

中心主题

▼ chp9 伪代码编程过程

▼ 9.1 创建类和方法的步骤概述

▼ 创建类

- 1 创建类的总体设计
- 2 创建对应的方法
- 3 复审并测试整个类

▪ 创建方法

▪ 9.2 伪代码

▼ 9.3 通过伪代码编程过程创建子程序

▪ 设计方法

▪ 编写具体代码

▼ 检查代码

- 自己先从脑海中先检查一遍

▼ 编译运行

- 这一步骤尽量放倒后面，防止过早给自己压力

▪ 逐行debug代码

▪ testCase

▼ 消除错误

- 如果这时候还是有很多错误，那就推倒重来

▪ 收尾工作

▪ 按需重复上述步骤

▼ 9.4 伪代码编程过程的替代方案

▪ 测试先行开发（test first devploment）

▪ 重构开发（开发维护过程中发现不好的设计就重构开发）

▪ 契约式设计（每一段程序都有前条件和后条件）

▪ 东拼西凑的开发

▼ chp10 使用变量的事项

▪ 10.1数据认知

▼ 10.2轻松掌握变量定义

- 禁用隐式声明 (vb)

▼ 10.3 变量初始化原则

- 靠近第一次使用的时候声明+初始化
- 尽可能使用final+const

▼ 10.4 作用域

- 变量引用局部化
- 尽可能缩短变量存活的时间
- 减小作用域的一般原则

▪ 10.5 持续性

▼ 10.6 绑定时间（变量与值）

- magicNumber；具名常量；方法读取
- 灵活性越强，复杂度越高：一个熟练的程序员会按照引入足够的灵活性来满足需求。但是不会增加需求之外的任何灵活性和以及相应的复杂度

▼ 10.7 数据类型与控制结构的关系

- 1 顺序型 2 选择型 3 迭代型

▪ 10.8 为每个变量只安排一个用途

▼ chp11 变量命名的力量

▼ 11.1 选择好变量命名的注意事项

- ▼ 最重要的命名注意事项
 - 名字应该尽可能的准确
 - 以问题为导向：描述what而不是how
 - 最适当的名字长度：8到16个字符
 - 变量名对作用域的影响
- ▼ 计算值限定词max index total放到变量最后
 - revenueTotal expenseTotal
 - 变量名中的常用对仗词

▼ 11.2 为特定类型的数据命名

- ▼ 循环索引
 - i j k
 - 嵌套循环：使用更长的名字

- ▼ 状态变量
 - 取一个比flag更好的名字（更具体的名字）
- ▼ 临时变量
 - 为temp、x等变量取更好的名字
- ▼ 布尔变量
 - done error success ok 等
 - 不要在变量前加is, isDone（可读性差）
 - 使用肯定的变量名, notOk（错误示范）
- 枚举类型
- 常量命名
- ▼ 11.3 命名规则的力量
 - 何时建立命名规则
 - 正式程度
- ▼ 11.4 非正式命名规则
 - 语言无关的命名规则
 - 语言相关的命名规则
 - 混合编程语言的注意事项
- ▼ 11.5 标准前缀
 - 用户自定义类型缩写（UDT）
 - 语义前缀
- ▼ 11.6 创建具备可读性的短名字
 - ▼ 一般指导性的原则
 - 使用词典里的标准缩写
 - 去掉所有非前置元音（computer-comptr |screen-scrn | apple- appl
 - 去掉虚词 and or the
 - 使用每个单词的第一个或前几个字母
 - 保留每个单词的第一个和最后一个字母
 - 使用名字中的每一个单词，不超过三个
 - 去除无用的后缀--ing ed等
 - 保留每个音节中最引人注意的发音
 - 确保不要改变变量的含义

▼ 语音缩写

- 基于单词发音而不是拼写来创建缩写(不建议)
- skating->sk8ing||highlight->hilite || before->b4

▼ 有关缩写的评论

- 不要为了只删除一个字母而创建缩写
- ▼ 缩写要保持一致
 - 对同一个缩写要不全用，要不就不用，缩写确定了也就不要改了
- 创建你可以读出来的名字
- 避免使用容易看错或者读错的字符组合
- 使用词典来解决缩写命名冲突
- 针对极短缩写增加命名对照表

▼ 11.7 应该避免的名称（增加不同变量的独特性）

- 避免使用令人误解的名字或缩写
- 避免使用具有相似含义的名字
- 避免含义不同但名字相似的名字：records resps
- 避免发音相近的名字 wrap rap
- 不要在名字中包含数字
- 避免拼写错误单词
- 避免使用常会拼错的单词
- 不要仅靠大小写区分变量名frd Frd FRD
- 不要使用程序预留关键字
- 不要使用与变量含义无关的名字
- 避免名字中易混淆的字符