第二十一章
软件测试故障模型

上一章回顾

- 业务测试的内容
- 业务测试验证点

课堂提问

- 业务测试的概念
- 业务测试需要关注什么
- 业务测试用例的注意事项

本章学习目标

- 熟练掌握二十一种故障模型

本章学习方法

- 运用

内容进度

- 故障模型
- 功能性测试的测试方法
  - 用户接口输入测试
  - 用户接口输出测试

故障模型

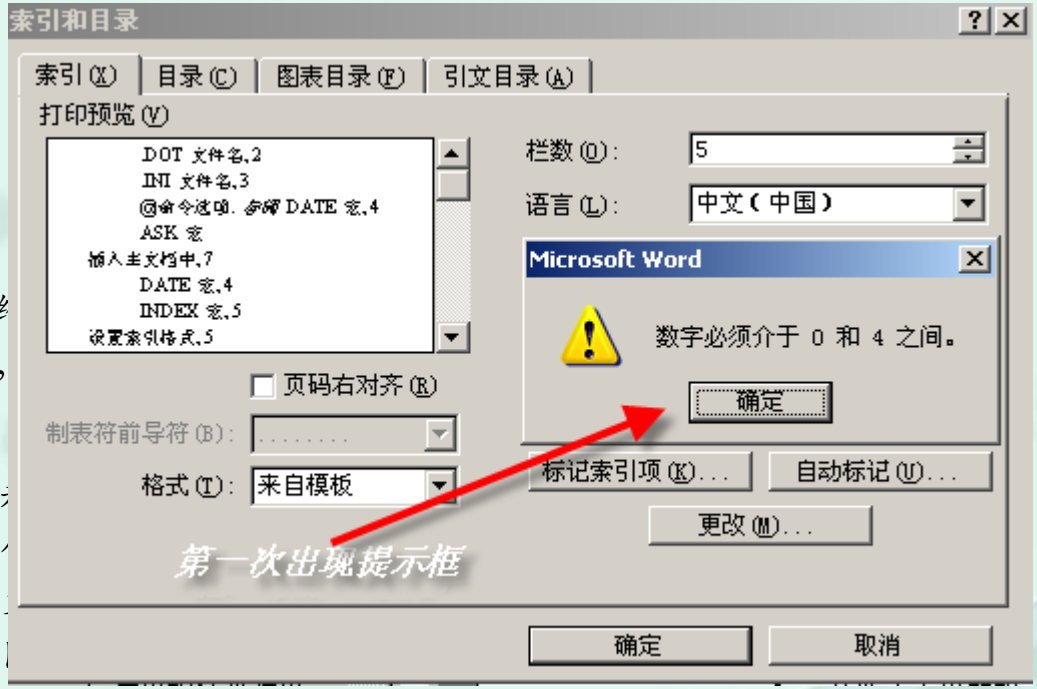
- 故障模型概念
  - 设计测试用例时有太多的单个输入变量、多个输入变量的组合，优秀的软件测试人员不会依靠运气，他们有着丰富的经验和直觉，可以从中找到哪些是要进行测试的，哪些不需要测试，哪些操作可能会引起软件失效。我们把这些测试人员的经验和直觉尽量归纳和固化，形成一些故障模型（Fault Model）。
  - 为软件测试工程师敏锐发现缺陷提供帮助

