

在信息处理和计算机领域内，一般认为软件是 文库 程序____、_ 文档____ 和_ 数据____ 。

数据流图的基本组成部分有 _ 数据的源点与终点____、数据流 ____、加工 、数据文件。

数据流图和数据字典共同构成了系统的 _逻辑 ____模型，是需求规格说明书的主要组成部分。

划分模块时尽量做到__ 高内聚、低耦合____，保持模块的独立性，尽量使用公共模块。 5. 类的实例化是_对象 _____。

人们常用硬件可靠性的定量度量方法来度量软件的可靠性和可用性，常用的度量软件可

靠性的两个指标是_成功地运行的概率 _____和 平均故障时间 _____。

将待开发的软件细化，分别估算每一个子任务所需要的开发工作量，然后将它们加起来，

将得到软件的总开发量。这种成本估算方法称为_ 自底向上 _____。

如果一个模块被 n 个模块调用，其中直接的上级模块的个数是 m 个（ $m \leq n$ ）那么该模块的扇入数是 __ N ____ 个。

结构化设计以 __数据流图 _____ 为基础，按一定的步骤映射成软件结构。

软件的风险分析可包括风险识别、风险预测 和风险驾驭（或风险管理）等 3 项活动。 13. 软件著作权登记的 3 种主要类型：（1） 著作权登记；（2）著作权延续登记；（3）权利转移备案登记。

软件工程管理的主要内容有：项目经费管理， 软件质量管理，项目进度管理和人员管理。

面向对象分析的目的是对客观世界的系统进行 __建模 _____ 。

软件维护工作的生产性活动包括分析评价、修改设计和 __编写程序代码 _____ 等。

为了使应用软件适应计算机环境的变化而 __ 修改软件_____ 的过程称为适应性维护。

一个进行学生成绩统计的模块其功能是先对学生的成绩进行累加，然后求平均值，则该模块的内聚性是 __顺序内聚 。

投资回收期就是使累计的经济效益等于 __

最初的投资费用_____ 所需的时间。

软件生存周期是指一个软件从提出开发要求开始直到

软件废弃_____为止的整个时期。

曾被誉为“程序设计方法的革命”的 结构化

程序设计,使程序设计从主要依赖于程序员个人的自由活动变成为有章可循的一门科学。

从结构化程序设计到 面向对象程序设计

，是程序设计方法的又一次飞跃。

在单元测试时，需要为被测模块设计__测试用例 ____。 名词解释

数据词典——是描述数据信息的集合,它对数据流图中的各个元素按规定格式进行详细的描述和确切的解释,是数据流图的补充工具。

数据流图——他以图形的方式反映系统的数据流程

白盒测试——按照程序内部的结构测试程序,检验程序中的每条路径是否都能按预定要求正确工作。有两种测试法既逻辑覆盖测试法和路径测试法

黑盒测试——按照程序的功能测试程序,检验与程序功能有关的输入、输出与程序执行是否正确。有四种方法既等价分类法、边界值分析法、错误猜测法和因果图法

完善性维护——为了适应用户业务和机构的发展变化而对软

件的功能、性能进行修改、扩充的过程称为完善性维护。因为各种用户的业务和机构在相当长的时期内不可能是一成不变的,所以功能、性能的增加是不可避免的,而且这种维护活动在整个维护工作中所占的比重很大

软件可靠性——指在给定的时间内,程序按照规定的条件成功地运行的概率

软件配置——是一个软件在生存周期内,他的各种形式、各种版本的文档与程序的总称

软件再工程——运用逆向工程、重构等技术,在充分理解原有软件的基础上,进行分解、综合、并重新构建软件,用于提高软件的可理解性、可维护性可复用性或演化性。

α 测试——是在一个受控的环境下,由用户在开发者的“指导”下进行的测试,

由开发者负责记录错误和使用中出现的问题。

β 测试——是由软件的最终用户（多个）在一个或多个用户场所来进行。由用户负责记下遇到的所有问题，包括主观认定的和真实的问题，定期向开发者报告，开发者在综合用户的报告之后进行修改，最后将软件产品交付给全体用户使用。

11. 聚集关系——表示类或对象之间的整体与部分的关系 12. 泛化关系——表示类或对象之间的一般与特殊的关系

内聚——一个模块内部各个元素彼此结合的紧密程度的度量。

耦合——一个软件结构内不同模块之间互连程度的度量。

软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。

(软件过程) 是为了获得高质量软件所需要完成的一系列任务的框架。

系统流程图实质上是 (物理) 数据流图。

ER 模型可以作为 (用户) 与分析员之间有效的交流工具。

第二范式满足第一范式条件，而且每个非关键字属性都由整个关键字决定。

并发系统中遇到的一个主要问题是 (定时) 问题。

形式化的规格说明消除了 (二义性)，从而可以减少差错。

需求分析阶段得出的 (数据流图) 是总体设计的极好的出发点。

软件工程师应该在需求分析阶段所确定的系统数据需求的基础上，进一步设计 (数据库)。

面向数据流的设计方法是把信息流映射成 (软件结构)

