

包：

package，用于管理源代码

特点：

1. 包以公司域名反写命名 例如 com,baidu
2. 实质是一个有层次的文件夹，点代表下一级
3. 包在类中，代码第一行需要加上包的路径
4. 使用包中的类时，需要导入包
5. 为了区分不同的类，可以使用全类名。即包路径+类名

面向对象的特征：

1.封装

2.继承

3.多态

4.抽象

封装：

把类的实现部分隐藏起来，选择性的暴露方法

标准：高内聚，低耦合

内聚：模块内部成员的关联程度

耦合：模块间的关联程度

访问修饰符：

public ：公开的，在任何地方都可以访问

使用方法

在类的外部： 对象. 属性名

在类的内部： 属性名 或 this. 属性名

protected ：受保护的。 同包，本类，或子类中可以访问

问

使用方法

在类的外部： 对象. 属性名

在类的内部： 属性名 或 this. 属性名

省缺：在本类或同包中，可以访问。

`private`: 私有的，只能在本类中访问（可以设置控制器，

通过制器在外部调用）

控制器即为`get`和`set`方法

设置器

即为`set`方法

```
访问修饰符 无返回值类型 set属性名（首字母大写）（数据类型 属性名）{  
  
    this.属性 = 属性名;  
}
```

访问器

即为`get`方法

```
访问修饰符 返回值类型 get属性名（首字母大写）（）{  
  
    return 属性名;  
}
```

继承：

子类继承父类的属性和方法

特点：

1. 单根继承：全部都是`Object`的子类
2. 单向继承：一个类只能有一个父类，一个类可以有多个子类
3. 构造方法不能被继承
4. 子类的构造方法内需要调用父类的构造方法
`super` 用于调用父类的属性和方法
`super()` 调用父类的无参构造方法，必须出现构造方法的第一行
5. 子类的构造方法内没有调用父类的构造方法，
系统会默认调用父类的无参构造方法
6. 调用父类的构造方法必须出现在第一行
7. 当父类的构造方法不满足子类需求时，子类可以重写父类的

方法

不完全重写： 在父类的基础上增加新的功能

完全重写： 子类重写这个方法，不使用父类的代码

@Override

```
public void 构造方法 () {
```

```
    内容
```

```
}
```

final :

1. 修饰变量： 变量的值不能被修改
2. 修饰属性： 一般有 public static final 修饰
3. 修饰方法： final 修饰方法，方法不能重写
4. 修饰类： 类不能被继承