

串讲 2 部分：

[2001 年统考]在典型的软件体系结构中应包括 5 种构件，它们是 (A)、(B)、(C)、(D)、(E)。

其中，(A) 具有简单的输入/输出关系，没有运行状态的变化。例如，数值计算、过滤器、转换器等。(B) 用于存放共享的、永久性的、结构化的数据。例如，数据库、文件、符号表、超文本等。(C) 使得执行的操作与运行状态紧密耦合。例如，抽象数据类型、面向对象系统中的对象、许多服务器等。(D) 管理其它构件运行的时间、时机及次序。例如，调度器、同步器等。(E) 在实体之间传递信息。例如，通信机制、用户界面等。

供选答案：

- | | | | | |
|-----------------|--------------|-------|---------------|--------------|
| A: ①过程构件 | ②功能构件 | ③信息构件 | ④纯计算构件 | ⑤连接构件 |
| B: ①调用构件 | ②接口构件 | ③耦合构件 | ④存储构件 | ⑤共享构件 |
| C: ①耦合构件 | ②管理构件 | ③监督构件 | ④连接构件 | ⑤测量构件 |
| D: ①显示构件 | ②互连构件 | ③聚合构件 | ④链接构件 | ⑤控制构件 |
| E: ①链接构件 | ②异质构件 | ③信息构件 | ④通信构件 | ⑤继承构件 |

答案：A.④ B.④ C.② D.⑤ E.①

例题

软件详细设计工具可分为 3 类，即图形工具、设计语言和表格工具。图形工具中，(A) 简单而应用广泛。(B) 表示法中，每个处理过程用一个盒子表示，盒子可以嵌套。(C) 可以纵横延伸，图形的空间效果好。

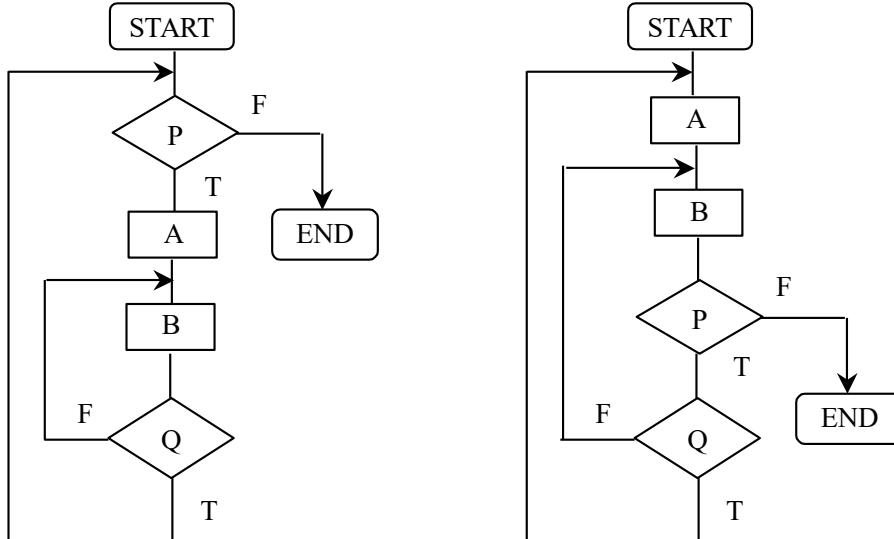
(D) 是一种设计和描述程序的语言，它是一种面向 (E) 的语言。

供选答案：

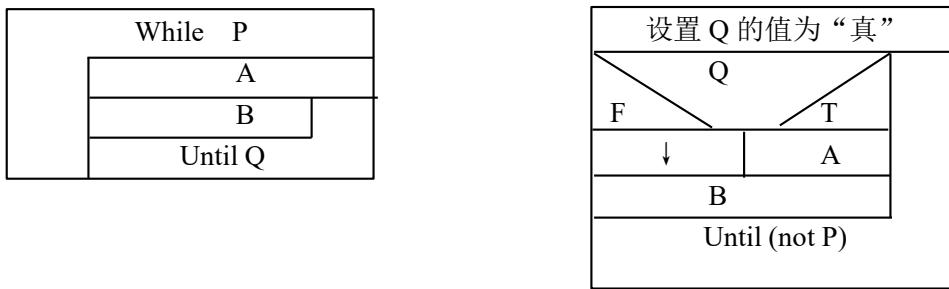
- | | | | |
|--------------|-------------|---------|---------|
| A-C: ①NS 图 | ②流程图 | ③HIPO 图 | ④PAD 图 |
| D: ①C | ②PDL | ③PROLOG | ④Pascal |
| E: ①人 | ②机器 | ③数据结构 | ④对象 |

