**事务分析设计的步骤？**

(1) 在DFD图中确定事务中心、接收部分(包含全部接收路径)和发送部分(包含全部动作路径);

　(2) 画出SC图框架,把DFD图的三部分，分别映射"为事务控制模块,接收模块和动作发送模块.一般得到SC图的顶层和第一层(如果第一层简单可以并入顶层);

　　(3) 分解和细化接收分支和动作分支,完成初始的SC图;

　　(4) 对初始结构图按照设计准则进行精化与改进。

**什么是模块化？模块设计的准则？**

　　模块化是按规定的原则将一个大型软件划分为一个个较小的、相对独立但又相关的模块。

　　模块设计的准则：

　　(1) 改进软件结构, 提高模块独立性:在对初步模块进行合并、分解和移动的分析、精化过程中力求提高模块的内聚，降低藕合。

　　(2) 模块大小要适中:大约50行语句的代码，过大的模块应分解以提高理解性和可维护性;过小的模块,合并到上级模块中。

　　(3) 软件结构图的深度、宽度、扇入和扇出要适当。一般模块的调用个数不要超过5个。

　　(4) 尽量降低模块接口的复杂程度；

　　(5) 设计单入口、单出口的模块。

　　(6) 模块的作用域应在控制域之内。

**可行性研究包括哪几方面的内容？**

答：

(1)经济可行性：是否有经济效益，多长时间可以收回成本；

　(2)技术可行性：现有技术能否实现本系统，现有技术人员能否胜任，开发系统的资源能否满足；

　(3)运行可行性：系统操作在用户内部行得通吗？

　(4)法律可行性：新系统开发是否会侵犯他人、集体或国家利益，是否违反国家法律。

**良好的编码风格应具备哪些条件？**

答:应具备以下条件：

（1）使用标准的控制结构；

（2）有限制地使用GOTO语句；

（3）源程序的文档化 （应具备以下内容）

①有意义的变量名称 —— “匈牙利命名规则 ”。

②适当的注释 ——“注释规范”。

③标准的书写格式：

——用分层缩进的写法显示嵌套结构的层次（锯齿形风格）；

——在注释段的周围加上边框；

——在注释段与程序段、以及不同程序段之间插入空行；

——每行只写一条语句；

——书写表达式时，适当使用空格或圆括号等作隔离符。

**作为测试工程师, 如果你发现需求一直在变化, 怎么办**

通过与产品经理沟通, 稳定产品需求

优先测试需求稳定的功能模块

对需求变化频繁的功能部分, 编写通用性较好的测试用例

**软件工程的七条基本原理：**

1. 用分阶段的生存周期计划严格管理
2. 坚持进行阶段评审
3. 严格实施产品的控制
4. 采用现代程序技术
5. 结果应该能清楚的审查
6. 开发小组的成员应少而精
7. 承认不断改进软件工程的必要性

用例图、类图、对象图、构件图、部署图|状态图、时序图、协作图、活动图

**简述文档在软件工程中的作用。**

答：

(1) 提高软件开发过程的能见度

(2) 提高开发效率

(3) 作为开发人员阶段工作成果和结束标志

(4) 记录开发过程的有关信息便于使用与维护；

(5) 提供软件运行、维护和培训有关资料；

(6) 便于用户了解软件功能、性能。

文档也是软件产品的一部分，没有文档的软件就不成软件

软件文档的编制在软件开发工作中占有突出的地位和相当大的工作量

高质量的文档对于发挥软件产品的效益有着重要的意义

静态逻辑模型包括用例图、类图、对象图、构件图和部署图。其中用例图描述系统功能；类图描述系统的静态结构，描述各种类及其关系，结构，层次等；对象图描述系统在某个时刻的静态结构；构件图描述实现系统的元素的组织；部署图描述系统环境元素的配置。

动态逻辑模型包括状态图、时序图、协作图和活动图。用以描述对象之间的互相作用。状态图描述一个特定对象的所有可能状态及引起其状态转移的事件；时序图用来描述对象之间动态的交互关系，着重体系对象间消息传递的时间顺序；协作图用于描述相互协作的对象间的交互关系和链接关系；活动图用来描述系统元素的活动，即可用来描述操作（类的方法）的行为，也可以描述用例和对象内部的工作过程。

内聚：标志一个模块内各个元素彼此结合的紧密程度。按照从弱到强的顺序，分为：偶然性内聚、逻辑性内聚、时间性内聚、过程性内聚、通信性内聚、顺序性内聚、功能性内聚。

耦合：是对一个软件结构内各个模块之间互连程度的度量。耦合强弱取决于模块间接口的复杂程度，调用模块的方式，以及通过接口的信息。按照从弱到强的顺序，分为：非直接耦合、数据耦合、特征耦合、控制耦合、外部耦合、公共耦合、内容耦合。

