

## 一、选择题出题考点：（15道题，每题2分）

软件危机的概念，产生的主要原因；可行性研究的概念与理解；软件维护的概念和理解；软件配置项；结构程序设计的概念；模块复杂度；白盒测试中逻辑覆盖的理解；单元测试、集成测试、验收测试和回归测试的概念及理解；模块的高内聚低耦合特性理解；典型的软件开发过程模型理解，瀑布模型、螺旋模型、喷泉模型、增量模型，并理解各模型的优势；可行性研究的步骤：结构化分析方法：详细设计阶段（书本6.3过程设计工具，程序流程图）；软件维护活动；软件生命周期分为几个阶段；逆向工程的概念及理解；UML中的对象模型、动态模型和功能模型的理解；边界值测试；

### 下面是一些参考资料

典型的软件过程模型有瀑布模型、增量模型、演化模型（原型模型、螺旋模型）、喷泉模型、基于构件的开发模型和形式化方法模型。下面我们一一来看看：

#### 【瀑布模型】

1. 定义：将软件生存周期中的各个活动规定为依线性顺序连接的若干阶段的模型，包括需求分析、设计、编码、测试、运行与维护。它规定了由前至后、相互衔接的固定次序，如同瀑布流水逐级下落。
2. 适用于：以文档作为驱动、适合于软件需求很明确的软件项目。
3. 优点：容易理解，管理成本低；强调开发的阶段性早期计划及需求调查和产品测试。
4. 不足：客户必须能够完整、正确、清晰地表达出他们的需要；需求或设计的错误往往是在项目后期才被发现，对于项目风险控制能力较弱，经常延期。
5. 关键点：需求明确

#### 【增量模型】

1. 定义：将需求分段为一系列增量产品，每一增量可以分别开发。根据第一个增量，可以快速开发出核心产品。
2. 适用于：软件体系结构开放，加入新构件过程简单。
3. 优点：第一个可交付版本所需要的时间和成本很少；所承担的风险不大；减少用户需求的变更。
4. 特点：增量模型是一种非整体开发的模型，是一种进化式的开发过程。增量模型从部分需求出发，先建立一个不完整的系统，通过测试运行这个系统取得经验和反馈，进一步使系统扩充和完善。如此反复进行，直至软件人员和用户对所设计的软件系统满足为止。
5. 不足：如果没有对变更要求进行规划，那么会导致后来增量的不稳定；如果需求不稳定完整，那么会导致重新开发；管理发生的成本、进度和配置的复杂性会超出组织的能力。

#### 【原型模型】

1. 定义：快速建立起来的可以在计算机上运行的程序，或仅仅是一个演示界面。
2. 适用于：需求不够明确的项目。
3. 优点：能快速、低成本地构建原型。
4. 不足：必须要求具有技能高水平的原型化人员。
5. 关键词：需求不够明确

### 【喷泉模型】

1. 定义：以用户需求为动力，以对象作为驱动的模型。具有迭代性和无间隙性。
2. 适用于：面向对象的开发方法。
3. 优点：各阶段没有明显的界线，可以同步开发，提高了软件开发效率，节省了时间。
4. 不足：各开发阶段是重叠的，不利于项目的管理；严格要求文档，使得审核的难度加大。

### 【基于构件的开发模型】

1. 定义：利用预先包装的构件来构造应用系统。
2. 适用于：需要一定的构件模型支持的软件项目。
3. 优点：构件组装模型导致软件的复用，提高了开发效率；允许多个项目同时开发，降低了费用，提高了可维护性。
4. 不足：构件的引入具有较大的风险；过分依赖于构件，构件的质量影响产品的质量；需要精干的、有经验的分析人员和开发人员，客户的满意度低。

### 【形式化方法模型】

1. 定义：建立在严格数学基础上的一种开发方法。
2. 适用于：需要生成计算机软件形式化的数学规格说明。
3. 优点：易于发现需求的歧义性、不完整性和不一致性；易于对分析模型、设计模型和程序进行验证。
4. 不足：需要通过严密的数学演算。

