

## 第一章 4、5、6、7:

### 4: 什么是软件工程?软件工程的基本原理有哪些?

答: 概述:

软件工程是指导计算机软件开发和维护的一门工程学科,采用工程的概念、原理、技术和方法对软件进行开发和维护。把经过时间检验的管理技术和能得到的最好的技术方法结合起来,以经济地开发出高质量的软件并有效的进行维护。

基本原理:

- 1.用分阶段生命周期计划严格要求。
- 2.坚持进行阶段评审。
- 3.实行严格的产品控制。
- 4.采用现代程序设计技术。
- 5.结果应能清楚地审查。
- 6.开发小组的成员应该少而精。
- 7.承认不断改进软件工程实践的必要性。

### 5: 软件生命周期各阶段是如何划分的?各阶段的基本任务是什么?

答: 软件生命周期分为问题定义、可行性研究、需求分析、开发阶段、维护这 5 个阶段。各个阶段的主要任务是如下:

#### 1、问题定义

要求系统分析员与用户进行交流,弄清“用户需要计算机解决什么问题”然后提出关于“系统目标与范围的说明”,提交用户审查和确认。

#### 2、可行性研究

一方面在于把待开发的系统的目标以明确的语言描述出来,另一方面从经济、技术、法律等多方面进行可行性分析。

#### 3、需求分析

弄清用户对软件系统的全部需求,编写需求规格说明书和初步的用户手册,提交评审。

#### 4、开发阶段

开发阶段由四个阶段组成:概要设计、详细设计、实现、测试

#### 5、维护

维护包括四个方面:

(1) 改正性维护:在软件交付使用后,由于开发测试时的不彻底、不完全、必然会有一部分隐藏的错误被带到运行阶段,这些隐藏的错误在某些特定的使用环境下就会暴露。

(2) 适应性维护:是为适应环境的变化而修改软件的活动。

(3) 完善性维护:是根据用户在使用过程中提出的一些建设性意见而进行的维护活动。

(4) 预防性维护:是为了进一步改善软件系统的可维护性和可靠性,并为以后的改进奠定基础。

### 6: 常用的软件开发模型有哪些?各有什么特点?

答:

瀑布模型:

1.里程碑或基线驱动。2.过程逆转性很差或者说不可逆转。逆转可能会延误工期,增加成本,造成损失

1.开发阶段清晰,便于评审、审计、跟踪、管理和控制。

1.不可逆或很难可逆。2.问题会积累,错误会传递发散扩大,导致成本和质量失控。

- 1.在开发时间内需求不变化或很少变化。
- 2.分析设计人员对此领域非常熟悉。
- 3.低风险项目。
- 4.用户使用环境稳定（如系统软件，工具软件）

#### 快速原型模型

- 1.容易适应需求的变化

1.克服瀑布模型的缺点，减少由于软件需求不明确带来的开发风险。

1.所使用的开发工具和技术不一定符合主流的发展；快速建立起来的系统架构加上连续的修改可能会导致产品质量低下。

- 1.需求不明确或复杂系统。
- 2.用户无法自主提出应用需求。

#### 增量模型：

- 1.软件由一系列增量构件组成

1.人员分配灵活，刚开始不用投入大量人力资源。如果核心产品很受欢迎，则可增加人力实现下一个增量。当配备人员不能在设定的时间内完成产品时，它提供了一种先推进核心产品的途径，这样即可先发部分功能给客户，对客户起到镇静剂作用。

2.增量能够有计划的管理技术风险。

1.由于各个构件是逐渐并入已有的软件体系结构中，所以加入构件必须不破坏已构造好的系统部分，这需要软件具备开放式的体系结构。

2.在开发过程中，需求变化是不可避免的，增量的灵活性可以使其适应这种变化的能力大大优于瀑布和快速原型模型，但也容易退化为边改边做模型，从而使软件过程的控制失去整体性。