答案：A.② B.① C.④ D.② E.①

[2001 年统考]下面是两个程序流程图，试分别用 N-S 图和 PAD 图表示之，并计算它们的 McCabe 复杂性度量。



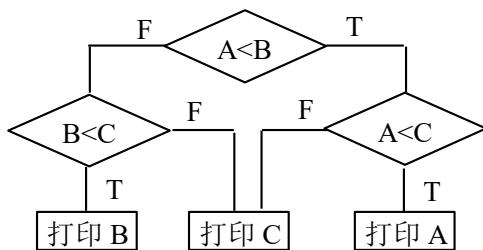
答案：



PAD 图略

McCabe 复杂性度量都为 3。(7-6+2)

下图是使用 BASIC 语言编写的一个打印 A、B、C 三数中最小者的程序流程图。其中出现了 6 个 GOTO 语句，1 个向前，5 个向后，程序可读性很差。



```
if (A<B) goto 120;  
if (B<C) goto 110;  
100  print C;  
     goto 140;  
110  print B;  
     goto 140;  
120  if (A<C) goto 130;  
     goto 100;  
130  print A;  
140
```

试利用基本控制结构，将程序中的 GOTO 语句消去。

使用 if-then-else 结构化构造，则上述程序段可改成如下形式：

```
if (A<B and A<C) then  
    print A  
else if (A>=B and B<c) then  
    print B  
else  
    print C
```