内容进度

- 故障模型
- 功能性测试的测试方法
  - 用户接口输入测试
  - 用户接口输出测试

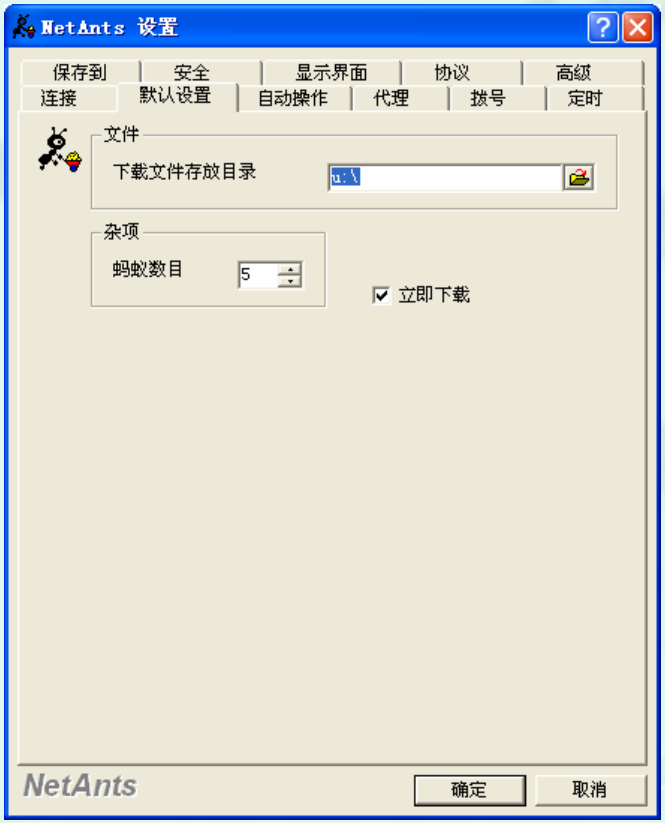
方法1：输入非法数据

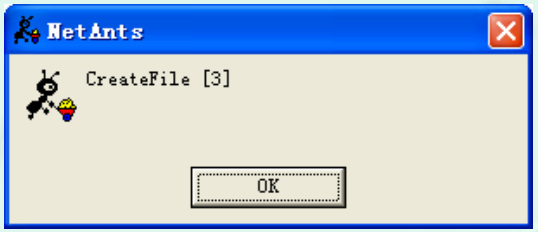
- 案例演示
- 原理分析
  - 处理非法输入的方法
    - 输入时过滤非法数据，
    - 程序内部捕获错误信息，
- 如何发现这类错误
  - 举例：假设“软件测试工
    个文件进行保存，保存文
    则保存的该工程师信息的
    意工程师姓名输入的隐含
    - 输入非法类型：文件名中不能包括的9个非法字符，系统保留字等；
    - 输入超长字符：255个字符；
- 注意
  - 检查错误信息，保证正确、易懂！
  - 举例： 错误信息：Error 5-unkown data!