3.如果增量包之间存在相交的情况且未很好处理，则必须做全盘系统分析，这种模型将功能细化后分别开发的方法较适应于需求经常改变的软件开发过程

- 1.系统容易拆分。
- 2.开发人力比较少。
- 3.特别适用于商业软件（如 QQ，网游）

#### 螺旋模型：

1.是一种周期性的方法进行系统开发。

2.有许多“中间版本”。

3.每个周期都包括需求定义、风险分析、工程实现和评审 4 个阶段

1.设计上灵活，可在项目各个阶段进行变更。

2.以小的分段来构建大型系统，使成本计算变得简单容易。

3.客户始终参与每个阶段的开发，保证了项目不偏离正确方向以及项目的可控性。

1.建设周期长，而软件技术更新比较快，所以经常出现软件开发结束后，和当前的技术水平有了很大的差距，无法满足当前用户需求。

- 1.系统庞大，风险高。
- 2.需求不太明确。

#### 喷泉模型：

1.该模型的各个阶段没有明显的界限，开发人员可以同步进行开发。

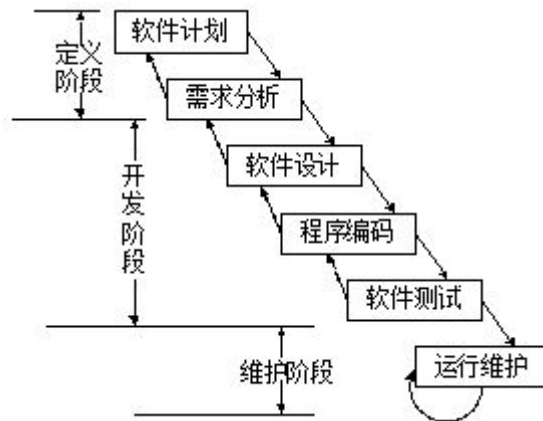
1.提高软件项目开发效率，节省开发时间，适用于面向对象的软件开发过程。

1.由于喷泉模型在各个开发阶段是重叠的，因此在开发过程中需要大量的开发人员，因此不利于项目的管理。

2.要求严格管理文档，使得审核难度加大，尤其是面对可能随时加入的各种信息、需求与资料的情况。

- 1.面向对象的软件开发过程

7：请画出传统的软件开发模型，即瀑布模型，并标出 3 个时期。



答：

## 第二章 4：成本/效益分析中成本估算采用哪些方法?成本/效益分析采用哪些方法?

答：

成本估算：

### 1.代码行技术：

代码行技术是比较简单的定量估算方法，它把开发每个软件功能的成本和实现这个功能需要用的源代码行数联系起来。估计出源代码行数以后，用每行代码的平均成本乘以行数就可以确定软件的成本每行代码的平均成本主要取决于软件的复杂程度和工资水平。

### 2.任务分解技术：

这种方法首先把软件开发工程分解为若干个相对独立的任务，再分别估计每个单独的开发任务的成本，最后累加起来得出软件开发工程的总成本估计每个任务的成本时。通常先估计完成该项任务需要用的时间(以人月为单位)，再乘以每人每月的平均工资而得出每个任务的成本。

### 3.自动估计成本技术：

采用自动估计成本的软件工具可以减轻人的劳动，并且使得估计的结果更客观。但是，采用这种技术必须有长期搜集的大量历史数据为基础，并且需要有良好的数据库系统支持。

成本/效益分析：

### 1.货币的时间价值

### 2.投资回收期：

所谓投资回收期就是使累计的经济效益等于最初投资所需要的时间。

### 3.纯收入：

纯收入就是在整个生命周期之内系统的累计经济效益(折合成现在值)与投资之差。这相当于比较投资开发一个软件系统和把钱存在银行中这两种方案的优劣。

### 4.投资回收期：

把资金存入银行或贷给其他企业能够获得利息，通常用年利率衡量利息多少。类似地也可以计算投资回收期，用它衡量投资效益的大小，并且可以把它和年利率相比较，在衡量工程的经济效益时，它是最重要的参考数据。