例题

从下列叙述中选出 5 条符合程序设计风格指导原则的叙述。

①嵌套的重数应加以限制

②尽量多使用临时变量

③不滥用语言特色

④不用可以省略的括号

⑤使用有意义的变量名

⑥应尽可能把程序编得短些

⑦把常见的局部优化工作留给编译程序去做

⑧注解越少越好

⑨程序的格式应有助于读者理解程序

⑩应尽可能多用 GOTO 语句

答案：①、③、⑤、⑦、⑨是正确的

从下列关于模块化程序设计的叙述中选出 5 条正确的叙述。

①程序设计比较方便，但比较难以维护

②便于由多个人分工编制大型程序

③软件的功能便于扩充

④程序易于理解，也便于排错

⑤在主存储器能够容纳得下的前提下，应使模块尽可能大，以便减少模块的个数

⑥模块之间的接口叫做数据文件

⑦只要模块之间的接口关系不变，各模块内部实现细节的修改将不会影响别的模块

⑧模块间的单向调用关系叫做模块的层次结构

⑨模块越小，模块化的优点越明显。一般来说，模块的大小都在 10 行以下

答案：②、③、④、⑦、⑧是正确的

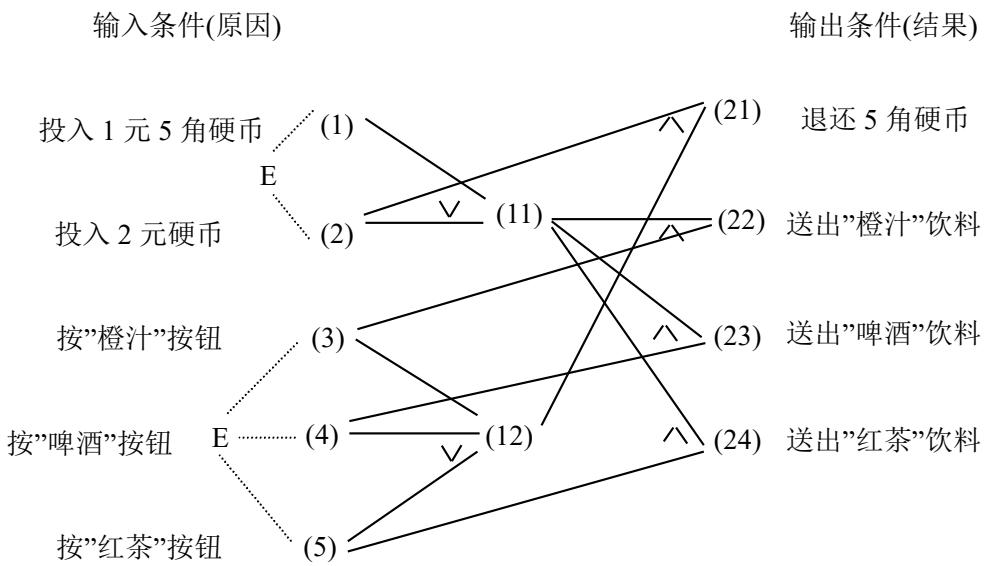
例题

[2000 年统考]设要对一个自动饮料售货机软件进行黑盒测试。该软件的规格说明如下：

有一个处理单价为 1 元 5 角钱的盒装饮料的自动售货机软件。若投入 1 元 5 角硬币，按下“橙汁”、“啤酒”或“红茶”按钮，相应的饮料就送出来。若投入的是 2 元硬币，在送出饮料的同时退还 5 角硬币。

1) (5 分) 试利用因果图法，建立该软件的因果图；

2) (5 分) 设计测试该软件的全部测试用例。



因果图（加入了两个中间结点 11 和 12。11 表示已投币，12 表示已按钮）

测试用例

| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|------|--------------|------|---|---|---|---|---|---|---|---|---|----|----|
| 输入 | 投入 1 元 5 角硬币 | (1) | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 投入 2 元硬币 | (2) | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| | 按“橙汁”按钮 | (3) | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| | 按“啤酒”按钮 | (4) | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| | 按“红茶”按钮 | (5) | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 中间结点 | 已投币 | (11) | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| | 已按钮 | (12) | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 输出 | 退还 5 角硬币 | (21) | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| | 送出橙汁饮料 | (22) | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 送出啤酒饮料 | (23) | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | 送出红茶饮料 | (24) | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

软件测试是软件质量保证的主要手段之一，测试的费用已超过 (A) 的 30%以上。因此，提高测试的有效性十分重要。“高产”的测试是指 (B)。根据国家标准 GB8566-88《计算机软件开发规范》的规定，软件的开发和维护划分为 8 个阶段，其中，单元测试是在 (C) 阶段完成的，集成测试的计划是在 (D) 阶段制定的，确认测试的计划是在 (E) 阶段制定的。

供选择的答案：

- | | |
|------------|---------|
| A: ①软件开发费用 | ②软件维护费用 |
| ③软件开发和维护费用 | ④软件研制费用 |
| ⑤软件生存期全部 | |
- | | |
|------------------------------------|-----------------------------|
| B: ①用适量的测试用例运行程序，证明被测程序正确无误 | ②用适量的测试用例运行程序，证明被测程序符合相应的要求 |
| ③用少量的测试用例运行程序，发现被测程序尽可能多的错误 | |
| ④用少量的测试用例运行程序，纠正被测程序尽可能多的错误 | |
- | | |
|----------------|-------|
| C-E: ①可行性研究和计划 | ②需求分析 |
|----------------|-------|

- | | |
|-------|--------|
| ③概要设计 | ④详细设计 |
| ⑤实现 | ⑥集成测试 |
| ⑦确认测试 | ⑧使用和维护 |

答案: A.① B.③ C.⑤ D.③ E.②

集成测试也叫做 (A) 或 (B)。通常，在 (C) 的基础上，将所有模块按照设计要求组装成系统。子系统的集成测试特别称为 (D)，它所作的工作是要找出子系统和系统需求规格说明之间的 (E)。需要考虑的问题是在把各个模块连接起来的时候，穿越模块接口的数据是否会 (F)；一个模块的功能是否会对另一个模块的功能产生不利的影响；各个 (G) 组合起来，能否达到预期要求的 (H)；(I) 是否有问题；单个模块的误差累积起来是否会放大。

供选择的答案：

- | | | |
|------------|---------|-------|
| A-D: ①单元测试 | ②部件测试 | ③组装测试 |
| ④系统测试 | ⑤确认测试 | ⑥联合测试 |
| E-I: ①子功能 | ②丢失 | ③父功能 |
| ④局部数据结构 | ⑤全局数据结构 | ⑥不一致 |
| ⑦一致 | | |