方法1：输入非法数据

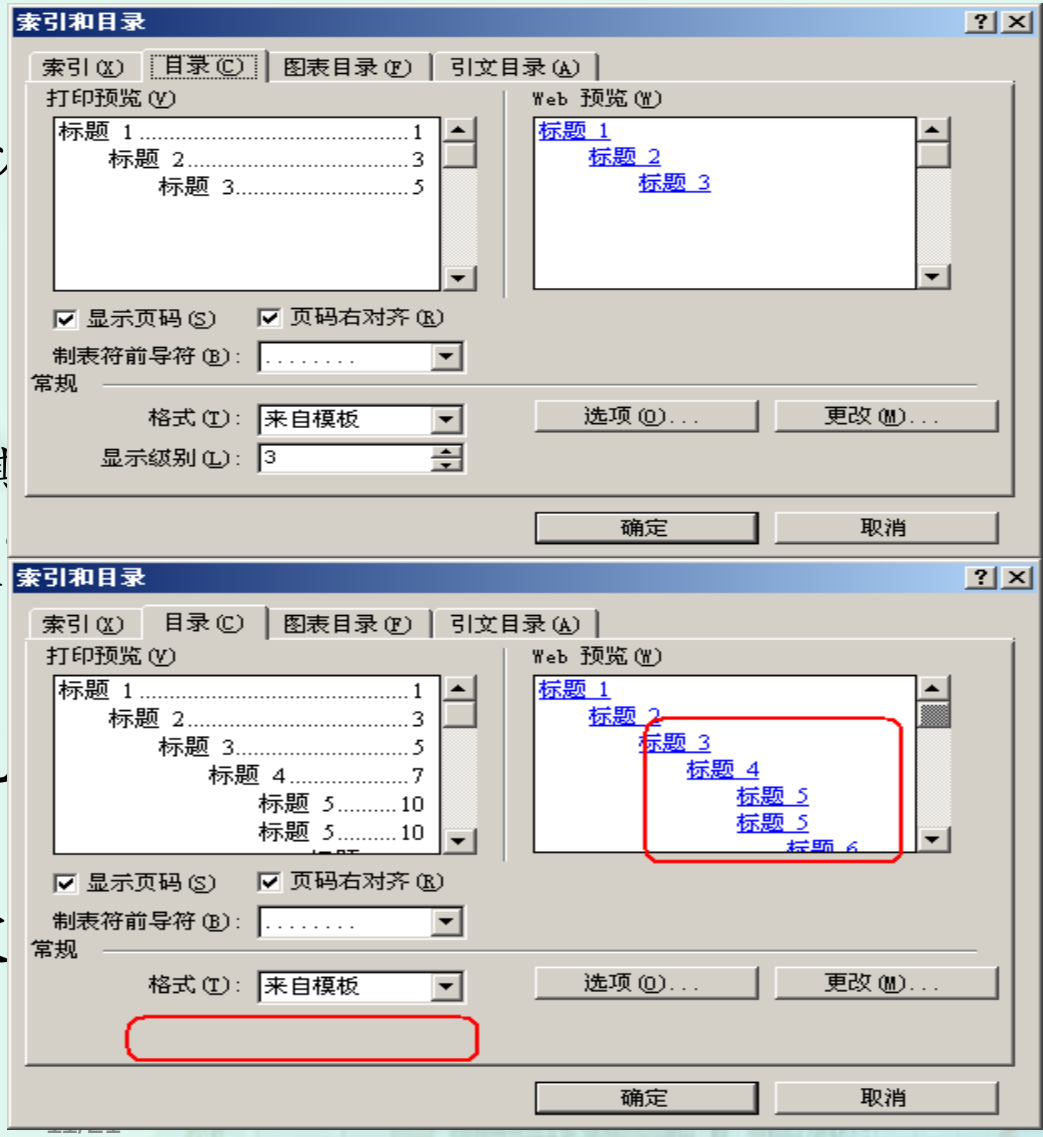
- 实战演练





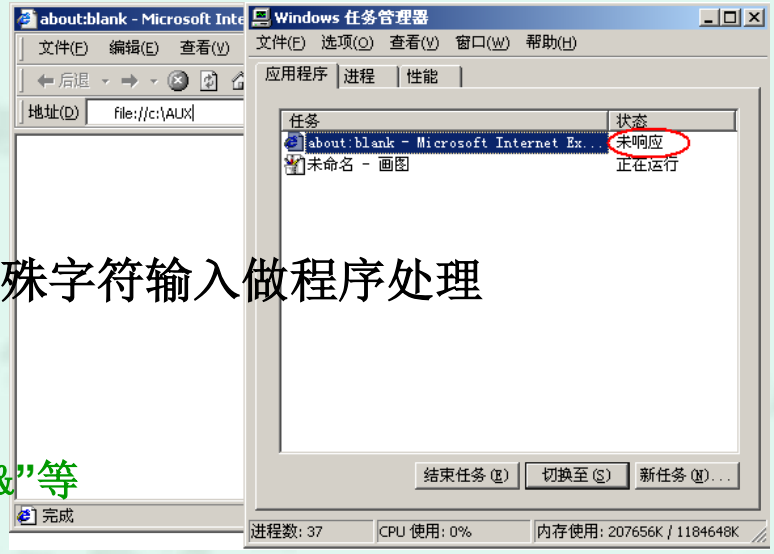
方法2：输入默认值

- 案例演示
  - 环境：Word2000（可以
- 此类缺陷产生原因
  - 定义变量时未赋初值
  - 赋初值不正确
  - 再次赋初值后对程序其
- 如何发现这类错误
- 测试方法小结
  - 全面理解需求规求
  - 同时深刻理解被
- 实战演练

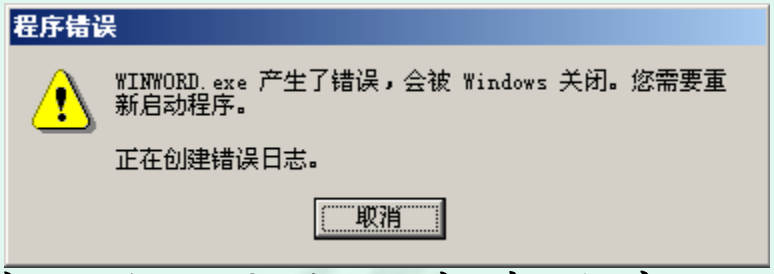


方法3：输入特殊字符集

- 案例演示
  - 环境：Win2000、IE5
- 此类缺陷产生原因
  - 特殊字符处理问题，没有对特殊字符输入做程序处理
  - 注意系统保留字符串
  - 注意应用程序处理特殊字符
    - C语言中的“\n”、“++”、“&”等
- 如何发现这类错误?
- 测试方法小结
- 实战演练