软件设计的任务：把分析阶段产生的软件需求说明转换为用适当手段表示的软件设计文档。不管采用何种软件设计方法，软件设计一般都包括数据设计、体系结构设计、接口时间和过程设计等内容。

软件设计的“四项基本原则”：模块独立性，低耦合，高内聚，公共模块。

**计算机四级软件工程论述题**

计算机四级软件工程论述题

1.软件测试是软件质量保证的重要措施。测试的目的是在软件投入生产性运行前,应尽可能地发现并排除软件中隐含的各种错误。测试是对软件规格说明、设计和编码的全面和最后的审查。因此,软件测试应贯穿在整个软件开发的全过程。

**⑴试陈述软件测试的基本任务和方法。(6分)**

**⑵简述黑箱测试和白箱测试的内容和方法。(10分)**

**⑶陈述软件测试过程的基本组成,并介绍各种测试对象的测试内容和方法(15分)**

答案:(1)软件测试是软件开发过程中的重要阶段,是软件质量保证的重要手段.其任务可归纳为三个方面:

1)预防软件发生错误

2)发现改正程序错误

3)提供错误诊断信息

目前,软件测试的方法有三种:动态测试,静态测试和正确性证明.

(2)黑箱测试是一种功能测试,测试时完全不考虑程序内部细节,结构和实现方式,仅检验程序结果与说明书的一致性.测试实例设计完全以说明书为准.黑箱测试不关心程序内部的逻辑,而只是根据程序的功能说明来设计测试用例.在使用黑箱法时,手头只需要有程序功能说明就可以了.黑箱法分以下几种:等价分类法.边缘值分析法,因果图法,错误推测法。

白箱测试是一种结构测试,它与程序内部结构相关,要利用程序结构的实现细节知识设计测试实例.它将涉及程序设计风格,控制方法,潭语句,数据库设计,编码细节.白箱测试时将包括:

1)语句测试——要求程序中每个语句最少检查一次.

2)分支/判定测试——要求程序中每个分支路径最少检查一次.

3)条件覆盖---每一判定中的每个条件，分别按“真”、“假”至少各执行一次；

4)条件组合覆盖---求出判定中所有条件的各种可能组合值，每一可能的条件组合至少执行一次。

5)路径测试——要求程序中每条路径最少检查一次.

白箱测试考虑的是测试实例对程序内部逻辑的覆盖程度.为了衡量测试的覆盖程度,需要建立一些标准.覆盖程序从低到高分别为:语句覆盖,判定覆盖,条件覆盖,判定/条件覆盖,条件组全覆盖.

(3)

1)单元测试

通过测试发现该模块单元的子程序或过程的实际功能与该模块的功能和接口的描述是否相符,以及是否有编码错误存在.

2)集成测式

集成测试是在组装软件模块的同时,进行测试以查找与接口有关的错误.组装模块的过程可以分为自顶向下组装法和自底向上组装法.

3)有效性测试

有效性测试是指:当软件的运行达到了用户的期望时,则认为软件是有效的.

4)系统测试

系统测试是指将软件系统与硬件,外设或其他系统元素结合在一起,对整个软件系统进行测试.

5)验收测试

系统测试通过后,用户根据合同进行验收测试,确定系统功能和性能的可接受性.

2.**原型化方法或称快速原型化方法**是一种当前常用的软件工程方法，尽管其特征民传统的结构化生命周期法有所不同，但从原型化方法的演变仔细分析，仍可看出它与结构化方法的内在联系，试回顾两种方法的发展特征和演进历史并论述下述问题：

**(1)两种方法的基本特征和内在联系。(14分)**

**(2)结构化生命周期方法的优点及其适应对象。(8分)**

**(3)实施快速原型法的前提和条件。(8分)**

答案:

(1)结构化方法的基本特征:是一类预先严格定义需求的方法,它强调用户需求第一,并在需求分析报告中即基本冻结了用户的需求,其开发的阶段划分,文档的规范都基于需求定义的明确。

快速原型方法的基本特征:是一类动态定义需求的方法,它强调快速获取用户基本需求,快速建立需求模型,并与用户交互,迭代并修改和完善模型.

两种方法间的联系:结构化方法实施的过程,培养了一批熟悉业务,掌握领域知识的人员,他们是实施原型化的理想人员;结构化方法实施中形成了一批软件开发工具.它们使快速开发原型系统成为可能;结构化方法的应用积累了一批原型系统,为原型化方法实施提供了基础;原型化方法是一类需求定主的快速方法,一旦需求定义明确;即可实施严格的结构化方法.