答案: A.③ B.⑥ C.① D.② E.⑥ F.② G.① H.③ I.⑤

软件测试方法可分为黑盒测试和白盒测试两种。

黑盒测试法是通过分析程序的 (A) 来设计测试用例的方法。除了测试程序外，它还适用于对 (B) 阶段的软件文档进行测试。

白盒测试法是根据程序的 (C) 来设计测试用例的方法。除了测试程序外，它也适用于对 (D) 阶段的软件文档进行测试。

白盒法测试程序时常按照给定的覆盖条件选取测试用例。(E) 覆盖比 (F) 严格，它使得每个判定的每条分支至少经历一次。(G) 覆盖既是判定覆盖，又是条件覆盖，但它并不保证使各种条件都能取到所有可能的值。(H) 覆盖比其他条件都要严格，但它不能保证覆盖程序中的每一条路径。

单元测试一般以 (I) 为主，测试的依据是 (J)。

供选择的答案：

- | | | |
|--------------|----------|-----------|
| A、C: ①应用范围 | ②内部逻辑 | ③功能 |
| ④输入数据 | | |
| B、D: ①编码 | ②软件详细设计 | ③软件总体设计 |
| ④需求分析 | | |
| E、F、G、H: ①语句 | ②判定 | ③条件 |
| ④判定/条件 | ⑤多重条件 | ⑥路径 |
| I: ①白盒法 | ②黑盒法 | |
| J: ①模块功能规格说明 | ②系统模块结构图 | ③系统需求规格说明 |

答案: A.③ B.④ C.② D.② E.② F.① G.④ H.⑤ I.① J.①

[99 年统考]

从下列关于程序设计风格和软件测试的叙述中，选出 5 条正确的叙述

- ①用黑盒法测试时，测试用例是根据程序内部逻辑设计的。
- ②尽量用公共过程或子程序去代替重复的代码段。
- ③测试是为了验证该软件已正确地实现了用户的要求。
- ④对于连锁型分支结构，若有 n 个判定语句，则有 2^n 条路径。
- ⑤尽量采用复合的条件测试，以避免嵌套的分支结构。
- ⑥GOTO 语句概念简单，使用方便。在某些情况下，保留 GOTO 语句反能使写出的程序更

加简洁。

⑦发现错误多的程序模块，残留在模块中的错误也多。

⑧黑盒测试方法中最有效的是因果图法。

⑨在做程序的单元测试时，桩（存根）模块比驱动模块容易编写。

⑩程序效率的提高主要应通过选择高效的算法来实现。

答案：④、⑤、⑥、⑦、⑩

例题

对一个长度为 100,000 条指令的程序进行测试，记录下来的数据如下：

. 测试开始，发现错误个数为 0；

. 经过 160 小时的测试，累计改正 100 个错误，此时，MTTF=0.4 小时；

. 又经过 160 小时的测试，累计改正 300 个错误，此时，MTTF=2 小时。

1) 估计程序中固有的错误总数。

2) 为使 MTTF 达到 10 小时，必须测试和调试这个程序多长时间？

3) 给出 MTTF 与测试时间 t 之间的函数关系

由 Shooman 公式

$$MTTF = \frac{I_T}{K(E_T - E_C(T))}$$

其中， $I_T = 10^5$ ， $MTTF_1 = 0.4$ ， $T_1 = 160$ ， $n_1 = 100$ ， $MTTF_2 = 2$ ， $T_2 = 320$ ， $n_2 = 300$

得

$$\left\{ \begin{array}{l} MTTF_1 = \frac{I_T}{K(E_T - E_C(T_1))} \\ MTTF_2 = \frac{I_T}{K(E_T - E_C(T_2))} \end{array} \right.$$

$$\left\{ \begin{array}{l} 0.4 = \frac{100000}{K(E_T - 100)} \\ 2 = \frac{100000}{K(E_T - 300)} \end{array} \right.$$

1) 解得程序中固有的错误总数 $E_T = 350$ ，此外， $K = 1000$ 。

2) 设 $MTTF = 10$ ，有

$$10 = \frac{100000}{1000(350 - x)}$$

解得 $x = 340$ (即为使 MTTF 达到 10 小时，需要改正 340 个错误)。由可靠性累积曲线

$$E_C(t) = E_T (1 - e^{-Kt})$$

$$\begin{cases} 100 = 350(1 - e^{-160K_1}) \\ 300 = 350(1 - e^{-320K_1}) \end{cases}$$