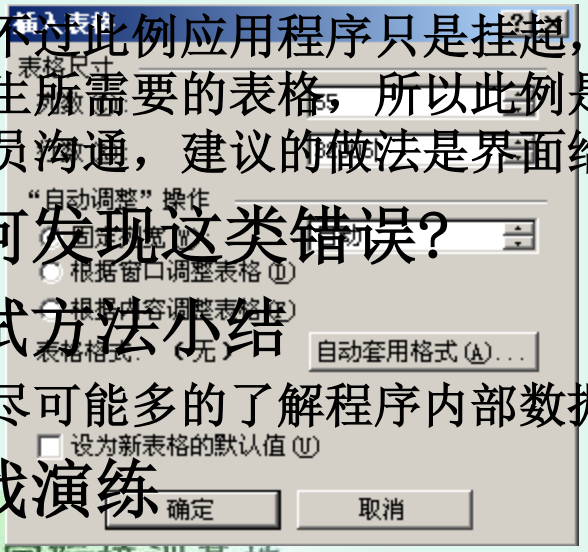
方法4：输入使缓冲区溢出的数据

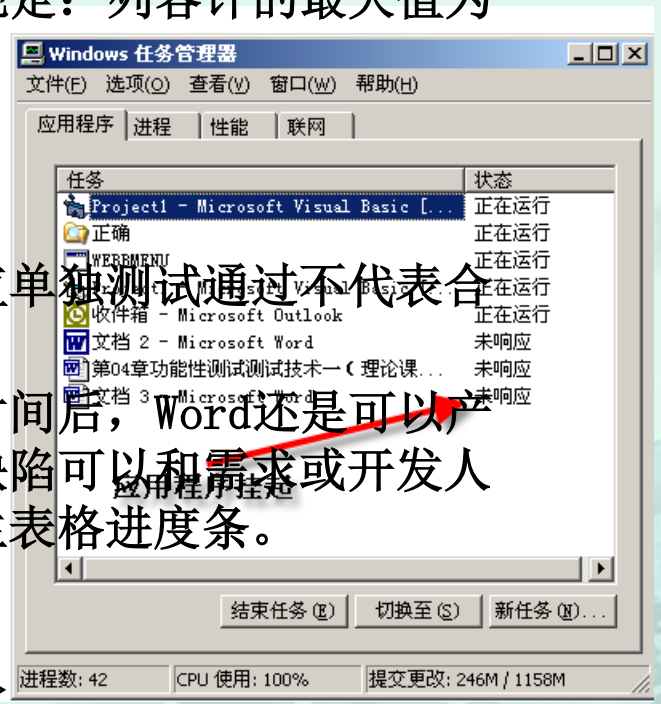


- 案例演示
  - 环境：Win2000、Word2000
- 此类缺陷产生原因
  - 输入的数据未经检查，超过该值固定大小内存缓冲区，影响其他内存单元，严重的引起程序关闭。
- 如何发现这类错误
  - 获得需求（包括详细设计说明），输入最大字符串和超过最大字符串要求的输入数据
- 测试方法小结
  - 加强和开发人员沟通，了解没有写到需求或设计文档中的变量范围
- 实战演练

方法5：输入产生错误的合法数据组合

- 案例演示
  - 在Word中插入表格，需求规格说明书中规定：列容许的最大值为63，行容许的最大值为32767
  - 输入：列=55，行=32005，结果？
- 此类缺陷产生原因
  - 测试多个输入值的组合，每个合法输入值单独测试通过不代表合法输入值的组合测试也能通过。
  - 不过此例应用程序只是挂起，等待一段时间后，Word还是可以产生所需要的表格，所以此例是否确定为缺陷可以和需求或开发人员沟通，建议的做法是界面给出产生产生表格进度条。
- 如何发现这类错误？
- 测试方法小结
  - 尽可能多的了解程序内部数据结构，多与开发人员沟通。
- 实战演练





用户接口输入测试小结

- 输入非法数据
- 输入默认值
- 输入特殊字符集
- 输入使缓冲区溢出的数据
- 输入产生错误的合法数据组合

内容进度

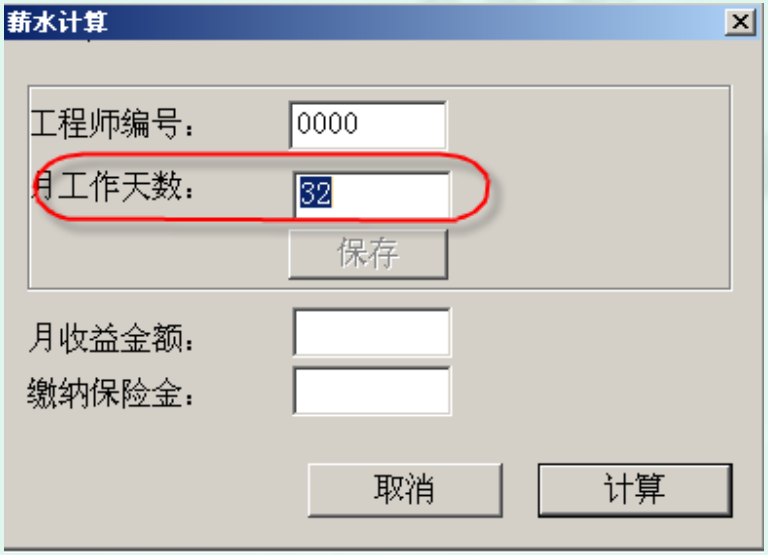
- 故障模型
- 功能性测试的测试方法
  - 用户接口输入测试
  - 用户接口输出测试

