

编程入门基础

——抽象类与接口

主讲教师：耿宇航

抽象方法

- 只有定义的形式,而没有具体实现的方法,叫做抽象方法
- 抽象方法,为子类规定了方法的**调用格式**,具体操作内容等待子类去完成。
- 生活中的例子:
 - 空白的简历
 - 只有格式而没有内容
 - 如何编程?
 - 先分析需求
 - 再设计
 - 再编码
 - 再测试

人类思维的优点

- 人能够抽象地、概括地思考问题
- 抛开细节，只思考共性
- 抛开具体，只思考一般的特征

名词解释

- 抽象
- 抽 = 抽取, 抽出
- 象 = 表象, 特征, 特质
- 抽象 = 抽出 (概括出) 共同的特征
- 抽象就是抽出共性!

抽象类

- 至少包含一个抽象方法的类叫抽象类
- 抽象类不能实例化对象
- 抽象类的唯一用途是被其它类继承
- 抽象类提供概括性的，高层的描述

语法

- 以**abstract**修饰方法
- 以**abstract**修饰类
- 在其中定义了抽象方法的类，必须声明为**abstract**类型
- 继承抽象类的类，有责任去实现它所继承的所有抽象方法。
- 继承类也可以不实现抽象方法，这时，它必须也声明为**abstract**类型

抽象类的作用

- 从更一般的观点上求解问题
- 避免陷入细节
- 软件的蓝图化设计
- 实现类间的松耦合
- 黑社会成功之密诀：
 - 你知道的越少就越安全

抽象类能做什么？

- 站在高层去处理流程，在不知道细节的情况下也能处理。
- 条件：具体实现的类，**必须**从这个抽象类继承（直接或间接）
- 问题：某个具体类如果具有多个抽象的性质如何处理？
 - 不能使用多重继承

绝对抽象的类-接口

- 接口就虚而又虚的类
 - 它只有方法的形式描述，没有任何实际内容
 - 它不包含任何实例变量
- 只能从一个类继承，但可以声明实现了多个接口
 - 接口是多重继承的替代品

接口

- 接口就是一个合同，一个约定
- 实现接口，就是宣誓遵守这个约定
- 接口的目的：便于类间的交流，合作
- 生活中的接口：
 - GB220v三相插头标准就是接口
 - 川菜标准就是接口
 - 8欧姆双声道耳机就是接口

接口的妙用

- 与接口打交道，而不是具体的类，这有利于实现类间的松耦合
- 同一接口不同的实现间，可以相互替换

接口的语法

接口的定义：

interface 接口名 **extends** 父接口

{

常量定义

方法定义

}

接口的使用：

某类名 **implements** 接口1, 接口2,...

{

实现接口中的方法

}

注意

- 接口中的数据具有： public, static, final属性。
- 接口中的方法具有： public, abstract属性
- 允许从一个类继承， 允许实现多个接口

设计提示

- 能用接口的地方，不用类
- 一个模块尽量少知道其它模块的信息，知道的越少，对今后的维护越有利

谢 谢 !