解得 $K_1=0.01$ 。代入

$$340 = 350(1 - e^{-0.01t}) \text{ 得 } t=(\ln(35))/0.01=356\text{h}$$

因此求得为使 MTTF=10，测试和调试该程序需要时间约为 356 小时。

3) MTTF 与测试时间 t 之间的函数关系：

因为 $E_C(t) = E_T(1 - e^{-K_1 t})$ ，则 $E_T - E_C(t) = E_T e^{-K_1 t}$ 。代入 Shooman 公式：

$$MTTF = \frac{I_T}{K(E_T - E_C(T))} = \frac{I_T}{K \cdot E_T e^{-K_1 t}} = \frac{100000}{1000 \cdot 350 e^{-0.01t}} = \frac{2}{7e^{-0.01t}}$$

例题

定义一个人参加劳动时间的长短为 (A)，其度量单位为 PM (人月) 或 PY (人年)。而定义完成一个软件项目(或软件任务)所需的(A)为(B)，其度量单位是人月/项目(任务)，记做 PM (人月)。进一步地，定义单位 (A) 所能完成的软件 (C) 的数量为软件 (D)，其度量单位为 LOC/PM。它表明一般指 (E) 的一个平均值。例如，一个软件的开发工作量如表所示，该软件共有源代码 2900 行，其中 500 行用于测试，2400 行是执行 (F) 的源代码。则劳动生产率是 (G) (LOC/PM)。

软件开发所需工作量例

| 阶段 | 软件计划 | 需求分析 | 设计 | 编码 | 测试 | 总体 |
|------------|------|------|-----|-----|-----|------|
| 需要工作量 (人月) | 1.0 | 1.5 | 3.0 | 1.0 | 3.5 | 10.0 |

供选择的答案：

A、B、D: ①生产率 ②工作量 ③成本 ④劳动量

E: ①开发全过程 ②某开发阶段 ③软件生存期 ④某开发任务

F、C: ①软件 ②程序 ③进程 ④产品

G: ①520 ②120 ③320 ④240

答案: A.④ B.② C.④ D.① E.① F.② G.④

劳动生产率是 $(2900-500)/10=240$ (LOC/PM)

例题

[2000 年统考]应用 Coad 和 Yourdon 方法构造和评审 OOA 概念模型时，应按如下的五个层次进行：

第一个层次是识别 (A)。它们是对应用中的概念的抽象，可以从用户要求书或其它规格说明中找到。

第二个层次是识别 (B)。它们是 (A) 所保存的信息，同时要给出各个 (A) 之间的实例连接。这些实例连接包括有 1 对 1、1 对多、多对多、多对 1 等联系。

第三个层次是识别 (C)。它们是 (A) 可提供的操作，同时要根据需要的功能给出各个操作之间的消息连接。

第四个层次是建立 (D)。有两种 (D): 整体与部分；泛化与特化。前者表明一种聚合关系，后者表明一种继承关系。

第五个层次是识别 (E)。它相当于高层的模块或子系统。

供选答案:

- A: ①实体 ②规格说明 ③类与对象 ④过程抽象 ⑤功能抽象
B: ①属性 ②数据项 ③服务 ④操作 ⑤事务
C: ①属性 ②数据项 ③服务 ④数据抽象 ⑤实体联系
D: ①框架 ②属性 ③结构 ④通信 ⑤连接
E: ①层次 ②事件 ③状态 ④继承 ⑤主题

答案: A.③ B.① C.③ D.③ E.⑤

在面向对象软件开发过程中特别重视复用。软件构件应独立于当初开发它们的应用而存在。在以后的应用开发中，可以调整这些独立构件以适应新问题的需要。因此，应使得类成为一个 (A) 的单元。这样就有一个 (B) 生存期问题。(B) 生存期有自己的步骤，与任何特定应用的开发 (C)。按照这些步骤，可以完整地描述一个基本 (D)，而不仅仅考虑当前正在开发的系统。系统开发的各个阶段都可能会标识新的类。随着各个新类的标识，(B) 生存期引导开发工作逐个阶段循序渐进。

在设计与实现类时，应尽可能利用既存类提供为当前应用所需要的功能，利用既存类的 3 个可能途径是：(E) 复用既存类，对既存类进行 (F) 以得到满足要求的类，重新开始进行开发。