(2)结构化生命周期方法的主要优点是:方法强调需求分析的重要性,即体现软件开发中用户至上的原则;开发过程的阶段划分,各阶段的具体任务明确;各阶段有具体的描述工具易于掌握;强调文档的重要性,并给出文档的内容和格式,因此方法具有易操作性,是软件开发人员应该掌握的基本方法.

结构化方法更适合于需求能够预先明确定义的软件系统,如系统软件或某些实时控制软件等.

(3)实施快速原型法的基本前提条件一般是:有高水平的实施原型化的人员,他们掌握工具,了解业务,能快速获取用户需求;有快速建造模型系统的工具,二者缺一不可.有相应原型系统的积累可加速实施,但不是实施原型化方法的前提.

3.软件产品生产周期长、耗资巨大，必须特别注意保证质量，而通常保证软件质量的措施可归为四方面，即**复审、复查、管理复审和测试**，不同的方面反映了软件质量保证措施中的不同需要，试回答以下问题并适当加以阐述：

**(1)、复审、复查、管理复审和测试各自包括的具体内容是什么，它在哪些方面对软件质量的保证产生了作用?(12分)**

**(2)、软件复审和软件测试之间有什么联系，又有什么差别?各自有什么侧重?(6分)**

**(3)、软件测试的目的是什么，对其具体的内容和实现过程做—扼要陈述，无需对测试方法做出介绍、(12分)**

答案:

本题主要考查考生对软件质量保证措施的全面了解程度。软件质量保证措施除了常规的对编码进行测试外，还有软件生命周期阶段成果的复审，对各阶段文档和材料的复查，从管理的角度对软件开发工作的复审以及对编码的测试，要求学生建立软件质量保证的整体概念。

(1)、关于复审、复查、管理复审和测试共12分，每方面给3分。

1)复审是搭在编码以前对分析、设计成果的非正规和正规审查，其重点是发现系统性的错误或缺点。

2)复查是指对阶段产生的文档和材料的检查，以保证下阶段工作的开始。

3)管理复查是指从项目管理的角度，从总体、成本和进度等方面进行检查。

4)测试是指对编码的查错和排错，应说测试的内容和过程，如单元测试、集成测试、系统测试等。

(2)、软件复审是软件在编码前对分析文档和设计文档的审查，其目的是发展和纠正在分析和设计阶段中可能产生的系统性错误，它是软件测试的固有和重要内容和步骤。软件测试是对基于正确设计基础上所开发的程序的测试。二者是从不同方面对软件质量的保证。

涉及到上述内容即可得6分，缺少可适当扣分。

(3)、软件测试的目的从编码阶段来说是发现程序中可能出现的错误并排除错误。测试的具体内容是从不同范围和对象中来发现可能存在的错误并排除之，包括：单元测试即对模块进行测试，再对由模块集成的子系统进行测试，再将子系统集成起来进行系统测试，测试中将应用到测试实例和测试数据。涉及到上述内容即可得12分，部分者可适当扣分。

4.(2002年)**软件维护**是软件生命周期的重要组成，而影响维护的因素较多，除了人员因素、技术因素和管理因素外，还有程序自身的因素，具体影响因素可归纳为：

**1)软件系统的规模;**

**2)软件系统的年龄;**

**3)软件系统的结构;**

试回答以下问题：

(1).给出上述三项因素的解释，阐述“规模”、“年龄”和“结构”的具体含义。(10分)

(2).扼要分析影响维护工作量的因素，即分析可能增加维护工作量的因素和可能减少维护工作量的因素。(10分)

(3).简单叙述软件可维护性度量的含义。(10分)

答案:

(1).系统的规模。指软件系统规模的大小，系统规模越大，维护困难越大。

系统的年龄。系统运行时间越长，可能经过多次修改，从而造成维护的困难。

系统的结构。系统结构合理与否，会给维护带来困难。(符合上述要求的给10分，不足者可适当减分。)

(2).增加维护工作量的因素有：

软件系统的年龄和规模、结构合理性、程序复杂性、用户的数量、应用的变化和文档的质量。

减少维护工作量的因素有：

结构化技术的使用，自动化工具使用，数据库技术应用，高性能软件使用，新的软件技术的应用。(满足上述内容的给10分，否则根据回答的比例适当减分。)

(3).可维护性度量表示软件系统维护工作的强度或维护工作量的大小。人们为了度量可维护性，常将可维护性度量问题分解为对：可测试性、可理解性、可修改性、可移植性、可靠性、有效性和可用性的度量。(前三项每项2分，后四项每项1分。)

5.(2003年)

**(1).试划分软件维护阶段所执行的任务的归类。(注意，并不需要罗列软件维护的细节内容)(15分)**

**(2).试陈述软件维护管理所涉及的四个方面，并简述各方面所包括的具体内容。(15分)**

答案:

(1).本题属于软件维护技术的相关内容，在软件维护阶段的任务归类中，将软件维护阶段的任务执行分为三部分：理解现有系统;修改现有系统;重新确认修改部分。

(2).软件维护管理包括四部分内容，它们是：

1)控制程序的改变

①论证程序改变的合理性改变来自多方面，要分析其要求的合理性，并不一定接收所有的改变要求。并应注意：

Ⅰ.维持目标中版本的选择方式，是单一版本，还是多用户版本。

Ⅱ.近期是否有替换该程序的计划，如有可推迟改变。

Ⅲ.是否改变程序的范围和目标，如是，则应慎重决定。

②对改变进行控制，除紧急任务外，其改变需按计划执行，改变控制应包括：

Ⅰ.研究程序如何改变，并安排计划。

Ⅱ.向改变申请者提供报告，改变计划或拒绝理由。Ⅲ.执行计划。

2)程序质量检查

①保证程序设计标准。②对改变过程进行检查。

3)安排维护计划

①研究改变申请，确定实现改变的可行性和费用。②制定维护计划应考虑的因素：

Ⅰ.程序的稳定(持久)性。Ⅱ.程序的错误类型。Ⅲ.申请改变的用户数量。

4)建立维护机构

①建立专职维护结构--维护小组。②设立专职维护人员。③制定维护政策。

6.(2004年)软件工程项目在实施中必须进行科学的管理，以保证工程的进度和质量。软件的配置管理是软件管理的重要内容之一。试述：

**(1)软件配置和软件配置管理的概念。(12分)**

**(2)列举10个软件配置的配置项。(10分)**

**(3)软件配置管理的基本任务是什么?(8分)**

答案:

(1)一个完整的软件系统的构成就是软件配置.(4分)

软件配置管理是软件生命周期内管理变化的一组活动.包括:

1)标识变化;2)控制变化;

3)确保实现了变化;4)报告变化;(4分)

软件在设计,生产和使用中,会产生不同的版本.软件配置管理应对不同的软件版本进行标识和管理.(4分)

(2)软件配置由配置项组成,软件配置项有:

·系统规格说明书、·项目开发计划、·软件需求说明书、·原型系统

·总体(或概要)设计说明书·详细设计说明书·源程序清单·测试计划

·测试报告·操作手册·用户手册·目标程序·软件问题报告

·维护请求·软件变更通知·软件工程标准·项目开发总结

(以上配置每回答项出1个给1分,越过10个最多给10分)

(3)软件配置管理的基本任务包括:

1)制定文件书写标准和标识方法.(2分)

2)控制配置变更,记录与评价软件配置的.(2分)

3)实现对已完成文档的存取管理和跟踪版本变化.(2分)

4)对版本进行检查和维护.(2分)

**1.(2003年)**

**(1).试划分软件维护阶段所执行的任务的归类。(注意，并不需要罗列软件维护的细节内容)(15分)**

**(2).试陈述软件维护管理所涉及的四个方面，并简述各方面所包括的具体内容。(15分)**

答案:

(1).本题属于软件维护技术的相关内容，在软件维护阶段的任务归类中，将软件维护阶段的任务执行分为三部分：理解现有系统;修改现有系统;重新确认修改部分。

(2).软件维护管理包括四部分内容，它们是：

1)控制程序的改变

①论证程序改变的合理性改变来自多方面，要分析其要求的合理性，并不一定接收所有的改变要求。并应注意：

Ⅰ.维持目标中版本的选择方式，是单一版本，还是多用户版本。

Ⅱ.近期是否有替换该程序的计划，如有可推迟改变。

Ⅲ.是否改变程序的范围和目标，如是，则应慎重决定。

②对改变进行控制，除紧急任务外，其改变需按计划执行，改变控制应包括：

Ⅰ.研究程序如何改变，并安排计划。

Ⅱ.向改变申请者提供报告，改变计划或拒绝理由。

Ⅲ.执行计划。

2)程序质量检查

①保证程序设计标准。 ②对改变过程进行检查。

3)安排维护计划

①研究改变申请，确定实现改变的可行性和费用。

②制定维护计划应考虑的因素：

Ⅰ.程序的.稳定(持久)性。Ⅱ.程序的错误类型。Ⅲ.申请改变的用户数量。

4)建立维护机构

①建立专职维护结构--维护小组。②设立专职维护人员。③制定维护政策。

2.(2004年)

软件工程项目在实施中必须进行科学的管理，以保证工程的进度和质量。软件的配置管理是软件管理的重要内容之一。试述：

**(1)软件配置和软件配置管理的概念。(12分)**

**(2)列举10个软件配置的配置项。(10分)**

**(3)软件配置管理的基本任务是什么?(8分)**

答案:

(1)一个完整的软件系统的构成就是软件配置.(4分)

软件配置管理是软件生命周期内管理变化的一组活动.包括:

1)标识变化;2)控制变化;3)确保实现了变化;4)报告变化;(4分)

软件在设计,生产和使用中,会产生不同的版本.软件配置管理应对不同的软件版本进行标识和管理.(4分)

(2)软件配置由配置项组成,软件配置项有:

·系统规格说明书·项目开发计划·软件需求说明书

·原型系统·总体(或概要)设计说明书·详细设计说明书

·源程序清单·测试计划·测试报告·操作手册

·用户手册·目标程序·软件问题报告·维护请求·软件变更通知

·软件工程标准·项目开发总结

(以上配置每回答项出1个给1分,越过10个最多给10分)

(3)软件配置管理的基本任务包括:

1)制定文件书写标准和标识方法.(2分)

2)控制配置变更,记录与评价软件配置的.(2分)

3)实现对已完成文档的存取管理和跟踪版本变化.(2分)

4)对版本进行检查和维护.(2分)

计算机四级软件工程论述题真题及答案

4. （2002 年）

软件维护是软件生命周期的重要组成，而影响维护的因素较多，除了人员因素、技术因素和管理因素外，还有程序自身的因 素，具体影响因素可归纳为:

1） 软件系统的规模;

2） 软件系统的年龄;

3） 软件系统的结构; 试回答以下问题：

**（1） •给岀上述三项因素的解释，阐述规模、年龄和结构的具 体含义。（10分）**

**（2） .扼要分析影响维护工作量的因素，即分析可能增加维护 工作量的因素和可能减少维护工作量的因素。（10分）**

**（3） •简单叙述软件可维护性度量的含义。（10分）**

答案：

(1) .系统的规模。指软件系统规模的大小，系统规模越大， 维护困难越大。

系统的年龄。系统运行时间越长，可能经过多次修改，从而 造成维护的困难。

系统的结构。系统结构合理与否，会给维护带来困难。(符 合上述要求的给10分，不足者可适当减分。)

(2) .增加维护工作量的因素有：

软件系统的年龄和规模、结构合理性、程序复杂性、用户的 数量、应用的变化和文档的质量。

减少维护工作量的因素有：

结构化技术的使用，自动化工具使用，数据库技术应用，高 性能软件使用，新的软件技术的应用。(满足上述内容的给10分, 否则根据回答的比例适当减分。)

(3) .可维护性度量表示软件系统维护工作的强度或维护工作 量的大小。人们为了度量可维护性，常将可维护性度量问题分解 为对：可测试性、可理解性、可修改性、可移植性、可靠性、有 效性和可用性的度量。(前三项每项2分，后四项每项1分。)

**什么是软件维护？**

软件维护（Software maintenance）是一个软件工程名词，是指在软件产品发布后，因修正错误、提升性能或其他属性而进行的软件修改。

软件维护活动类型总起来大概有四种：纠错性维护（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。除此四类维护活动外，还有一些其它类型的维护活动，如：支援性维护（如用户的培训等）。

改正性维护是指改正在系统开发阶段已发生而系统测试阶段尚未发现的错误。这方面的维护工作量要占整个维护工作量的17%～21%。所发现的错误有的不太重要，不影响系统的正常运行，其维护工作可随时进行：而有的错误非常重要，甚至影响整个系统的正常运行，其维护工作必须制定计划，进行修改，并且要进行复查和控制

适应性维护是指使用软件适应信息技术变化和管理需求变化而进行的修改。这方面的维护工作量占整个维护工作量的18%～25%。由于计算机硬件价格的不断下降，各类系统软件屡出不穷，人们常常为改善系统硬件环境和运行环境而产生系统更新换代的需求；企业的外部市场环境和管理需求的不断变化也使得各级管理人员不断提出新的信息需求。这些因素都将导致适应性维护工作的产生。进行这方面的维护工作也要像系统开发一样，有计划、有步骤地进行。

完善性维护是为扩充功能和改善性能而进行的修改，主要是指对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能与性能特征。这些功能对完善系统功能是非常必要的。另外，还包括对处理效率和编写程序的改进，这方面的维护占整个维护工作的50%～60%，比重较大．也是关系到系统开发质量的重要方面。这方面的维护除了要有计划、有步骤地完成外．还要注意将相关的文档资料加入到前面相应的文档中去。

预防性维护为了改进应用软件的可靠性和可维护性，为了适应未来的软硬件环境的变化，应主动增加预防性的新的功能，以使应用系统适应各类变化而不被淘汰。例如将专用报表功能改成通用报表生成功能，以适应将来报表格式的变化。这方面的维护工作量占整个维护工作量的4%左右。