## 第三章 2、3、4、5：

### 2：数据流图的作用是什么?它包含哪些基本元素?

答：作用：数据流图是用于表示系统逻辑模型的一种工具。它从数据传递和加工的角度，以图形的方式描述数据在系统中流动和处理过程。它表示了系统内部信息的流向以及系统的逻辑

辑处理功能。

元素：数据流、数据存储、加工、数据流的源点和终点。

**3：数据字典的作用是什么?每张卡片上包含哪些基本内容?**

答：作用：数据词典是数据的集合，它对数据流图中的各个元素作完整的定义和说明，是数据流图的补充工具。数据流图和数据词典共同构成系统的逻辑模型，两者缺一不可。

内容：数据流、数据项、数据结构、数据存储、加工逻辑、外部实体。

**4：ER图的作用是什么?它有哪些基本元素?**

答：作用：ER图主要用于描述数据库中表的关系

元素：实体、属性、关系。

**5：什么是结构化分析?它的目的和任务是什么?**

答：定义：结构化分析实质上是一种创建模型的活动。为了开发出复杂的软件系统，系统分析员应该从不同角度抽象出目标系统的特性，使用精确的表示方法构造系统的模型，验证模型是否满足用户对目标系统的需求，并在设计过程中逐渐把和实现有关的细节加进模型中，直至最终用程序实现模型。

任务：结构化设计包括架构设计、接口设计、数据设计和过程设计等任务。

**第四章 2、3、4、5、8、9、10：**

**2：软件工程的基本定理是什么?**

答：基本原理：

- 1.用分阶段的生命周期计划严格管理
- 2.坚持进行阶段评审
- 3.实行严格的产品控制
- 4.采纳现代程序设计技术
- 5.结果应能清楚地审查
- 6.开发小组的人员应少而精
- 7.承认不断改进软件工程实践的必要性

**3：衡量模块独立性的指标是什么?**

答：内聚和耦合

模块的独立程度可以由两个定性标准衡量，这两个标准分别是内聚和耦合。

耦合衡量不同模块彼此间互相依赖（连接）的紧密程度；内聚衡量一个模块内部各个元素彼此结合的紧密程度。

**4：什么是内聚?内聚分几种?哪种内聚最好,哪种最差?**

答：内聚有如下的种类，它们之间的内聚度由弱到强排列如下：

- （1）偶然内聚。模块中的代码无法定义其不同功能的调用。但它使该模块能执行不同的功能，这种模块称为巧合强度模块。
- （2）逻辑内聚。这种模块把几种相关的功能组合在一起，每次被调用时，由传送给模块参数来确定该模块应完成哪一种功能
- （3）时间内聚：把需要同时执行的动作组合在一起形成的模块为时间内聚模块。
- （4）过程内聚：构件或者操作的组合方式是，允许在调用前面的构件或操作之后，马上调

用后面的构件或操作，即使两者之间没有数据进行传递。

(5) 通信内聚:指模块内所有处理元素都在同一个数据结构上操作(有时称之为信息内聚),或者指各处理使用相同的输入数据或者产生相同的输出数据。

(6) 顺序内聚: 指一个模块中各个处理元素都密切相关于同一功能且必须顺序执行,前一功能元素输出就是下一功能元素的输入。

(7) 功能内聚: 这是最强的内聚,指模块内所有元素共同完成一个功能,缺一不可。功能内聚最好;偶然内聚最差。

## 5: 什么是耦合?耦合分几种?哪种耦合最好,哪种最差?

答: 耦合可以分为以下几种,它们之间的耦合度由高到低排列如下:

(1) 内容耦合: 如果发生下列情形,两个模块之间就发生了内容耦合  
一个模块直接访问另一个模块的内部数据  
一个模块不通过正常入口转到另一模块内部;  
两个模块有一部分程序代码重叠(只可能出现在汇编语言中);  
一个模块有多个入口。