1. 逆向工程从源代码或目标代码中提取设计信息，通常在原软件生命周期的（软件维护）阶段进行。

## 二、填空题：（10 道题，每题 1 分）

考点：软件工程的概念，软件开发过程模型，可行性研究；结构化分析方法；详细设计，软件维护，容错软件，软件测试，软件结构，软件开发。

下面是一些参考资料

软件工程的概念。所谓软件工程，是指运用计算机科学、数学管理科学等相关原理，以工程化的思想和方法，来解决软件问题的工程。

软件工程的目的：①提高软件的生产率；②提高软件质量、降低软件成本。

软件危机的概念（软件工程的来由）：八九十年代随着软件行业的发展，以前的开发方法已经无法适应当前的状态，开发出来的软件质量问题比较多，经常延期不能按时交付、成本越来越高，形成了恶性循环，也就是常说的“软件危机”。为了解决软件危机，就引入了项目工程的思想来进行软件开发，也就是软件工程。

软件工程的三要素：①方法：为完成开发各项软件任务采用的技术叫做方法。  
②工具：为了利用方法所需的特定的支撑环境。  
③过程：为了获得高质量低成本的软件，要完成的一系列的任务  
的框架

## 【瀑布模型】

1. 定义：将软件生存周期中的各个活动规定为依线性顺序连接的若干阶段的模型，包括需求分析、设计、编码、测试、运行与维护。它规定了由前至后、相互衔接的固定次序，如同瀑布流水逐级下落。

2. 适用于：以文档作为驱动、适合于软件需求很明确的软件项目。

3. 优点：容易理解，管理成本低；强调开发的阶段性早期计划及需求调查和产品测试。

4. 不足：客户必须能够完整、正确、清晰地表达出他们的需要；需求或设计的错误往往是在项目后期才被发现，对于项目风险控制能力较弱，经常延期。

5. 关键点：需求明确

## 【增量模型】

1. 定义：将需求分段为一系列增量产品，每一增量可以分别开发。根据第一个增量，可以快速开发出核心产品。

2. 适用于：软件体系结构开放，加入新构件过程简单。

3. 优点：第一个可交付版本所需要的时间和成本很少；所承担的风险不大；减少用户需求的变更。

4. 特点：增量模型是一种非整体开发的模型，是一种进化式的开发过程。增量模型从部分需求出发，先建立一个不完整的系统，通过测试运行这个系统取得经验和反馈，进一步使系统扩充和完善。如此反复进行，直至软件人员和用户对所设计的软件系统满足为止。

5. 不足：如果没有对变更要求进行规划，那么会导致后来增量的不稳定；如果需求不稳定完整，那么会导致重新开发；管理产生的成本、进度和配置的复杂性会超出组织的能力。

结构化开发方法（Structured Developing Method）是现有的软件开发方法中最成熟，应用最广泛的方法，主要特点是快速、自然和方便。结构化开发方法由结构化分析方法（SA 法）、结构化设计方法（SD 法）及结构化程序设计方法（SP 法）构成的。

结构化分析（Structured Analysis，简称 SA 法）方法是面向数据流的需求分析方法，是 70 年代末由 Yourdon, Constantine 及 DeMarco 等人提出和发展，并得到广泛的应用。它适合于分析大型的数据处理系统，特别是企事业管理系统。

SA 法也是一种建模的活动，主要是根据软件内部的数据传递、变换关系，自顶向下逐层分解，描绘出满足功能要求的软件模型。

结构化分析方法的基本思想是“分解”和“抽象”。

分解：是指对于一个复杂的系统，为了将复杂性降低到可以掌握的程度，可以把大问题分解成若干小问题，然后分别解决。

抽象：分解可以分层进行，即先考虑问题最本质的属性，暂把细节略去，以后再逐层添加细节，直至涉及到最详细的内容，这种用最本质的属性表示一个子系统的方法就是“抽象”。

结构化程序设计方法的要点是使用 顺序、选择和循环 结构，自顶向下、逐步求精地构造算法或程序

在详细设计阶段，一种历史最悠久、使用最广泛的描述程序逻辑结构的工具是 程序流程图。

详细描述处理过程常用的三种工具是图形、语言和 表格。

软件维护有哪些内容？

答：（1）改正性维护。在软件交付使用后，一些隐含的错误在某些特定的使用环境下会暴露出来。为了识别和纠正错误，修改软件性能上的缺陷，应进行确定和修改错误的过程，这个过程就称为校正性维护。

- (2) 适应性维护。为了使应用软件适应计算机硬件、软件环境及数据环境的不断发生的变化而修改软件的过程称为适应性维护。
- (3) 完善性维护。为增加软件功能、增强软件性能、提高软件运行效率而进行的维护活动称为完善性维护。
- (4) 预防性维护。为了提高软件的可维护性和可靠性而对软件进行的修改称为预防性维护。