供选择的答案:

- A: ①可复用 ②可测试 ③可适用 ④可靠
B: ①应用 ②寿命 ③类 ④软件
C: ①相关 ②密切相关 ③负相关 ④无关
D: ①概念 ②实体 ③事件 ④事情
E、F: ①修改 ②更新 ③照原样 ④演化

答案: A.① B.③ C.④ D.② E.③ F.④

类常常被看作是一个抽象数据类型的实现，更合适的是把类看作是某种 (A) 的一个模型。事实上，类是单个的 (B) 单元。类的用户能够操纵的操作叫做类的 (C)。类定义的其余部分给出数据定义和辅助功能定义，包括类的实现。

类的实现常常包括了其它类的实例，这些实例 (D) 被其它对象存取，包括同一个类的其它实例。类的实现可能还包括某些私有方法，实现它们的类可以使用，而其它任何对象都不能使用。

类，就它是一个数据值的聚合的意义来看，与 Pascal 中的记录或 C 中的结构类似，但又有差别。类扩展了通常的记录语义，可提供各种级别的 (E)。类不同于记录，因为它们包括了操作的定义，这些操作与类中声明的数据值有相同的地位。

供选择的答案:

- A. ①功能 ②概念 ③结构 ④数据
B. ①语法 ②词法 ③语义 ④上下文环境
C. ①界面 ②操作 ③行为 ④活动
D. ①可自由地 ②可有控制地 ③可通过继承 ④应受保护不
E. ①可移植性 ②可重复性 ③可访问性 ④继承性

答案: A.② B.③ C.① D.④ E.③

例题一

简述软件工程的概念和基本原理。

答案：软件工程概念：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、技术和方法来开发与维护软件，把经过时间考验而证明正确的管理技术和当前能够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地维护它，这就是软件工程。

软件工程基本原理：

- (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

例题二

简述数据流图的四种成分以及构造数据流图的步骤。

答案：四种成分：数据源点/终点、加工处理、数据流和数据存储。

构造步骤：

- (1) 从问题描述中提取数据流图的上述四种成分
- (2) 构造系统的抽象模型，即画出顶层数据流图
- (3) 根据问题实际和分解原则，逐层分解高层数据流图中的处理，得到下一层次的数据流图子图。每一层次的分解对应上一个层次的一个处理。

例题三

简述常用的三种需求分析方法以及进行需求分析的原则。

答案：方法有结构化分析方法、面向对象的分析方法、原型化分析方法。

原则是：

- (1) 能够表达和理解问题的信息域和功能域。
- (2) 能够对问题进行分解和不断细化，建立问题的层次结构。
- (3) 能够给出系统的逻辑表示和物理表示。

例题四

简述什么是模块独立性？实现模块独立性的优点是什么？衡量模块独立程度的两个定性标准是什么及其含义？

答案：模块独立性是软件系统中每个模块只涉及软件要求的具体子功能，而和软件系统中其他的模块接口是简单的，符合信息隐藏和局部化原则，模块间关联和依赖程度尽可能小。

优点：

- (1) 有效的模块化的软件功能被划分，并且接口被简化，比较容易开发出来。
- (2) 由于因设计和编码修改引起的副作用受到局限，错误传播的可能性被减少，并且

模块复用成为可能，所以独立的模块更易于维护和测试。

两个度量标准：

- (1) 體合：衡量不同模块彼此间互相依赖（连接）的紧密程度。
- (2) 内聚：衡量一个模块内部各个元素彼此结合的紧密程度。

例题五

简述什么是静态分析测试？什么是动态执行测试？

答案：(1) 静态分析技术：静态分析技术不执行被测试软件，可对需求分析说明书、软件设计说明书、源程序做结构检查、流图分析、符号执行等来找出软件错误。可以人工进行分析，也可以用测试工具静态分析程序来进行，被测试程序的正文作为输入，经静态分析程序分析得出分析结果。

(2) 动态分析技术：是执行被测程序，由执行结果分析程序可能出现的错误。可以人工设计程序测试用例，也可以由测试工具动态分析程序来做检查与分析。动态测试包括功能测试和结构测试。它把程序看作为一个函数，输入的全体称为函数的定义域，输出的全体称为函数的值域，函数则描述了输入的定义域与输出值域的关系。