(2) 公共耦合: 若一组模块都访问同一个公共数据环境,则它们之间的耦合就称为公共耦合。公共的数据环境可以是全局数据结构、共享的通信区、内存的公共覆盖区等。

(3) 外部耦合: 一组模块都访问同一全局简单变量而不是同一全局数据结构,而且不是通过参数表传递该全局变量的信息,则称之为外部耦合。

(4) 控制耦合: 如果一个模块通过传送开关、标志、名字等控制信息,明显地控制选择另一模块的功能,就是控制耦合

(5) 标记耦合: 一组模块通过参数表传递记录信息,就是标记耦合。这个记录是某一数据结构的子结构,而不是简单变量。其实传递的是这个数据结构的地址;也就是地址传递。

(6) 数据耦合: 指两个模块之间有调用关系,传递的是简单的数据值,一个模块访问另一个模块时,彼此之间是通过简单数据参数(不是控制参数、公共数据结构或外部变量)来交换输入、输出信息的,相当于高级语言的值传递。

(7) 非直接耦合: 两个模块之间没有直接关系,它们之间的联系完全是通过主模块的控制和调用来实现的。

内容耦合最好;非直接耦合最差。

## 8: 总体设计采用哪些图介工具?它们有什么区别?

答:

1.层次图: 描述层次结构。

2.HIPO 图: 在层次图的基础上,把图中除了顶层的方框外都加上编号\*\*\*\*基本形式: 输入,处理,输出。

3.结构图(SC): 表达程序结构图形的表示方法,反映程序模块间的层次关系和联系。

成分: 模块,模块间调用关系,通信,辅助控制符号。

结构图的四中类型: 传入,传出,变换,协调

## 9: 结构化设计的目的、任务和步骤是什么?

答: 结构化设计的目的: 使程序的结构尽可能反映要解决的问题的结构。

结构化设计的任务: 把需求分析得到的数据流图 DFD 等变换为系统结构图(SC)。

基本步骤: 分为概要设计和详细设计两个阶段。

10: 什么是变换分析?什么是事务分析?如何将 DFD 图映射为与之相应的初始 SC 图?

答: 变换分析设计是一个顺序结构, 由输入、变换和输出三部分组成, 其工作过程有 3 步: 取得数据、变换数据和给出数据。

事务分析设计是将它的输入流分离成许多发散的数据流, 形成许多加工路径, 并根据输入的值选择其中一个路径来执行。

变换分析主要是将变换型的 DFD 图映射为与之相应的初始 SC 图。采用以下几步来完成:

(1) 在 DFD 图上区分输入、变换中心和输出 3 个部分, 标出它们之间的分界。

(2) 画出初始的 SC 图的框架, 这一框架由顶层的主控模块和供它调用的输入、变换处理和输出三个分支框组成。

(3) 对输入、变换处理和输出这三个分支进行自顶向下的分解, 画出各个分支所需要的全部模块。

(4) 将初始的 SC 图的框架和各个分支所需要的全部模块合起来, 就构成了初始的 SC 图。

事务分析主要是将事务型的 DFD 图映射为与之相应的初始 SC 图。采用以下几步来完成:

(1) 在 DFD 图上划分接收、事务中心和发送 3 个部分, 标出它们之间的分界。

(2) 画出初始的 SC 图的框架, 将上述 3 个部分映射为事务控制模块、接收分支和动作发送分支。

(3) 分解和细化接收分支和发送分支, 画出各个分支所需要的全部模块。

(4) 将初始的 SC 图的框架和各个分支所需要的全部模块合起来, 就构成了初始的 SC 图。

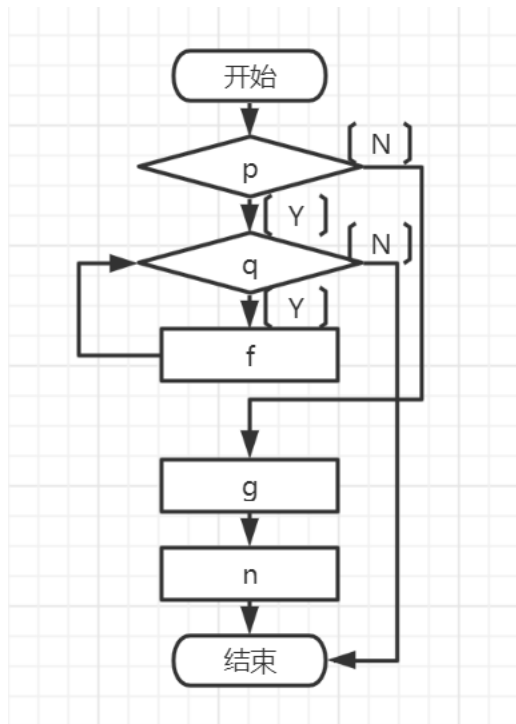
第五章 ppt 最后 1、2:

1.START

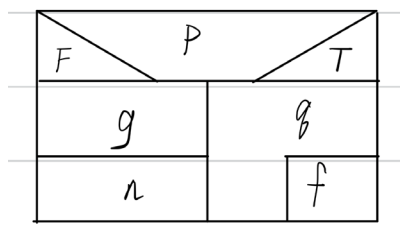
```
IF p THEN
    WHILE q DO
        f;
    ENDDO
ELSE
    BLOCK
        g;
        n;
    ENDBLOCK
ENDIF
STOP
```

答:

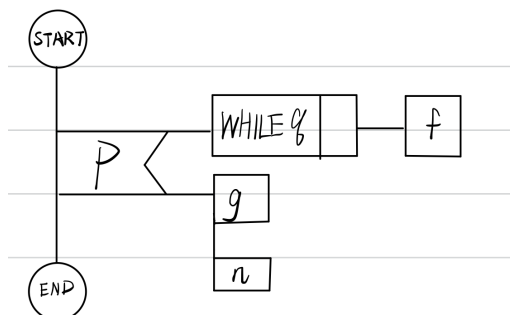
流程图:



NS 图:



PAD 图:



2.START

FOR I=1 TO N

INPUT F(I);

ENDFOR

k=0;

DO WHILE k<N

A(k)=0;

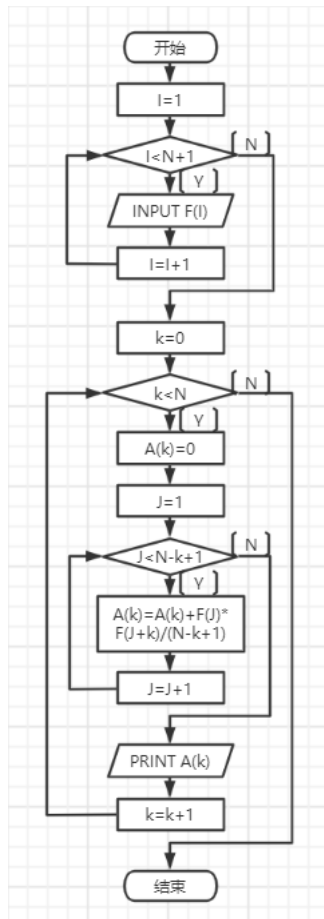
```

FOR J=1 TO N-k
    A(k)=A(k)+F(J)*F(J+k)/(N-k+1);
ENDFOR
PRINT A(k);
k=k+1;
ENDDO

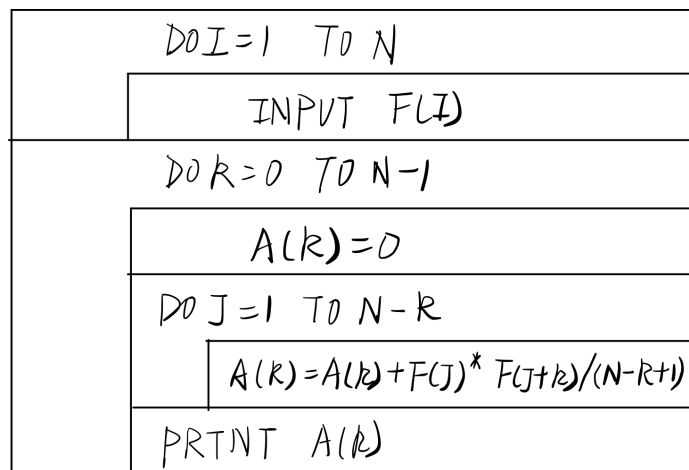
```