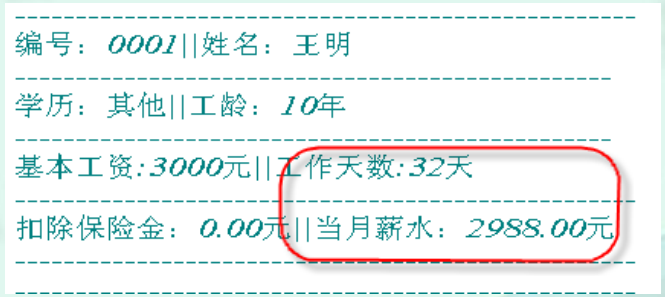
方法6：同一个输入产生各种可能输出

- 案例分析
  - 输入：一个电话打来
  - 输出：
    - 状态一：如果此电话正在使用，则打来电话的人听到的声音应该是占线的提示音。
    - 状态二：如果此时电话未使用，则打来电话的人听到的声音应该是等待接听的提示音。
- 缺陷产生原因
  - 开发人员可能没有判断当前所处状态，就想当然的给出了输出。
- 如何发现这类错误
  - 熟悉被测软件业务知识，阅读各种程序文档，明确输入可能产生的输出。

方法7：产生不符合业务规则的无效输出

- 案例演示
- 缺陷产生原因
  - 程序开发人员对业务了解不深刻
- 如何发现这类错误?





用户接口输出测试小结

- 产生同一输入的各种可能输出
- 强制产生不符合业务规则的无效输出

内容进度

- 用户接口输出测试
- 数据结构的测试

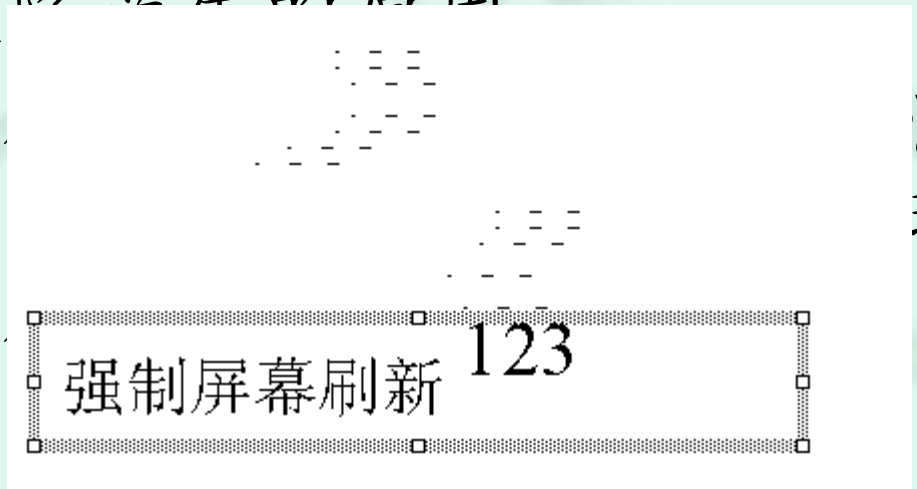
方法8：输出属性修改后的结果

- 案例演示
  - 输出具有可修改的属性
  - 本案例是否为缺陷可以根据需求做进一步判断
- 缺陷产生的原因
  - 开发人员在创建对象时设立了初始值，但当用户修改输出对象属性，开发人员编写的对应代码没有考虑这些属性值的修改对其他变量的影响。
- 如何发现这类错误及测试方法小结?

攻击8：强制通过输出修改属性

方法9：检查屏幕刷新

- 案例演示
- 缺陷产生的原因
  - 慢；刷新频率
    现的现象。
  -
- 如
- 测试方法小结
  - 注意增加、删除和移动屏幕上的对象能发现类似的缺陷



用户接口输出测试小结

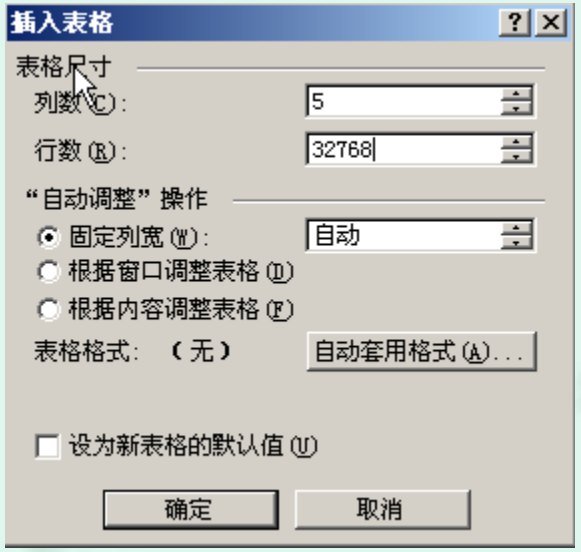
- 产生同一输入的各种可能输出
- 强制产生不符合业务规则的无效输出
- 强制通过输出修改属性
- 检查屏幕刷新