**什么是交互设计师、交互设计？有哪些特点**

交互设计师（InteractionDesign）是指参与完成对产品与它的使用者之间的互动机制进行分析、预测、定义、规划、描述和探索的过程的设计师，即定义人造物的行为方式的工作者。

交互指的是产品与它的使用者之间的互动过程，而交互设计师则是秉承以用户为中心的设计理念，以用户体验度为原则，对交互过程进行研究并开展设计的工作人员。

交互设计是以一种目标导向设计，所有的工作内容都是围绕着用户行为去设计的。交互设计师通过去设计用户的行为，让用户更方便更有效率的去完成产品的目标任务，获得愉快地用户体验。

可用性（usability）是交互设计的基本而且重要的指标，它是对可用程度的总体评价。也是从用户角度衡量产品是否有效、易学、安全、高效、好记、少错的质量指标。

同时，交互设计（Interaction Design）的目标不止于此，它还包括要考虑用户的期望（Donald Norman先生说到期望设计是交互设计的下一站）和体验，可用性保证产品可用，基本功能完备且方便；而体验在于给用户一些与众不同的或者意想之外的感觉。也就是说，可用，是产品应该做到的，理所应当的，体验则是额外的惊喜和收获。

ISO9241/11中的定义是：可用性是一个多因素概念，涉及到容易学习、容易使用、系统的有效性、用户满意，以及把这些因素与实际使用环境联系在一起针对特定目标的评价 [4] 。

**等价分类法的基本思想？**

等价分类法是指一种典型的、重要的黑盒测试方法，它将程序所有可能的输入数据划分为若干个等价类。然后从每个部分中选取具有代表性的数据当做测试用例。测试用例由有效等价类和无效等价类的代表数据组成，从而保证测试用例具有完整性和代表性。根据程序的输入特性，将程序的定义域划分为有限个等价区段“等价类”，从等价类中选择出的用例具有代表性，即测试某个等价类的代表值据等价于对这一类其他值得测试。若果某个等价类的一个输入数据测试中查出了错误，说明该类中其他测试用例也会有错误。

**什么是模块独立性？如何度量**

模块独立性是指每个模块只完成系统要求的独立的子功能（即：功能单一），并且与其他模块的联系最少且接口简单（比如：只有一个接口）。模块的独立程度可以由两个定性标准度量；内聚和耦合耦合衡量不同模块之间互相依赖的紧密程度。

但是，在一个系统的模块结构中没有哪两个模块可以完全独立，所以，要力争模块之间尽量独立，以得到一个质量良好的模块结构。一般采用两个准则衡量软件的模块独立性：模块间的耦合，模块的内聚

一、

**1、简述软件的概念。**

软件是计算机程序、数据结构和文档的有机结合。

**2、什么是软件危机？**

落后的软件生产方式无法满足迅速增长的计算机软件需求，从而导致软件开发与维护过程中出现一系列严重问题的现象。这种现象早在20世纪60年代被定义为“软件危机”。

**3、软件危机主要包含哪两方面的问题？**

一是如何开发软件，以满足对软件日益增长的需求；二是如何维护数量不断膨胀的已有软件。

**4、简述软件危机产生的原因。**

软件危机产生原因是由于软件产品本身的特点以及开发软件的方式,方法,技术和人员引起的：a）软件的规模越来越大，结构越来越复杂； b）软件开发的管理困难而复杂； c）软件开发费用不断增加； d）软件开发技术落后； e）生产方式落后；f）开发工具落后,生产率提高缓慢。

**5、什么是软件工程？**

软件工程就是用科学的知识和技术原理来定义，开发，维护软件的一门学科。或

软件工程是将系统性的、规范化的、可定量的方法应用于软件开发、运行和维护。

**6、简述软件生存周期的概念。**

软件生命周期由软件定义、软件开发、软件运行（维护）三个时期组成。

定义阶段的主要任务是解决“做什么”的问题。通常被分为三个阶段：问题定义、可行性研究、需求分析。

开发阶段的任务是解决“如何做”的问题。通常包括软件设计，实现，测试等工作。

维护阶段的主要任务是使软件保持用户满意的需要。

**7、软件工程的目标和内容是什么？**

软件工程目标：软件工程旨在开发满足用户需要、及时交付、不超过预算和无故障的软件。

其主要内容如下：a)实现预期的软件功能，达到较好的软件性能，满足用户的需求；b)增强软件过程的可见性和可控性，保证软件的质量；c)提高所开发软件的可维护性，降低维护费用；d)提高软件开发生产率，及时交付使用；e)合理预算开发成本，付出较低的开发费用。