程序的质量和 GO TO 语句的数量成 (反比)。

(易变性) 指系统响应时间相对于平均响应时间的偏差。

人机界面设计是 (接口) 设计的一个重要组成部分。

通常把 (编码) 和测试统称为实现。

(黑盒) 测试法把程序看做是一个黑盒子，完全不考虑程序的内部结构和处理过程

(验收) 测试也称为确认测试。

集成测试主要目标是发现与 (接口) 有关的问题。

通常 (白盒) 测试在测试过程的早期阶段进行。

软件 (可靠性) 是程序在给定的时间间隔内，按照规格说明书的规定成功运行的

概率

通常把诊断和改正错误的过程称为（改正）性维护。

传统的软件开发方法以算法为核心，开发过程基于功能分析和（功能分解）。

面向对象的软件技术在利用（可重用）的软件成分构造新的软件系统时，有很大的灵活性。

在 C++ 中，多态性是通过（虚函数）来实现的。

如果在聚集关系中处于部分方的对象可以同时参与多个处于整体方对象的构成，则该聚集称为（共享）聚集。

与多重继承相反的是（不相交）继承。

面向对象方法学认为，客观世界由（对象）组成。

在用面向对象观点建立起的 3 种模型中，（对象模型）是最基本，最重要，最核心的。

（主题）是指导读者理解大型、复杂模型的一种机制。

确定了类中应该定义的属性之后，就可以利用（继承）机制共享公共性质。

对于仅存储静态数据的系统来说，（动态）模型并没有什么意义，但它在开发交互式系统时却起着很重要的作用。

我们可以把面向对象设计分为系统设计和（对象）设计

由于类的封装性，使用者无须了解实现细节就可以使用适当的（构造）函数，按照需要创建类的实例。

把子系统组织成完整的系统时，有水平层次组织和（垂直块）组织两种方案可供选择

在面向对象设计过程中，应该对系统的人机（交互）子系统进行详细设计

关系数据库管理系统的理论基础是（关系代数）。

面向对象分析得出的对象模型，通常并不详细描述类中的（服务）。

如果关联的重数是多元的，则需要用一个（指针集合）实现关联。

可以通过增加（冗余）关联提高访问效率。

仅当存在真实的一般-特殊关系时，利用（继承）机制实现行为共享才是合理的。

更完整的类库通常还提供了独立于具体设备的（接口）类。

所谓（参数）类，就是使用一个或多个类型去参数化一个类的机制。

测试面向对象的软件时，应该把操作作为（类）的一部分来测试。

开始集成面向对象系统以后，必须对（类间协作）进行测试。

（软件项目管理）先于任何技术开始之前开始，并且贯穿于软件的整个生命周期之中。

如果把项目（持续时间）延长一些，则可降低完成项目所需的工作量。

（云班）按照(软件开发工作的主体)，可以将软件生存周期过程分为基本过程、支持过程和组织过程。

软件开发和维护过程中存在很多严重问题，这与软件的规模有关。（×）

导出软件结构属于典型的可行性研究过程（×）（知识点）（1）确定项目规模和目标（2）研究正在运行的系统（3）建立新系统的逻辑模型（4）导出和评价各种方案（5）推荐可行性方案（6）编写可行性研究报告（7）递交可行性研究报告

面对复杂的系统时，一个比较好的方法是分层次地描绘这个系统。（√）

数据字典是开发数据库的第一步，而且是很有价值的一步。（√）

层次方框图不能代表完整的数据结构。（×）

不需要验证需求的现实性（×）

所谓形式化方法，是描述系统性质的基于数学的技术。（√）

总体设计过程通常由两个主要阶段组成，其中系统设计阶段确定软件结构（×）

简单地说，理想的内聚模块只做一件事情。（√）

一个模块的扇出表明有多少个上级模块直接调用它。（×）

层次图方框间的连线表达的是组成关系。（×）

结构图中箭头尾部是实心圆表示传递的是控制信息。（√）

面向数据结构的设计方法的目标是给出设计软件结构的一个系统化的途径。（×）

优化是先使它快起来，再使它工作起来。

任何流图的环形复杂度都等于判定节点的数目加 1。（×）

软件测试的目的是发现并改正错误。（×）

调试的目标是充分暴露程序当中的错误。（×）

所有测试都应该追溯到用户的需求（√）

白盒测试又称为功能测试。(×) (知识点) 白盒测试也称为结构测试, 主要用于检测软件编码过程中的错误。黑盒测试又称为功能测试, 主要检测软件的每一个功能是否能够正常使用。