串讲 1 部分：

例题

开发软件时，对于提高软件开发人员工作效率至关重要的是 (A)。
软件工程中描述软件生存周期的瀑布模型一般包括计划、(B)、设计、
编码、测试、维护等几个阶段。其中，设计阶段在管理上又可以依次
分成 (C) 和 (D) 两个步骤。(p.13)

供选答案：

A: ①程序开发环境 ②操作系统的资源管理功能

③开发程序人员数量 ④计算机的并行处理能力

B: ①需求分析 ②需求调查

③可行性分析 ④问题定义

C、D: ①方案设计 ②代码设计 ③概要设计 ④数据设计

⑤运行设计 ⑥详细设计 ⑦故障处理设计 ⑧软件体
系结构设计

答案: A. ① B. ① C. ③ D. ⑥

有人将软件的发展过程划分为 4 个阶段:

第一阶段(20世纪50年代)称为“程序设计的原始时期”。这时既没有(A),也没有(B),程序员只能用机器指令编写程序。

第二阶段(20世纪60年代)称为“基本软件期”。出现了(A),并逐渐普及。随着(B)的发展,编译技术也有较大的发展。

第三阶段(20世纪60年代末-70年代中)称为“程序设计方法时代”。

这一时期,与硬件价格下降相反,软件开发费用急剧上升。人们提出了(C)和(D)等程序设计方法,设法降低软件的开发费用。

第四阶段(20世纪70年代中期至今)称为“软件工程时期”,软件开发技术不再仅仅是程序设计技术,而是包括了与软件开发的各个阶段,如(E)、(F)、编码、单元测试、综合测试、(G)及其整体有关的各种管理技术。

供选择的答案:

A-D: ①汇编语言 ②操作系统 ③虚拟存储器概念 ④高级语
言

⑤结构化程序设计 ⑥数据库概念 ⑦固件 ⑧模块化程
序设计

E-G: ①使用和维护 ②兼容性的确认 ③完整性的确认 ④设计
⑤需求定义 ⑥图象处理

答案： A.(1) B.(4) C.(5) D.(8) E.(5) F.(4)

G.(1)

例题

判断正误：可行性研究报告应评述为了合理地达到开发目标而可能选择的各种方案，以便用户进行抉择。因此，编写者不必提出结论。

答案： 错

例题 请用判定表表示下面描述的耗电记费系统中电费的计算方法
耗电记费系统可以采用固定价格收费和浮动价格收费两种方式。若采用固定价格方式收费，对每月耗电 $100\text{KW}\cdot\text{h}$ 以下的用户只征收最低标准费，超过 $100\text{KW}\cdot\text{h}$ 的用户按价格表 A 收费；若采用浮动价格方式收费，则每月耗电 $100\text{KW}\cdot\text{h}$ 以下的用户按价格表 A 收费，超过 $100\text{KW}\cdot\text{h}$ 的用户按价格表 B 收费。

答案：

| 条 件 | 收费方式 | 固定 | 固定 | 浮动 | 浮动 |
|--------|-----------|------------|--------|------------|----|
| 耗电规模 | >100 | ≤ 100 | >100 | ≤ 100 | |
| 操作 | 征收最低标准费 | | ✓ | | |
| | 按价格表 A 收费 | ✓ | | | ✓ |
| | 按价格表 B 收费 | | | ✓ | |

| 条 件 | 固定 | 固定 | 浮动 | 浮动 |
|--------|--------|------------|--------|------------|
| | >100 | ≤ 100 | >100 | ≤ 100 |

| | | | | |
|----|-----------|---------|-----------|-----------|
| 操作 | 按价格表 A 收费 | 征收最低标准费 | 按价格表 B 收费 | 按价格表 A 收费 |
|----|-----------|---------|-----------|-----------|

例题

软件需求分析的任务不应包括 (A)。进行需求分析可使用多种工具，但 (B) 是不适用的。在需求分析中，分析员要从用户那里解决的最重要的问题是 (C)。需求规格说明书的内容不应当包括 (D)。该文档在软件开发中具有重要的作用，但其作用不应当包括 (E)。