内容进度

- 用户接口输出测试
- 数据结构的测试

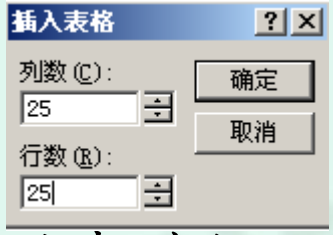
方法10：数据结构溢出

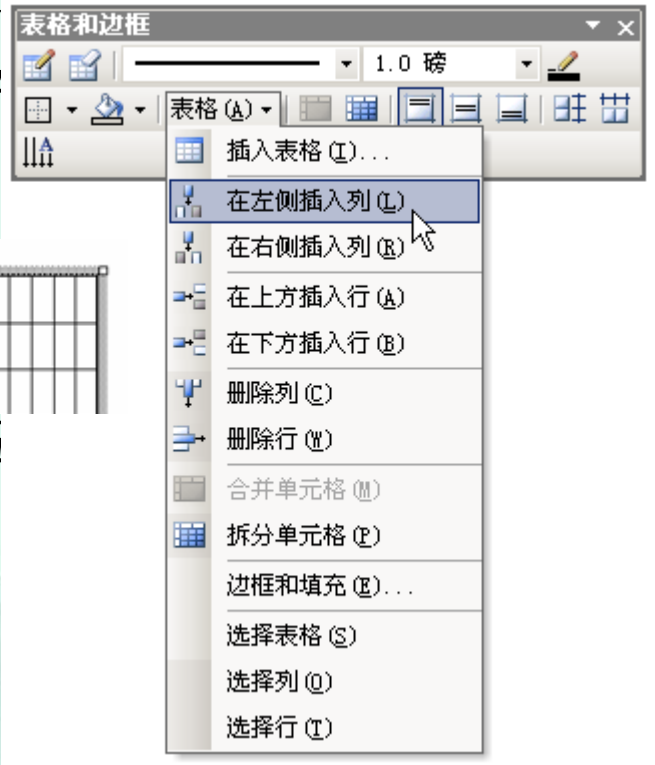
- 案例演示
- 缺陷产生的原因
  - 数据结构限制
  - 内存限制
  - 硬盘限制
- 如何发现这类错误
  - 上溢
  - 下溢
- 测试方法小结
  - 数组



方法11：数据结构不符合约束

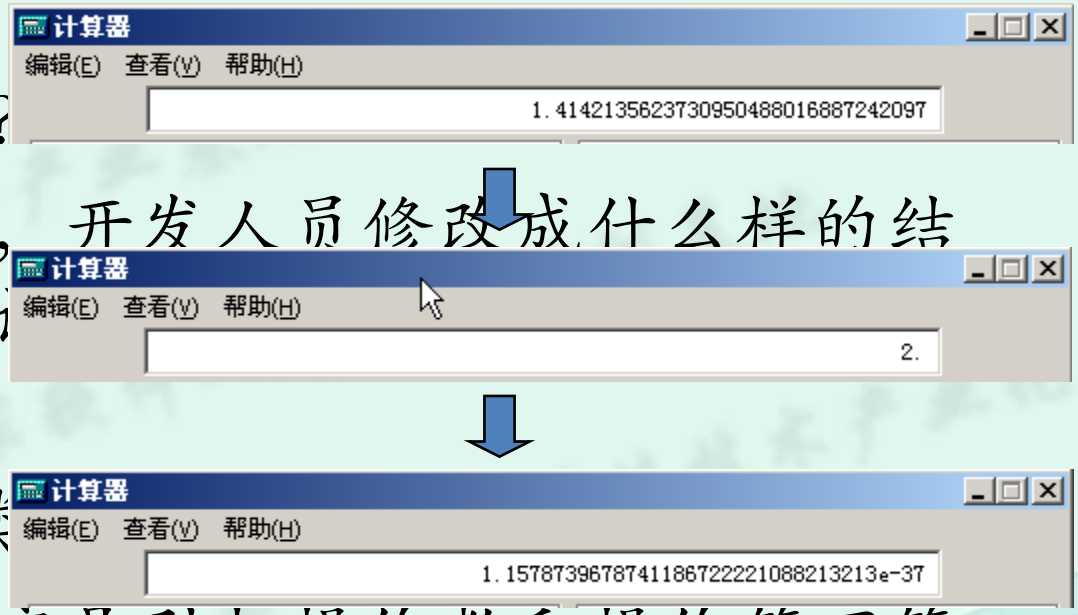
- 案例演示
- 缺陷产生的原因
  - 在建立数据项时对数据属性的约束进行了检查，而修改数据项的代码未做约束
- 如何发现这类错误
  - 修改属性判断是否进行约束
- 测试方法小结
  - 了解内部数据结构约束，尝试进行测试。