如果一个人是程序的设计者又是编写者, 或是程序的编写者又是测试者, 则审查小组应该再增加一个程序员。(√)

Alpha 测试由软件的最终用户们在一个或多个客户场所进行。(×)

类中定义的方法, 是允许施加于该类对象上的操作。(√)

对面向对象的软件进行维护, 主要通过从已有类派生出一些新类来实现。(√)

对象本质上具有并行性。(√)

在 C++ 中, 函数重载是通过动态联编来实现的。(×)

细化关系描述了两个模型元素之间的语义连接关系。(×)

实际上, 用面向对象方法开发软件时, 阶段的划分是十分模糊的。(√)

系统分析员必须把需求与实现策略区分开 (√)

需求陈述应该阐明怎么做而不是做什么。(×)

.在建立对象模型的过程中, 服务一定要确定下来。(×)

事件追踪图中箭头线之间的距离表示两个事件之间的精确时间差。(×)

我们应该降低继承耦合的强度 (×) (知识点) 与交互耦合相反, 继承的耦合程度要高。

紧密的继承耦合与高度的一般-特殊内聚是不一致的。(×)

使用面向对象方法开发软件时, 在分析和设计之间并没有明确的分界线。(√)

在提高可扩充性的过程中, 应该用一种方法遍历多条关联链。(×)

面向对象系统的确认测试是面向白盒的。(×)

正确性证明过程本身也可能发生错误。(√)

(云班) 支持过程是指那些与软件生产组织有关的活动集。(×)

(云班) 软件生存周期的含义是表明软件的演化和发展具有周期性, 每过一段时间就需要更新。(×)

(云班) 当软件的一个版本被淘汰时, 则该软件的生存周期就结束了 (×)

(云班) 软件生存周期过程表示软件生存周期内的一个过程。(×)

(云班) 软件生存周期模型是对软件生存周期中过程、活动和任务的组织。(×)

- (云班) 软件开发模型是软件开发过程的活动和任务的结构框架。(×)
- (云班) 软件生存周期模型必须要将所有类型的过程、活动和任务考虑在内，形成满足每个软件项目需要的模型。(×)
- (云班) 模型一定是在某种特定意图下，从某种特定的角度物理环境的抽象(√)
- (云班) 软件开发中分层的基本动机时空之开发的复杂性 (√)
- (云班) 相比硬件而言，软件更容易被修改，而且更容易被正确的进行修改(×)
- (云班) 采用瀑布模式的开发过程是一种自顶向下的开发方法，而软件构件复用的开发过程是一种自底向上的开发方法 (√)
- (云班) 任何软件开发过程必须从软件需求入手 (√)
- (云班) 软件需求时代开发产品或系统的功能描述 (×)
- (云班) 质量属性必须要给出量化的测量指标 (√)
- (云班) 非功能需求必须依附于功能需求而存在 (√)
- (云班) 面向对象就是一种使用面向对象程序设计语言的程序开发方法 (×)
- (云班) 一个类中可以有多个操作，也可以没有操作 (√)
- (云班) 所有的类都必须与客观事物相对应，都必须有具体的实例 (×)
- (云班) 接口之间没有关联、泛化、实现和依赖，也不能参与泛化、实现和依赖关系 (×)
- (云班) 接口不仅可以描述类的外部可见操作，而且可以描述其内部结构 (×)
- (云班) 软件开发中分层的基本动机是控制开发的复杂性 (√)
- (云班) 支持过程是指那些与软件生产组织有关的活动集 (×)
- (云班) 软件生存周期模型是对软件生存周期中过程、活动和任务的组织 (√)
- (云班) 包间的依赖通常隐含了各包种元素之间存在的一个或多个依赖 (√)
- (云班) 包可以包含其他种类的模型元素，但本身不能嵌套在其他包中 (×)
- (云班) 对于关联另一端的类的每个对象，本端的类只可能会有一个对象出现(×)
- (云班) 在类的一个关联中，可以显示地命名该角色 (√)
- (云班) 在关联关系中，聚合是组合的一种特殊形式 (×)
- (云班) 对于每一个关联关系都需要说明其多重性，如果不说名，则默认是 (√)
- (云班) 注解和约束不是类图的一部分 (×)
- (云班) 类图主要使用“类”来抽象系统中各个组成部分，不包括系统环境(×)