容错软件的定义

归纳容错软件的定义，有以下 4 种：

- (1) 规定功能的软件，在一定程度上对自身错误的作用(软件错误)具有屏蔽能力，则称该软件为具有容错功能的软件，即容错软件。
- (2) 规定功能的软件，在一定程度上能从错误状态自动恢复到正常状态，则称该软件为容错软件。
- (3) 规定功能的软件，在因错误发生错误时仍然能在一定程度上完成预期的功能，则称该软件为容错软件。
- (4) 规定功能的软件，在一定程度上具有容错能力，则称该软件为容错软件。

软件结构（即由模块组成的层次系统）可以用层次图或结构图来描绘。

软件开发是一个自顶向下逐步细化和求精的过程，而软件测试是一个自低向上集成的过程

### 三、判断题考点：（共 10 道题，每题 1 分）

软件工程概念的理解；可行性研究的理解；逻辑覆盖的理解；软件测试的目的；模块化设计的理解；需求分析；代码审查；项目风险管理；软件重用，回归测试，软件维护

### 四、简答题：（共 20 分，每小题 5 分）

1. 软件复用的概念及软件复用的种类有？
2. 数据字典的概念，及简述数据字典与数据流图关系？
3. 软件测试的概念及软件测试的目的？
4. 软件生命周期的各个阶段，及每个阶段的基本任务？
5. 针对面向对象设计，系统分解为几个子系统，各子系统的任务是什么？
6. 黑盒测试法的概念，及黑盒测试常见方法？
7. 软件项目管理的概念，并简要说明
8. 软件维护的概念，及软件维护的类型，各自的任务是什么？
9. 面向对象建模技术的三种模型是什么，三者之间的关系以及，分别用什么图形来描述？
10. 简述软件项目管理的主要内容。

### 五、大题（共 30 分，每小题 10 分）

1. 数据字典定义数据方法

参考链接：[软件工程~数据字典例子解释 数据字典实例详解-CSDN 博客](#)

2. 画数据流图

参考链接：[软件工程 -- 数据流图的画法 数据流图怎么画-CSDN 博客](#)

3. 判定树和判定表，任选一种

参考链接：[详细设计的工具——判定表与判定树 判定表和判定树-CSDN 博客](#)

4. 白盒测试

参考链接：[详解软件测试中白盒测试基本概念及四种白盒测试方法以及六种逻辑覆盖法（语句覆盖、判定覆盖、条件覆盖、判定条件覆盖、条件组合覆盖、路径覆盖）-CSDN 博客](#)

5、黑盒测试

参考链接：[黑盒测试的测试方法及其案例 黑盒测试用例-CSDN 博客](#)

6、状态转换图

参考链接：[【UML】 - ATM 机状态机图 atm 机状态转换图-CSDN 博客](#)

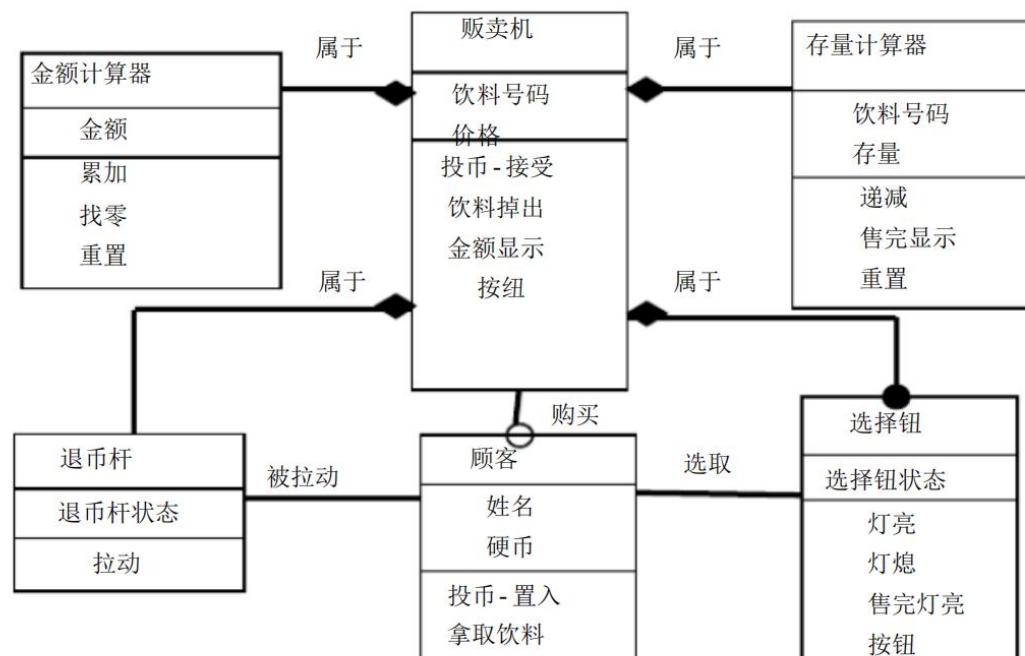
[复印机状态转换解析-CSDN 博客](#)

