

第十章 内部类

为什么使用内部类



- 什么是内部类
- 内部类的好处
- 内部类的分类



□ 实例内部类

□ 语法：

```
Class Outer{  
  
    Class Inner{  
  
    }  
  
}
```

□ 访问修饰



注意：

- ❑ 实例内部类内部不允许定义任何静态内容
- ❑ 外部类静态方法中不能访问实例内部类数据
- ❑ 如果外部类成员变量、内部类成员变量、内部类方法中局部变量同名。



□ 定义

```
class Outer{  
    static Class Inner{  
    }  
}
```



注意：

□ 接口中可以定义静态内部类，静态内部接口



- ❑ 成员内部类继承问题
- ❑ 子类内部类与父类内部类同名情况



□ 定义

```
class Outer{  
  
    public void function(){  
        class Inner{  
  
        }  
    }  
}
```




- 注意：
- 局部内部类不能声明静态成员
- 局部内部类处于静态方法中只能访问外部类静态成员
- 局部内部类处于实例方法中可访问外部类静态成员和实例成员
- 局部类对局部变量的访问



□ 语法：

```
new 父类 ( 参数列表 ) | 实现接口 ( )  
{  
    //匿名内部类的类体部分  
}
```

□ 好处



□ 注意：

- 必须实现一个类或一个接口。
- 不能定义静态数据
- 不能定义构造器
- 不能定义构造器
- 不能是抽象的
- 传参问题
- 同名问题