方法12：操作数和操作符不符

- 案例演示
  - 是否是缺陷?
  - 如果是缺陷，开发人员修改成什么样的结果你作为测试人员，如何修复。

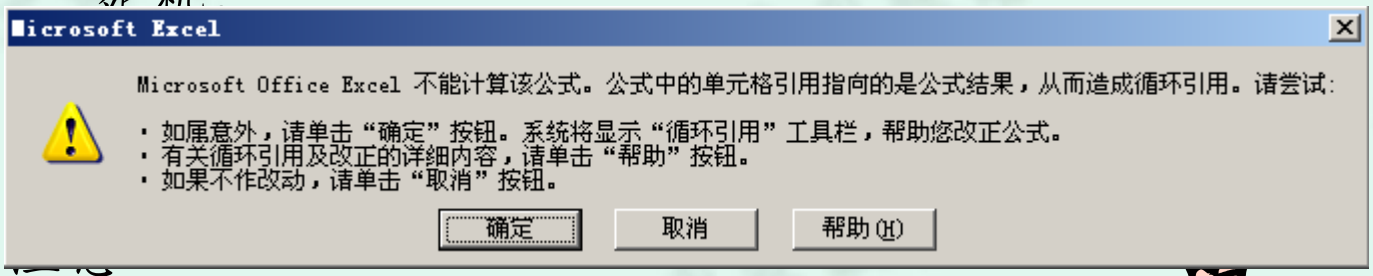


- 如何发现这类
  - 找到程序中容易引起操作数和操作符不符的计算、表达式等。
- 实战演练

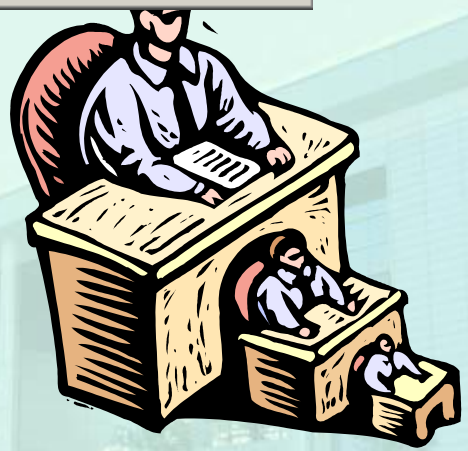
方法13：函数递归调用

- 案例演示
  - Excel案例演示
- 缺陷产生的原因
  - 函数递归调用，没有合[illegible]能会导致系统死机

	A	B	C
1	姓名	成绩	
2	张三	76	
3	李四	54	
4	王五	83	
5	赵六	61	
6		=SUM(B2:B6)	
7			



- [illegible]
  - 教材中的例子需要的环境
    - Win2000，Word2000



方法14：计算结果溢出

- 案例分析

```
sum = sum + value[i]
```

  - 如果value[0]=32700，value[1]=70，则？
-
```
const count 2
main()
{
int sum,value[count];
sum=0;
for (i =0 ; i <count;++i)
{
   sum = sum + value[i];
}
}
```
里
直
-
  - 输入非法值，强制数据产生溢出，观察程序的处理情况。

方法15：数据共享或关联功能出错

- 案例演示

时，

功能

测试方法小结

数据结构的测试小结

- 数据结构溢出
- 数据结构不符合约束
- 操作数与操作符不符
- 递归调用自身
- 计算结果溢出
- 数据共享或关联功能计算出错

本章学习目标

- 文件系统的测试
- 软件的故障模型

内容进度

- 文件系统的测试
- 软件的故障模型

方法16：使文件系统超载

- 案例
  - 假设“软件测试工程师管理系统”要保存10000个工程师信息，则保存时engineer.txt文件会有20M大小，如果此时磁盘只有10M可用空间了，“软件测试工程师管理系统”会如何动作呢?
- 此类缺陷产生的原因
  - 开发人员忽略了CreateFile、WriteFile等与操作系统交互的API错误代码检查。
- 如何发现这类问题?
  - 使用工具Canned Heat，模拟文件系统负载。

