|  |  |
| --- | --- |
| **类型** | **描述** |
| 类中方法的顺序 | 按照public,internal,protected,private这样的顺序从上往下放置。并且public中按照增删改查的顺序放置方法。 |
| 异常处理 | 尽可能的考虑到会出现异常的数据的场景，多使用条件判断来处理异常，而不是更多的try catch（当然，使用try catch也比不使用任何异常的判断要强） |
| 代码格式 | 代码写完之后要习惯进行统一的格式化 |
| 添加注释 | ① 类注释：类的注释，需要描述类的功能、依赖和如何使用  ②代码注释：复杂的逻辑应当添加注释  ③全局变量注释：每个全局变量需要写注释  ④程序流程变化注释：switch, if, while 等条件判断地方必须写注释  ⑤public方法注释：public的方法体中的代码，需要写好详尽的注释 |
| 格式规范 | ①代码缩进，采用两个空格来进行缩进，增强可读性  ②分解较长的行，每条语句只占一行，即一行只有一个语句  ③包命名，使用单一的、小写单词来命名每一个包，日java.io和java.net  ④类命名，类的每个单词的首字母大写使用名词来命名类一组有关系的属性、静态服务或常量的类命名采用复数  ⑤方法命名，方法的首单词的首字母小写，后面的单词的首字母大写使用动词来命名方法遵从JavaBeans规则来命名访问方法  ⑥变量命名，变量首单词的首字母小写，后面的单词的首字母大写使用名词来命名变量集合含义的变量采用复数形式进行命名  ⑦参数命名，当一构建器或一个“set”方法通过参数赋值给一个属性时将参数和属性采用同样的名字来命名  ⑧常量命名，每一个单词种的所有字母大写，单词之间采用下划线来进行分割，使用大写来命名常量。使它们能够很好与变量区分。 |
| 编程规范 | ①用等价的方法替换重复的表达式  ②在控制结构中尽量使用代码块来代替表达式  ③用圆括号清晰地表明运算得顺序  ④在switch语句的最后一个case语句块结束的地方加上一个break语句。  ⑤判两个对象的值是否相等，使用equals()而不是使用==  ⑥使用未检测到的、运行时异常来报告程序中未预测到的逻辑错误  ⑦使用可检验的异常处理来报告可能发生的程序错误，但是，在正常的程序运行中，这种错误是很少发生的。  ⑧用return代码报告可预测性的状态变化  ⑨不要对运行时异常或者错误不做任何处理  ⑩用一个finally语句块释放资源 |