答：

流程图：

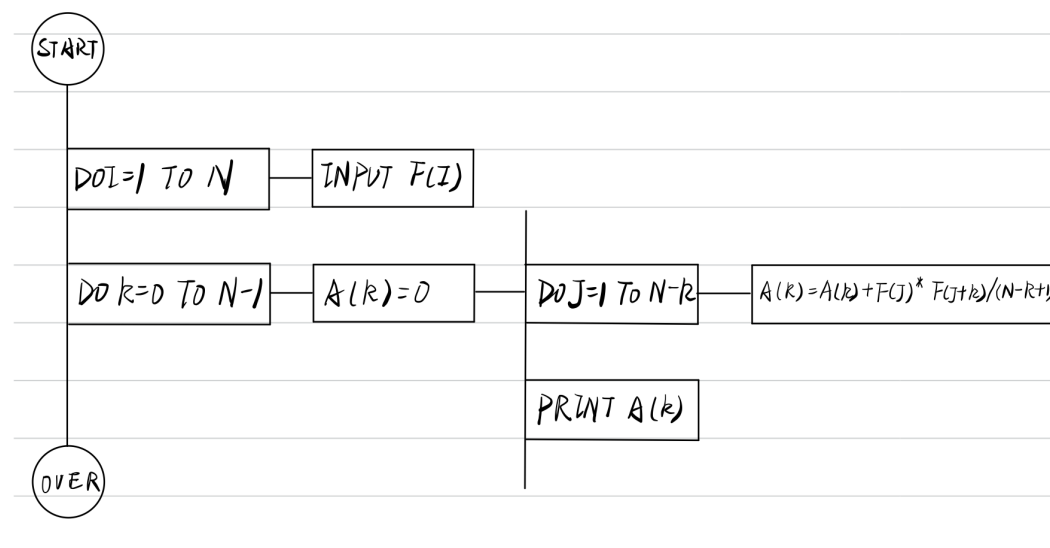


NS 图：





PAD 图:



第六章 1、2（上）、6:

1:

答: 面向对象方法学的出发点和基本原则, 是尽可能模拟人类习惯的思维方式, 使开发软件的方法与过程尽可能接近人类认识世界解决问题的方法与过程, 也就是使描述问题的问题空间(也称为问题域)与实现解法的解空间(也称为求解域)在结构上尽可能一致。

优点: (1)与人类习惯的思维方法一致(2)稳定性好(3)可重用性好(4)可维护性好

2（上）：什么是对象？

答: 对象是对问题域中某个实体的抽象, 由于客观世界中的实体通常都既具有静态的属性, 又具有动态的行为, 因此, 面向对象方法学中的对象是由描述该对象属性的数据以及可以对这些数据施加的所有操作封装在一起构成的统一体。

6: 什么是对象模型？什么是动态模型？什么是功能模型？3 种模型之间存在怎样的关系？

答: 对象模型表示静态的、结构化的系统的“数据”性质。它是对模拟客观世界实体的对象以及对象彼此间的关系的映射, 描述了系统的静态结构。

动态模型表示瞬时的、行为化的系统的“控制”性质, 它规定了对象模型中的对象的合法变化序列。

功能模型表示变化的系统的“功能”性质, 它指明了系统应该“做什么”, 因此更直接地反映了用户对目标系统的需求。

3 种模型之间的关系如下:

(1) 针对每个类建立的动态模型, 描述了类实例的生命周期或运行周期。

(2) 动态模型中的状态转换驱使行为发生, 这些行为在数据流图中被映射成处理, 同时与对象模型中的服务相对应。

(3) 功能模型中的处理对应于对象模型中的类所提供的服务。通常, 复杂的处理对应于复杂对象提供的服务, 简单的处理对应于更基本的对象提供的服务。一个处理可能对应多个服务, 一个服务也可能对应多个处理。