方法17：使介质忙或不可用

- 案例演示

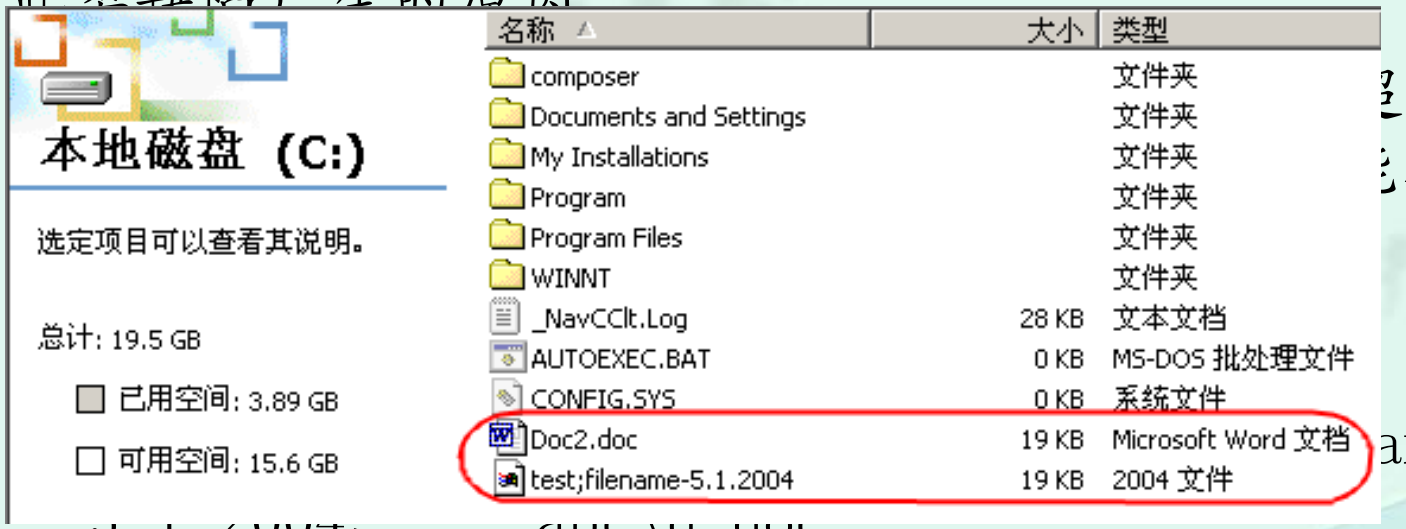
- 此类缺陷产生的原因
  - 开发人员没有考虑介质忙或者不可用的情况，未对此种情况做出处理。
- 如何发现这类问题
  - 使用工具Canned Heat，模拟介质忙或不可用的情况。

方法18：介质损坏

- 案例分析
- 缺陷产生的原因
  - 损坏的介质可能会是操作系统传回错误代码，这些错误代码没有在应用程序中编程处理。
  - 操作系统不能检测出所有的这些错误。
- 如何发现这类问题
  - 一般软件，不必考虑介质损坏问题。一般用在操作系统、设备驱动程序/控制器以及以安全为主的应用程序才会考虑此类测试。
  - 例如测试实现RAID5技术的软件，则需要模拟一块硬盘坏了之后，换一个硬盘，数据是否可以恢复。

方法19：使用不合法的文件名

- 案例演示
  - 环境：Win2000，Word2000
- 此类缺陷产生的原因



过255
被使用。

ame-

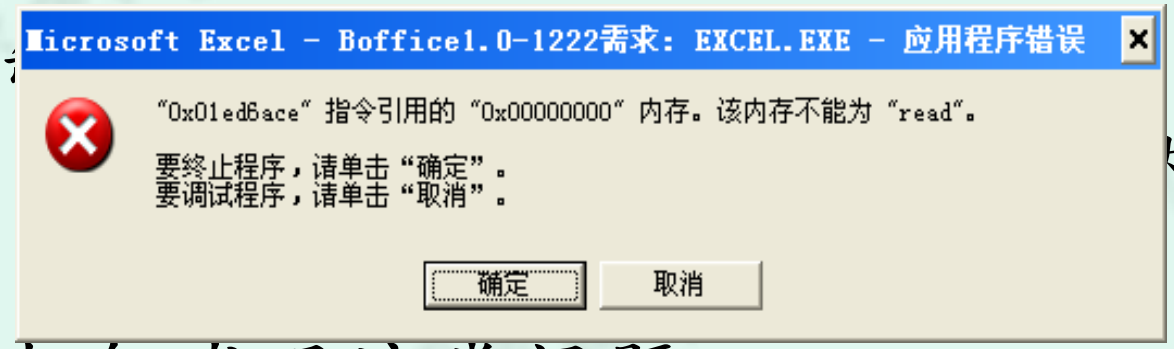
- 测试方法小结
  - 熟记文件名命名规则

方法20：更改文件访问权限

- 案例演示

- 此类缺陷产生的原因
  - 特别需要注意：不同的用户对相同文件具有不同的访问权限，需要考虑登录同一台机器的多个用户操作相同文件的权限问题。
- 如何发现这类问题?

方法21：文件内容受损

- 案例演示
- 

内容，对验证
- 如何发现这类问题
  - 手工损坏文件测试
  - 使用工具，模拟CRC（循环冗余校验）错误

文件系统的测试小结

- 使文件系统超载
- 使介质忙或不可用
- 介质损坏
- 使用不合法的文件名
- 更改文件访问权限
- 文件内容受损

总结

- 二十种故障模型的来源
- 理解二十一种故障模型
- 请预习第二十一章