供选答案：

- | | | |
|----|------------------|----------------------|
| A: | ①问题分析 | ②信息域分析 |
| | ③ 结构化程序设计 | ④确定逻辑模型 |
| B: | ①数据流图 | ②判定表 |
| | ③ PAD 图 | ④数据词典 |
| C: | ① 要让软件做那么 | ②要给软件提供哪些信息 |
| | ③要求软件工作效率如何 | ④要让软件具有什么样的结构 |
| D: | ①对重要功能的描述 | ② 对算法的详细过程性描述 |
| | ③软件确认准则 | ④软件的性能 |
| E: | ①软件设计的依据 | ②用户和开发人员对软件要“做什么” |
| | ③软件验收的依据 | ④ 软件可行性分析的依据 |

答案： A. ③ B. ③ C. ① D.② E.④

在结构化分析方法中，用以表达系统内数据的运动情况的工具有()。

供选答案：

- A.数据流图 B.数据词典 C.结构化英语 D.判定表与判定树

答案： A

例题

在众多的设计方法中，SD 方法是最受人注意的，也是最广泛运用的一种，这种方法可以同分析阶段的（A）方法及编程阶段的（B）方法前后衔接，SD 方法考虑如何建立一个结构良好的程序结构，它提出了评价模块结构质量的两个具体标准-块间联系和块内联系。SD 方法的最终目标是（C），用于表示模块间调用关系的图叫（D）。

另一种比较著名的设计方法是以信息隐蔽为原则划分模块，这种方法叫（E）方法。

供选择的答案：

A-B: ①Jackson ②SA ③SC ④Parnas ⑤SP

C: ①块间联系大，块内联系小 ②块间联系大，块内联系小

③块间联系小，块内联系大 ④块间联系小，块内联系小

D: ①PAD ②HCP ③SC ④SADT ⑤HIPO ⑥NS

E: ①Jackson ②Parnas ③Turing ④Wirth ⑤

Dijkstra

答案： A.② B.⑤ C.③ D.③ E.②

块间联系和块内联系是评价程序模块结构质量的重要标准。联系的方式、共用信息的作用、共用信息的数量和接口的（A）等因素决

定了块间联系的大小。在块内联系中，(B) 的块内联系最强。

SD 方法的总的原则是使每个模块执行(C)功能，模块间传送(D)参数，模块通过(E)语句调用其它模块，而且模块间传送的参数应尽量(F)。

此外，SD 方法还提出了判定的作用范围和模块的控制范围等概念。SD 方法认为，(G) 应该是(H) 的子集。

供选择的答案：

A: ①友好性 ②健壮性 ③简单性 ④安全性

B: ①巧合内聚 ②功能内聚 ③通信内聚 ④信息内聚

C: ①一个 ②多个

D: ①数据型 ②控制型 ③混合型

E: ①直接引用 ②标准调用 ③中断 ④宏调用

F: ①少 ②多

G-H: ①作用范围 ②控制范围

答案：A.③ B.② C.① D.① E.② F.① G.

① H.②

从下列有关系统结构图的叙述中选出正确的叙述

- 1) 系统结构图中反映的是程序中数据流的情况。
- 2) 系统结构图是精确表达程序结构的图形表示法。因此，有时可将系统结构图当作程序流程图使用。
- 3) 一个模块的多个下属模块在系统结构图中所处的左右位置是无关紧要的。

4) 在系统结构图中，上级模块与其下属模块之间的调用关系用有向线段表示。这时，使用斜的线段和水平、垂直的线段具有相同的含义。

答案：4

[99 年统考]

在完成软件概要设计，并编写出相关文档之后，应当组织对概要设计工作的评审。评审的内容包括：

分析该软件的系统结构、子系统结构，确认该软件设计是否覆盖了所有已确定的软件需求，软件每一成分是否可(A)到某一项需求。

分析软件各部分之间的联系，确认该软件的内部接口与外部接口是否已经明确定义。模块是否满足(B)和(C)的要求。模块(D)是否在其(E)之内。

供选择的答案：

A: ①覆盖 ②演化 ③追溯 ④等同 ⑤连接

B: ①多功能 ②高内聚 ③高耦合 ④高效率 ⑤可读性

C: ①多入口 ②低内聚 ③低耦合 ④低复杂度 ⑤低强度

D-E: ①作用范围 ②高内聚 ③低内聚 ④取值范围 ⑤控制范围

答案：A.③ B.② C.③ D.① E.⑤