内部类的对象，可以直接通过

外部类调用静态内部类的构造器）（创建

非静态的成员内部类对象，必须

先创建外部类的对象，通过外部类的对象

调用成员内部类的构造器）

②如何区分调用外部类和内部类的变量（尤其

是变量重名的时候）（外部类

名.this.属性名-----调用的是外部类的

属性；this.属性名-----调用的是内部

类的属性；属性名-----调用的是在主类

中设置的属性）

③局部内部类的使用

```
//局部内部类
//如下的使用方式较少
public void method1(){

    class InnnerClass{

    }

}
```

//常常使用一个方法，使其返回值为某个类或接口的对象。而这个类或接口在方法内部创建

```
public Comparable getComparable(){
    //1.创建一个实现Comparable接口的类:局部内部类
    class MyComparable implements Comparable{

        @Override
        public int compareTo(java.lang.Object o) {
            return 0;
        }

    }
    //2.返回一个实现类的对象
    return new MyComparable();
}
```

接口没有构造器，其余的都有构造器

```
//使用方式二
public Comparable getComparable1(){
    //返回一个实现Comparable接口的匿名内部类的对象
    return new Comparable(){

        @Override
        public int compareTo(java.lang.Object o) {
            // TODO Auto-generated method stub
            return 0;
        }

    };
}
```

