

30-32

▼ chp30 编程工具

- 1 设计工具
- 2 源代码工具
- 3 可执行码工具
- 4 工具导向的环境
- 5 打造你自己的编程工具
- 6 工具幻境

▼ chp31 布局与风格

- ▼ 1 基本原则
 - 编程工作量的一小部分是写让机器读懂的程序，大部分工作是写能让他人看懂的程序。
- ▼ 2 布局技术
 - ▼ 空白
 - 分组
 - ▼ 空行
 - 合理空行占比8%-16%
 - 缩进
 - 括号
- 3 布局风格
- 4 控制结构的布局
- 5 单条语句的布局
- 6 注释的布局
- 7 子程序的布局
- 8 类的布局

▼ chp32 自说明代码

- 1 外部文档
- 2 编程风格作文档
- 3 注释或不注释
- ▼ 4 高效注释之关键

- ▼ 采用不会打断或抑制修改的注释风格
 - 减少修改注释的工作量
- ▼ 用伪代码编程法减少注释时间
 - 伪代码即注释
 - 边开发边注释
- ▼ 5 注释技术
 - ▼ 注释单行
 - ▼ 不随意添加无关注释
 - 尽量不进行单行注释
 - 不要行尾注释
 - ▼ 注释代码段
 - 说明这块代码的意图，为啥这样做
 - 代码本身应是可以清晰表达意图的
 - 注释为后面的内容做铺垫
 - 不用缩略语
 - 不要注释特殊case的代码，应该重写之
 - ▼ 注释数据声明
 - ▼ 注释数值单位
 - 替代做法：将单位写入变量名称
 - 说明变量的使用范围
 - 注释编码含义：不允许魔数
 - 注释对输入数据的限制
 - 变量名有关的注释，通过变量名关联起来
 - 注释全局数据
 - ▼ 注释控制结构
 - 如果if else 嵌套太深，注释多。解决办法是重写程序降低复杂度
 - ▼ 注释子程序（方法）
 - 注释应靠近其说明的代码，不要都放在方法体外面
 - 子程序上附有一两句简单的话即可
 - 在声明参数处就注释，不用单独提出来
 - 利用javaDoc之类的代码说明工具

- 分清输入和输出数据
- ▼ 注释接口假设
 - 程序中假定了输入的排序，合法的数值时
 - 说明方法的局限性，对全局是否有修改
- ▼ 用注释标记程序的各个部分
 - @throws @param etc。。
- ▼ 注释类、文件、程序
 - ▼ 标注类的一般原则
 - 说明类的设计方法
 - 说明局限性、用法假设等
 - ▼ 注释类接口
 - 实现不看类具体实现就能明白类的用处
 - 不要在类接口说明实现细节
 - ▼ 注释文件的一般原则
 - 说明各文件的意图和内容
 - 将姓名、电子邮件、电话放进去
 - 包含版本控制标记
 - 注释块中包含法律通告
 - 文件名要与其内容相关
- 6 IEEE标准