# 内部类

## 博客一：

前言：一般情况下，类和类之间是谁相互独立的，内部类的意思是打破这种独立，让一个类成为另一个类的内部成员，和成员变量、成员方法同等级别。为什么要使用内部类？采用内部类这种技术，可以隐藏细节和内部结构，封装性更好，让程序的构造更加合理。

### 1、成员内部类

成员内部类的作用就是讲内部类当做外部类的一个成员变量/成员方法来使用，所以必须依赖于外部类的 对象才能调用，用法和成员变量/方法是一致的。

public class OuterClass {

//成员变量

private String outerName;

//成员方法

public void display() {

System.out.println("OuterClass display");

}

//成员内部类

public class InnerClass {

//成员变量

private String innerName;

//成员方法

public void display() {

System.out.println("InnerClass display");

}

}

public static void main(String[] args) {

//实例化外部类

OuterClass outerClass = new OuterClass();

outerClass.display();

//实例化非静态内部类需要外部类的实例

OuterClass.InnerClass innerClass = outerClass.new InnerClass();

innerClass.display();

}

}

成员内部类可以直接访问外部类中的所有成员。但外部类如果要访问成员内部类的成员需要通过实例来取。

### 2、局部内部类

局部内部类只能在局部作用域内使用，作用域外无法访问。

public class OuterClass {

//成员变量

private String outerName;

//成员方法

public void display() {

System.out.println("OuterClass display");

//局部内部类

class InnerClass {

//成员变量

private String innerName;

//成员方法

public void display() {

System.out.println("InnerClass display");

}

}

//局部内部类只能在局部作用域内使用，作用域外无法访问。

InnerClass innerClass = new InnerClass();

innerClass.display();

}

public static void main(String[] args) {

//实例化外部类

OuterClass outerClass = new OuterClass();

outerClass.display();

}

}

局部内部类只能访问作用域中加final关键字的成。

### 3、静态内部类

静态内部类的构造不需要依赖外部类的对象，类中的所有静态组件都不需要依赖任何对象，可以直接通过类本身进行构造。

public class OuterClass {

//成员变量

private String outerName;

//成员方法

public void display() {

System.out.println("OuterClass display");

}

//静态内部类

public static class InnerClass {

//成员变量

private String innerName;

//成员方法

public void display() {

System.out.println("InnerClass display");

}

}

public static void main(String[] args) {

//实例化外部类

OuterClass outerClass = new OuterClass();

outerClass.display();

//实例化静态内部类

OuterClass.InnerClass innerClass = new OuterClass.InnerClass();

innerClass.display();

}

}