#### 封装

1. 面相对象开发的好处：
   1. 和人类的思维习惯一致
   2. 交流更加流畅
   3. 提高设计和开发效率
2. 如何抽象出类
   1. 发现类
   2. 发现类中的属性
   3. 发现类中的方法
   4. 注意： 类中声明的属性和方法，我们只放和业务相关的属性和方法
3. 构造方法
   1. 作用：
      1. 初始化对象信息，比如属性值
      2. 创建对象 Student stu = new Student();
   2. 语法：

访问修饰符 构造方法名(参数列表){

//初始化代码

}

1. 构造方法不存在返回值类型
2. 构造方法名必须和类名一致
3. 参数列表可以根据现实情况添加和不添加
   1. 无参构造方法： 系统会默认提供一个无参构造方法

public Dog(){

}

当实例化对象的时候：

Dog dog = new Dog(); //这个时候调用的就是无参构造方法，通过构造方法初始化属性值

* 1. 有参构造方法： 是自定义的，参数的数量可以选择，但是当写入了有参构造方法之后，类中默认无参构造方法，则不会在存在，需要指定出无参构造。
  2. this关键字： 指代本类的对象，用以区分同名的成员变量。
  3. 快捷键：

shift+alt+s 选择倒数第二个 无参

shift+alt+s 选择倒数第三个 有参

1. 方法重载：
   1. 前提：必须是在同一个类中
   2. “一同三不同”：
      1. 方法名必须一致
      2. 参数列表不同（参数数据类型不同、参数个数不同）
      3. 和访问修饰符、返回值类型无关
2. 静态static
   1. 可以修饰变量和方法，调用的时候需要用类名.变量名和类名.方法名的方式去调用
   2. 所以，static修饰的变量应该是一个类的属性，而不是方法内的局部变量。
   3. 当项目运行，我们类中静态的变量和方法会优先加载到内存中。
3. 面向对象的三大特征：
   1. 封装：
      1. 定义：将类中的某些信息隐藏，外部不可以直接访问，但是类中可以提供对外访问的途径。
      2. 特点：隐藏内部细节，提供对外访问接口
      3. 好处：
         1. 只能通过规定方法访问数据
         2. 隐藏类的实现细节
         3. 方便加入控制语句
         4. 方便修改实现
      4. 实现过程：
         1. 先把属性设置成为private私有的
            1. private和public都是访问修饰符，

但是public表示公共的，也就是任何类都可以访问的，

而private则是私有的，它的访问权限是最小的，被private修饰的变量和方法只能够供本类调用，其他类无法调用。

* + - 1. 生成对外可以访问的(即访问修饰符为public 的getter方法和setter方法，其中getter方法是外部用来获取私有属性值的，setter是外部用来设置私有属性值的
      2. 可以在getter和setter方法中加入流程控制语句
  1. 继承：
  2. 多态：

1. 一个类的标准：
   1. 写上属性，并且设置成私有private 作用：私有的属性只能够让本类访问
   2. 封装getter和setter方法 作用： 给外部提供访问私有属性的接口
   3. 生成有参和无参构造 作用：1、创建对象使用构造方法 2、初始化对象信息
   4. 重写toString()方法 作用：测试数据

8、访问修饰符总结：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 访问修饰符 | 本类 | 同包 | 子类 | 其他 |
| private(私有的) | √ | - | - | - |
| 默认 | √ | √ | - | - |
| protected(受保护的) | √ | √ | √ | - |
| public(公共的) | √ | √ | √ | √ |

# 问题

1. 构造方法的作用是什么，有什么特点?
   1. 作用：
      1. 初始化对象信息，比如属性值
      2. 创建对象 Student stu = new Student();
   2. 语法：

访问修饰符 构造方法名(参数列表){

//初始化代码

}

1. 构造方法不存在返回值类型
2. 构造方法名必须和类名一致
3. 参数列表可以根据现实情况添加和不添加
4. 什么是方法重载，特点是什么
   1. 前提：必须是在同一个类中
   2. “一同三不同”：
      1. 方法名必须一致
      2. 参数列表不同（参数数据类型不同、参数个数不同）
      3. 和访问修饰符、返回值类型无关

3、被static关键字修饰的变量和方法有什么特点

* 1. 可以修饰变量和方法，调用的时候需要用类名.变量名和类名.方法名的方式去调用
  2. 所以，static修饰的变量应该是一个类的属性，而不是方法内的局部变量。
  3. 当项目运行，我们类中静态的变量和方法会优先加载到内存中。

1. 面向对象编程的三大特征?

封装、继承、多态

1. 封装的特点是什么？如何实现的?
   * 1. 特点：隐藏内部细节，提供对外访问接口
     2. 实现过程：
        1. 先把属性设置成为private私有的
           1. private和public都是访问修饰符，

但是public表示公共的，也就是任何类都可以访问的，

而private则是私有的，它的访问权限是最小的，被private修饰的变量和方法只能够供本类调用，其他类无法调用。

* + - 1. 生成对外可以访问的(即访问修饰符为public 的getter方法和setter方法，其中getter方法是外部用来获取私有属性值的，setter是外部用来设置私有属性值的
      2. 可以在getter和setter方法中加入流程控制语句

1. 一个实体类应该有哪些东西？作用分别是什么？
   1. 写上属性，并且设置成私有private 作用：私有的属性只能够让本类访问
   2. 封装getter和setter方法 作用： 给外部提供访问私有属性的接口
   3. 生成有参和无参构造 作用：1、创建对象使用构造方法 2、初始化对象信息
   4. 重写toString()方法 作用：测试数据
2. java的访问修饰符有哪些

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 访问修饰符 | 本类 | 同包 | 子类 | 其他 |
| private(私有的) | √ | - | - | - |
| 默认 | √ | √ | - | - |
| protected(受保护的) | √ | √ | √ | - |
| public(公共的) | √ | √ | √ | √ |