二、

**1、什么是项目？项目有哪两种常见形式？**

项目是在一定时间内满足一系列特定目标的多项相关任务的总称。项目包括通用商业软件项目和定制软件项目。

**2、简述项目的开发流程？**

答：招标、投标、合同签订、软件项目开发。其中软件项目开发分成若干阶段，用软件生命周期表示，每个阶段包括规定的明确任务，建立里程碑，以及对里程碑进行验证。

**3、简述软件生命周期各个阶段？**

项目计划阶段、需求分析阶段、系统设计阶段、编码测试阶段、综合测试阶段和运行维护阶段。

**4、列举项目的主要角色？**

机构领导，项目经理，需求分析员，系统设计师、程序员、测试员、配置管理员、质量保证员和产品维护人员。

**5、典型的项目组织方式有哪些？**

民主制程序员组、主程序员组和现代程序员组。民主制程序员组：适合项目规模小，以2～8开发人员为宜组内多数成员经验丰富、技术熟练减少通信问题，容易制定小组的质量标准。 主程序员组：主程序员、后备程序员和编程秘书。现代程序员组：技术负责人和行政负责人。

**6、简述检查点、里程碑、基线三者的关系。**

检查点比较细，里程碑比较粗，基线最粗；重要的检查点一般是里程碑，重要的里程碑一般就是基线。在实际项目中，周例会时检查点的表现形式，高层的阶段汇报会是基线的表现形式。

**7、软件生命周期计划阶段的主要任务？**

根据商业计划确定项目开发的总目标，估计项目需要的资源和软件项目成本，制定项目进度表等。

**8、软件生命周期需求分析阶段的主要任务？**

确定系统“做什么”，即目标系统具备的功能需求及性能需求，通过充分有效地沟通，与用户达成一致，提交软件需求规格说明书和用户界面原型。

**9、软件生命周期编码测试阶段的主要任务？**

包括编码和单元测试，按照编程规范，把设计结果写成正确的、容易理解和维护的代码，认真测试每一个模块。

三、

**1、可行性研究的要素包括哪些？**

经济，技术，社会，人和法律。

**2、可行性研究的任务是什么?**

可行性研究的任务是对以后的行动方针提出建议。如果问题没有可行的解，分析员应该建议停止这项开发工程，以避免时间、资源、人力和金钱的浪费。如果问题值得解，分析员应该推荐一个较好的解决方案，并且为工程制定一个初步的计划。

**3、什么是系统流程图？**

系统流程图简称流程图或系统图，是概括地描绘物理系统的传统工具。它的基本思想是用图形符号以黑盒子形式描绘组成系统的每个部件(程序，文档，数据库，人工过程等)。系统流程图表达的是数据在系统各部件之间流动的情况，而不是对数据进行加工处理的控制过程。

**4、可行性研究有哪些步骤及内容?**

（1）复查项目目标和规模：访问关键人员，仔细阅读分析材料，对问题定义阶段书写的规模目标报告书进一步复确认。改正含糊或不准确，清晰描述对目标系统的一切限制和约束；（2）研究目前系统：了解现有系统基本功能，保证新开发系统的功能性。了解现有系统缺点，保证新开发系统改善缺陷。了解现有系统运行使用所需费用，新系统应增加收入或减少使用费用。了解现有系统能做什么，无需了解他怎样做这些工作；（3）建立新系统的高层逻辑模型：系统流程图，数据流图和数据字典；（4）重新定义问题：分析员与用户以数据流图和数据字典为讨论基础，再次复查问题定义、工程规模和目标。如对问题有误解，或遗漏要求，在此阶段改正。

**5、简述软件质量的含义。**

软件质量是“软件与明确地和隐含地和定义的需求相一致的一种程度”。

**6、度量软件规模的常用技术主要有哪些？**

LOC(line of code,代码行)估算、FP(功能点)估算、DeLphi估算法和类比法。

**7、简述成本估算的步骤。**

（1）任务分解

（2）估算每个任务的成本

（3）直接估算成本

（4）间接估算成本

（5）项目总成本估算

（6）项目总报价

**8、简述风险及风险管理的目标。**

所有可能危害项目的因素都称之为风险，风险事件最终可能发生，也可能不发生，风险管理的目的是在风险产生危害之前识别它们，从而有计划地消除或削弱风险。

**9、简述风险管理的主要过程。**

风险识别，风险量化，风险应对计划制定和风险监控。

**10、简述什么是CMM。**

能力成熟度模型(capability maturity model，CMM)是用于评价软件机构的软件过程能力熟练度的模型。分为初始级，可重复级，已定义级，已管理级和优化级5个等级。CMM的策略是力图改进对软件过程的管理，而在技术方面的改进是其必然的结果。

