

OOA面对对象的分析

OOD面对对象的设计

OOP面对对象的程序

封装：

把相关数据及操作组织在一个类里。

方法重载。overload

参数传递，只有一种方式，就是按值传递。

子类在构造时，会调用父类的构造。

父类初始化在子类前面。

1. new时，虚拟机分配内存，构造对象，成员变量是默认初始化，
2. 初始化父类成员，代码块。
3. 初始化子类成员，代码块和子类的构造。

先初始化父类对象，初始化时父类和子类有先后顺序

多态，重载（编译时多态），运行时多态，定义类型必须是父类。

子类方法可以覆盖同名父类方法，子类属性可以覆盖父类同名属性。

声明是父类的时候，他只能看到父类的属性，方法可以覆盖（运行时多态，重载），属性（没有运行时多态）。子类覆盖父类的属性时，在声明为父类时，访问的是父类的属性，声明是子类时，访问的是子类的属性。

子类中不能直接访问父类的私有成员。

方法覆盖是多态的另一种方式。

子类中有与父类的一个方法具有相同的返回类型，相同的方法名，

子类不能覆盖父类的私有方法。在父类方法可许可访问时，只可以放大访问范围，不可以缩小。

子类可以替代父类（里式代换）

抽象类：含有抽象方法的类。抽象类不一定含有抽象方法。

final class 不可以被继承。最终类，

接口：特殊的抽象类

接口中的方法都是抽象方法。

一个类可以有多个接口。

接口与接口之间可以多继承

接口与抽象类的区别：

异：接口中所有方法都是抽象的，抽象类中可以有非抽象的方法。

且，抽象类中的方法可以是实现的。

在属性定义上，在接口中所有属性都是public static final 且默认就是。

抽象类中的属性不是public static final 可以是private protected 也可以是静态或非静态，常量或变量。

继承角度：接口可以多继承（多实现），抽象类只能单继承。

同：都不能实例化（不能new），只能用来继承。

内部类：在一个类的内部定义的类，称为内部类，有4种

- 1. 一个类的内部定义一个类，实例内部类。
- 2. 静态内部类
- 3. 局部内部类 方法里面定义一个类。局部内部类
- 4. 匿名内部类（创建的是父类对象，父类可以是接口，类，抽象类）

访问性：内部类可以直接访问外部类的属性和方法，静态内部类访问的外部类属性或方法要是静态 的；

外部类不能直接访问内部类，通过内部类实例对象可以访问内部类属性

访问权限控制：

	同一类中	同一个包中（包含同一个包中的子类）	子类(不同包的子类)	其他
public	yes	yes	yes	yes
protected	yes	yes	yes	no
default	yes	yes	no	no
private	yes	no	no	no

类只能是public或默认

Object 类，超类，Object类的方法 toString(), hashCode(), equals()

main方法，入口约定

包装类：

String类实例创建后值不会改变，值为常量，貌似修改是一个新的String类实例。

异常，

运行时发生的错误，处理这些异常就称为异常处理。

一旦发生异常，程序将突然停止，且控制将返回运行环境。

发生异常后如不对异常发生前分配的所有资源进行处理，这些资源将保留在分配状态，这将导致资源漏洞。

异常处理机制采用一个统一和相对简单的抛出和处理错误的机制。如果一个方法本身能引发异常，当所调用的方法出现异常时，

检查异常（非运行时y）；非检查异常（运行时异常），finally中进行资源的清理。