7、面向对象分析题

根据以下描述，使用面向对象的分析方法，分析有哪些主要对象和这些对象的主要属性、方法，画出对象图。

一个饮料自动售货机可以放置五种不同或部分相同的饮料，可由厂商根据销售状况自动调配，并可随时重新设置售价，但售货机最多仅能放置 50 罐饮料，其按钮设计在各种饮料样本的下方，若经金额计算器累计金额足够，则选择键灯会亮；若某一种饮料已销售完毕，则售完灯会亮。

顾客将硬币投入售货机，经累加金额足额的饮料选择键灯亮，等顾客按键选择。顾客按键后饮料由出物口掉出，并自动结算及找钱。顾客可在按下选择键前任何一个时刻，可以拉动退币杆取消交易收回硬币。



下面是一些参考题：

- 1.某项目组拟开发一个大规模系统，且具备了相关领域及类似规模系统的开发经验。下列过

- 程模型中，（B）最适合开发此项目。（2010年下半年）  
A.原型模型 B.瀑布模型 C.V模型 D.螺旋模型
- 2.为了有效地捕获系统需求，应采用（C）（2011年上半年）  
A.瀑布模型 B.V模型 C.原型模型 D.螺旋模型
- 3.若全面采用新技术开发一个大学记账系统，以替换原有系统，则宜选择采用（A）进行开发。（2011年下半年）  
A.瀑布模型 B.演化模型 C.螺旋模型 D.原型模型
- 4.某开发小组欲开发一个规模较大、需求较明确的项目。开发小组对项目领域熟悉且该项目与小组开发的某一项目相似，则适宜采用（A）开发过程模型。（2012年下半年）  
A.瀑布模型 B.演化模型 C.螺旋模型 D.喷泉模型
- 5.（A）开发过程模型最不适用于开发初期对软件需求缺乏准确全面认识的情况。（2013年下半年）  
A.瀑布模型 B.演化模型 C.螺旋模型 D.增量模型
- 6.某公司计划开发一产品，技术含量很高，与客户相关的风险也很多，则最适于采用（D）开发过程模型。（2015年上半年）  
A.瀑布模型 B.原型模型 C.增量模型 D.螺旋模型
- 7.（D）不是增量式开发的优势。  
A、软件可以快速地交付  
B、早期的增量作为原型，从而可以加强对系统后续开发需求的理解  
C、具有最高优先级的功能首先交付，随着后续的增量不断加入，使得最重要的功能得到更多的测试  
D、很容易将客户需求划分为多个增量

- 1.软件需求分析阶段的工作，可以分为以下4个方面：对问题的识别、分析与综合、编写需求分析文档以及（C）。  
A.总结 B.阶段性报告  
C.需求分析评审 D.以上答案都不正确
- 2.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（A）。  
A.数据流图 B.数据词典  
C.结构化 D.判定表与判定树
- 3.软件需求分析的任务不应包括（C）。  
A.问题分析 B.信息域分析  
C.结构化程序设计 D.确定逻辑模型
- 4.进行需求分析可使用多种工具，但（C）是不适用的。  
A.数据流图 B.判定表 C.PAD图 D.数据词典

5. 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（ A ）。

- A. 要让软件做什么
- B. 要给该软件提供哪些信息
- C. 要求软件工作效率如何
- D. 要让软件具有什么样的结构

6、为了产生结构化的流程图，应由三种基本控制结构顺序组合或 完整嵌套 而成。

7、PAD 是一种由左向右展开的二维树型结构，图中的竖线为程序的 层次线 。

8、PDL 具有严格的关键字外层语法，用于定义控制结构、数据结构和 模块接口设计 。

9、在详细设计阶段，除了对模块内的算法进行设计，还应对模块内的 数据结构 进行设计。

10、软件结构（即由模块组成的层次系统）可以用层次图或结构图来描绘。