**11、简述工程网络图的绘制步骤。**

（1）画工程网络图。

（2）计算事件的最早时刻。

（3）计算事件的最迟时刻。

（4）确定工程的关键路径。

（5）计算每个子任务的机动时间。

六、

**1、为什么要进行代码审查？**

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，可以查出最多达70％的逻辑设计错误和编码错误。

**2、简述代码审查的组织形式和内容？**

代码审查的形式主要有几种:同行评审；管理评审；单人评审；代码检查。

代码的审查内容主要有：代码的编写规范（注释的书写格式、命名规范、代码组织结构）、技术处理规范（异常处理、日志处理）、业务实现等。

**3、代码审查的步骤？**

制定评审计划； 评审准备； 评审会议； 对评审结果采取行动；

评审结果跟踪； 提交和归档。

七、

**1、进行软件测试的目的？**

目的是通过软件测试以最少的时间和人力找出软件中潜在的各种错误和缺陷。

**2、简述什么是软件测试？**

软件测试就是利用测试工具按照测试方案和流程对产品进行功能和性能测试，甚至根据需要编写不同的测试工具，设计和维护测试系统，对测试方案可能出现的问题进行分析和评估。执行测试用例后，需要跟踪故障，以确保开发的产品适合需求。

**3、简述什么是软件质量？**

软件质量就是“软件与明确的和隐含的定义的需求相一致的程度”。具体地说，软件质量是软件符合明确叙述的功能和性能需求、文档中明确描述的开发标准、以及所有专业开发的软件都应具有的隐含特征的程度。

**4、简述什么是静态测试？**

不运行被测程序本身，仅通过分析或检查源程序的语法、结构、过程、接口等来检查程序的正确性。

**5、简述什么是动态测试？**

通过运行被测程序，检查运行结果与预期结果的差异，并分析运行效率和健壮性等性能。目前，动态测试是公司测试工作的主要方式。

**6、软件测试原则有哪些？**

（1）尽早地和不间断地进行软件测试

（2）测试用例应由测试输入数据和与之对应的预期输出结果两部分组成

（3）程序员应避免检查自己的程序

（4）在设计测试用例时，应当包括合理的输入条件和不合理的输入条件

（5）充分注意测试中的群集现象。

（6）严格执行测试计划，排除测试的随意性

（7）应当对每一个测试结果做全面检查

（8）妥善保管测试计划、测试用例、出错统计和最终分析报告，为维护提供方便

八、

**1、什么是软件维护？**

软件维护主要是指根据需求变化或硬件环境的变化对应用程序进行部分或全部的修改，修改时应充分利用源程序。修改后要填写程序修改登记表，并在程序变更通知书上写明新旧程序的不同之处。

**2、什么是软件的可维护性？**

软件的可维护性可以定义为：软件能够被理解、改正、适应和完善以适应新的环境的难易程度。

**3、软件维护的目标是什么？**

改正性维护：为了改正软件系统中的错误，使软件能够满足预期的正常运行状态的要求而进行的维护。

适应性维护：为了使软件适应内部或外部环境变化，而去修改软件的过程。

完善性维护：满足使用过程中用户提出增加新功能或修改已有功能的建议维护。

预防性维护：为了提高软件的可维护性、可靠性等，为以后进一步改进软件打下良好基础而修改软件的活动。

**结对编程的意义及如何结对编程**

每人在各自独立设计、实现软件的过程中不免要犯这样那样的错误。在结对编程中，因为有随时的复审和交流，程序各方面的质量取决于一对程序员中各方面水平较高的那一位。这样，程序中的错误就会少得多，程序的初始质量会高很多，这样会省下很多以后修改、测试的时间。具体地说，结对编程有如下的好处：

（1）在开发层次，结对编程能提供更好的设计质量和代码质量，两人合作能有更强的解决问题的能力。

（2）对开发人员自身来说，结对工作能带来更多的信心，高质量的产出能带来更高的满足感。

（3）在心理上, 当有另一个人在你身边和你紧密配合, 做同样一件事情的时候, 你不好意思开小差, 也不好意思糊弄。

（4）在企业管理层次上，结对能更有效地交流，相互学习和传递经验，能更好地处理人员流动。因为一个人的知识已经被其他人共享。

（1）驾驶员：写设计文档，进行编码和单元测试等 XP 开发流程。

（2）领航员：审阅驾驶员的文档、驾驶员对编码等开发流程的执行；考虑单元测试的覆盖程度；是否需要和如何重构；帮助驾驶员解决具体的技术问题。

（3）驾驶员和领航员不断轮换角色，不宜连续工作超过一小时。领航员要控

制时间。

（4）主动参与。任何一个任务都首先是两个人的责任，也是所有人的责任。没有“我的代码”、“你的代码”或“她的代码”，只有“我们的代码”。

（5）只有水平上的差距，没有级别上的差异。尽管可能大家的级别资历不同，但不管在分析、设计或编码上，双方都拥有平等的决策权利。