

# 第7章内容概要

- 程序设计语言
- 编码风格
- 软件测试基础
- ★ ■ 单元测试
- 集成测试
- 确认测试
- 白盒测试技术
- 黑盒测试技术
- 调试
- 软件可靠性



1. 模块接口。主要检查下述几个方面：
  - 参数的数目、次序、属性或单位系统与变元是否一致；
  - 是否修改了只作输入用的变元；
  - 全局变量的定义和用法在各个模块中是否一致。
2. 局部数据结构：发现局部数据说明、初始化、默认值等方面的错误。
3. 重要的执行通路：选择最有代表性、最可能发现错误的执行通路进行测试。

# 测试重点

4. 出错处理通路。当评价出错处理通路时，应该着重测试下述一些可能发生的错误：
- 对错误的描述是难以理解的；
  - 记下的错误与实际遇到的错误不同；
  - 在对错误进行处理之前，错误条件已经引起系统干预；
  - 对错误的处理不正确；
  - 描述错误的信息不足以帮助确定造成错误的位置。

# 测试重点

## 5. 边界条件。

- 软件常常在它的边界上失效。
- 使用刚好小于、刚好等于和刚好大于最大值或最小值的数据结构、控制量和数据值的测试方案，非常可能发现软件中的错误。

# 代码审查

- 代码审查是一种非常有效的程序验证技术，对于典型的程序来说，可以查出30%到70%的逻辑设计错误和编码错误
- 审查小组的组成
  - 组长
  - 程序的设计者
  - 程序的编写者
  - 程序的测试者

# 代码审查

- 由审查小组对源程序正式进行人工测试。
  - 审查之前，小组成员应该先研究设计说明书，力求理解这个设计。
  - 审查会上倾听程序编写者的解释，并力图发现其中的错误。
  - 对照类似于上一小节中介绍的程序设计常见错误清单，分析审查这个程序。
- 预排：由一个人扮演“测试者”，其他人扮演“计算机”。
- 代码审查的优越性：一次审查会上可以发现许多错误，减少系统验证的总工作量

# 计算机测试

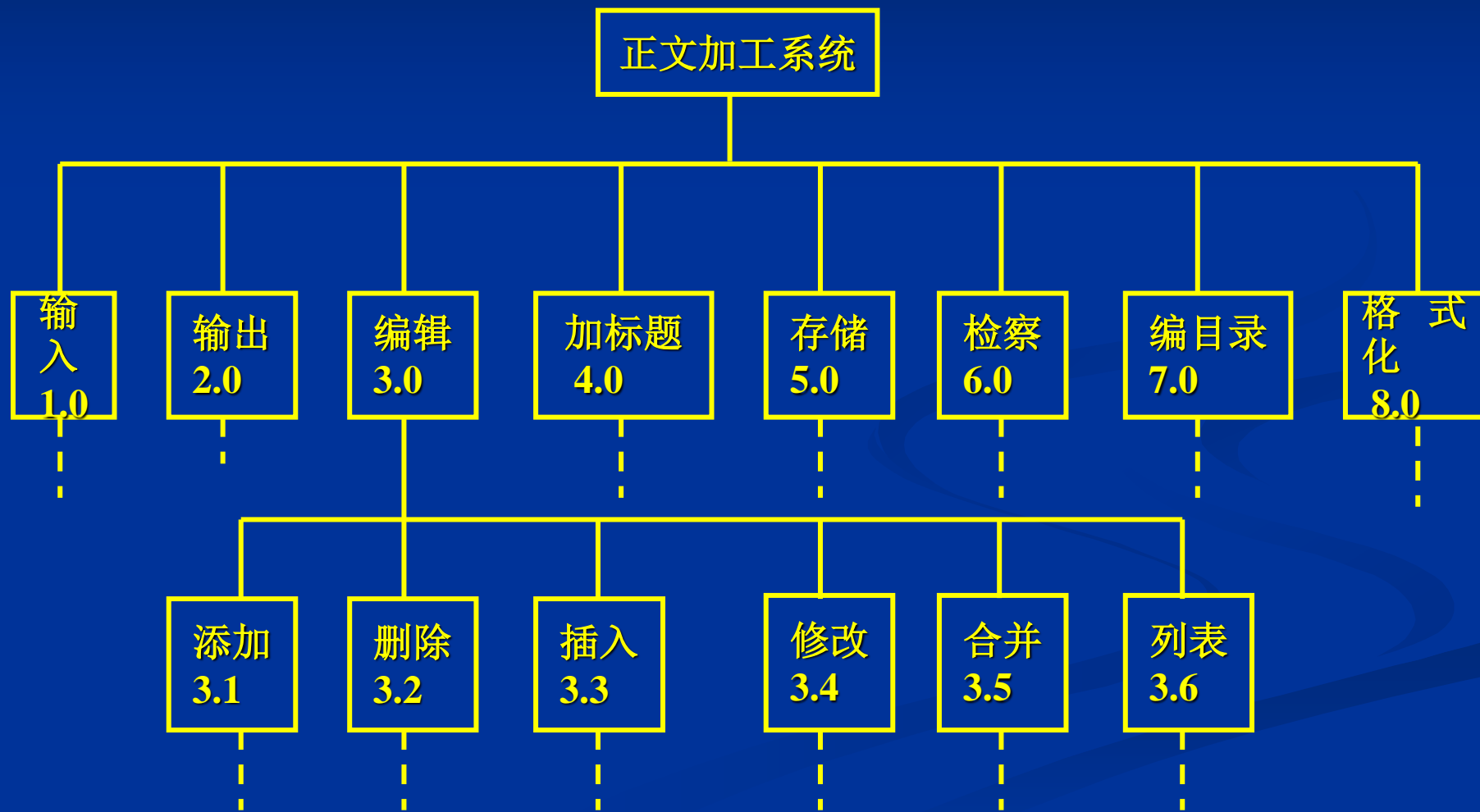
- 需要编写驱动程序（driver）和存根程序（stub）。
- 驱动程序也就是一个“主程序”，它接收测试数据，把这些数据传送给被测试的模块，并且印出有关的结果。
- 存根程序代替被测试的模块所调用的模块。因此存根程序也可以称为“虚拟子程序”。它使用被它代替的模块的接口，可能做最少量的数据操作，印出对入口的检验或操作结果，并且把控制归还给调用它的模块。





# 计算机测试

- 例如针对下图中3.0“编辑”模块进行测试:





# I . TEST STUB(\*测试正文编辑模块用的存根程序\*)

初始化;

输出信息“进入了正文编辑程序”;

输出“输入的控制信息是” CFUNCT;

输出缓冲区中的字符串;

IF CFUNCT=CHANGE

THEN

把缓冲区中第二个字改为\*\*\*

ELSE

在缓冲区的尾部加???

END IF;

输出缓冲区中的新字符串;

END TEST STUB

## II. TEST DRIVER(\*测试正文编辑模块用的驱动程序\*)

说明长度为2500个字符的一个缓冲区；

把CFUNCT置为希望测试的状态；

输入字符串；

调用正文编辑模块；

停止或再次初启；

**END TEST DRIVER**

# 计算机测试

- 编写驱动程序和存根程序是软件测试活动中不可避免的一种工作开销。
- 为减少这种开销，可以采用渐增式测试方法，在集成测试的过程中同时完成对单一模块的详尽测试。
- 模块内聚度高可以简化单元测试过程。