- (4) 功能模型(数据流图)中的数据存储, 以及数据的源点/终点, 通常是对象模型中的对象。功能模型中的数据流, 往往是对象模型中对象的属性值, 也可能是整个对象。
- (5)功能模型中的处理(或用例)可能产生动态模型中的事件。

## 第十章 2、3、4、10:

### 2: 什么是白盒测试? 采用哪些覆盖标准?

答: 如果产品内部活动方式已经知道, 可以测试它的内部活动是否符合设计要求。将被测程序视为一个透明的盒子, 根据被测程序的内部结构来设计测试用例。

白盒测试采用逻辑覆盖测试法和路径测试法, 逻辑覆盖有 5 种覆盖标准:

- ①语句覆盖
- ②判定覆盖
- ③条件覆盖
- ④判定-条件覆盖
- ⑤条件组合覆盖

### 3: 什么是黑盒测试, 它采用哪些测试技术?

答: 黑盒测试又称为“功能测试”, 是将测试对象看做一个黑盒, 在并不考虑软件产品的内部结构和处理过程的基础上对软件产品进行功能测试。黑盒测试注重软件产品的“功能性需求”。

黑盒测试常用的技术有: 边界值、等价类、错误推测法、功能分解法、因果图、判定表、正交试验法、场景法。

### 4: 软件测试的步骤是什么?这些步骤和开发哪些阶段对应?

答: 测试过程按 4 个步骤进行, 即单元测试、集成测试、系统测试及验收测试。

单元测试和编码对应, 集成测试和详细设计对应, 系统测试和概要设计对应, 验收测试和需求分析对应。

10: 某报表处理系统要求用户输入处理报表的日期, 日期限制在 2003 年 1 月至 2008 年 12 月, 即系统只能对该段期间内的报表进行处理, 如日期不在此范围内, 则显示输入错误信息。系统日期规定由年、月的 6 位数字字符组成, 前 4 位代表年, 后 2 位代表月。现要求用黑盒测试法中的边界值分析法设计测试用例。

| 输入条件       | 测试用例说明           | 测试用例    | 预期结果 | 选取理由     |
|------------|------------------|---------|------|----------|
| 输入日期的长度及类型 | 1位数字字符           | 5       | 显示出错 | 仅有1个合法字符 |
|            | 5位数字字符           | 20065   | 显示出错 | 比有效长度少1  |
|            | 6位数字字符           | 200611  | 输入有效 | 类型及长度均有效 |
|            | 7位数字字符           | 2003005 | 显示出错 | 比有效长度多1  |
|            | 有1个非数字字符         | 2006.5  | 显示出错 | 只有1个非法字符 |
|            | 全部是非数字字符         | MAYABC  | 显示出错 | 全部是非法字符  |
| 日期范围       | 在有效范围边界上<br>选取数据 | 200101  | 输入有效 | 最小日期     |
|            |                  | 200812  | 输入有效 | 最大日期     |
|            |                  | 200100  | 显示出错 | 刚好小于最小日期 |
|            |                  | 200813  | 显示出错 | 刚好大于最大日期 |
| 月份范围       | 月份为1月            | 200801  | 输入有效 | 最小月份     |
|            | 月份为12月           | 200112  | 输入有效 | 最大月份     |
|            | 月份<1             | 200800  | 显示出错 | 刚好小于最小月份 |
|            | 月份>12            | 200113  | 显示出错 | 刚好大于最大月份 |

答:

## 第十一章 1、4:

### 1: 软件维护活动包括哪几类?

答：错性维护（校正性维护）、适应性维护、完善性维护或增强、预防性维护或再工程。

**4：什么是预防性维护？**

答：预防性维护是定期进行的计划性维护活动，有助于预防未来可能出现的意外故障。简言之，就是在问题出现之前先行修复。