(云班)OOA 过程中建立模型规约可以集中进行,也可以分散在各个活动中(√)

(云班) OOA 的最终目的是产生一个只能直接反映问题域的 OOA 模型 (×)

(云班) 如果一个交互是由系统外的参与者发起的,第一个处理该监护的对象是主动对象 (√)

(云班) 与实现条件有关的对象是 OOA 阶段需要重点关注的对象 (×)

(云班) 在对象分类过程中,对于属性及操作相似的类需要考虑能够抽出一个一般类或部分类 (√)

(云班) 依赖关系是一种使用关系,如果被使用的类发生变化,那么另一个类的操作也会受到影响 (√)

(云班) 在 OOA 过程中,不需要考虑任何主动对象和主动操作 (×)

(云班) 一个类的属性必须适合这个类和他的全部特殊类的所有对象,同时,一个类中的操作也应适合这个类及其所有特殊类的每一个对象实例 (√)

(云班) OOA 过程中,为了提高操作的执行速度,可以考虑增加一些属性来保持操作的阶段性执行结果 (×)

(云班) 依赖关系是一种使用关系,如果被使用的类发生变化,那么另一个类的操作也会受到影响 (√)

(云班) 在初步建立类之间的关系时,可以暂时使用关联。在最终的类图中,若能用其他关系明确地指明类之间关系的含义,就不要适用关联 (×)

(云班) 关联关系可用动词或动宾结构来命名,关联的命名不可缺省 (×)

(云班) 两个对象之间的关联关系只能有一个 (×)

Q (云班) 与结构化的分析和设计类似,面向对象的分析和设计之间也存在一定的鸿沟 (×)

(云班) OOD 模型的四个部分均采用与 OOA 一致的概念、表示法、活动及策略 (×)

(云班) 所有的实现因素都能通过一些在 OOD 中新定义的独立组成部分而实行有效的隔离 (×)

(云班) 在 OOA 中未完成的细节定义需要在 OOD 完成,所以 OOD 的主要任务是对 OOA 的细化 (×)

(云班) OOA 强调如实地反映问题域,OOD 考虑实现问题,当所用语言不支持

多继承和多态，此时需要按编程语言调整多继承和多态。(×)

(云班)人机交互的设计需要重点考虑比例最大的适用人群的情况，并适当的兼顾其他人(√)

(云班)人机交互是 OOD 阶段的任务，在 OOA 阶段不需要考虑(×)

(云班)人机交互的交互内容需求不仅与系统的功能需求有关，而且与人的主观意识也有很大的关系(×)

(云班)人机交互部分的设计可以从 use case 抽取人机交互内容及过程，在抽取过程中需要删除所有与输入输出无关语句。(√)

1.系统流程图是描述物理模型的传统工具，用图形符号表示系统中各个元素表达了系统中各种元素之间的 信息流动 情况。

2.成本效益分析的目的是从 经济 角度评价开发一个项目是否可行。

3.自顶向下结合的渐增式测试法，在组合模块时有两种组合策略；深度优先策略和宽度优先策略。

4.独立路径是指包括一组以前没有处理的语句或条件的一条路径。从程序图来看，一条独立路径是至少包含有一条 其他独立路径中从未有过的 的边的路径。

5.汇编语言是面向 机器 的，可以完成高级语言无法完成的特殊功能。如与外部设备之间的一些接口工作。

6.单元测试一般以 白盒 测试为主，黑盒 测试为辅。

7.详细设计的任务是确定每个模块的内部特性，即模块的算法、使用的数据。

8.所有软件维护申请报告要按规定方式提出，该报告也称 软件问题 报告。

9...有两类维护技术:在开发阶段使用来减少错误、提高软件可维护性的面向维护的技术;在维护阶段用来提高维护的效率和质量的 维护支援 技术。

10.科学与工程计算需要大量的标准库函数，以便处理复杂的数值计算，可供选择的语言有:FORTRAN 语言、PASCAL 语言、C 语言和 PL/1 语言。

11.目前使用最广泛的软件工程学方法分别是传统方法学和 面向对象方法学

12.可行性研究主要集中在以下四个方面 经济可行性、技术可行性、法律可行性和抉择。

13.常见的软件概要设计方法有 3 大类:以数据流图为基础构造模块结构的结构化设计方法以数据结构为基础构造模块的 Jackson 方法，以对象、类、继

承和通信为基础的 面向对象的设计方法。

14. 数据流图 和 数据字典 共同构成系统的逻辑模型。

三、判断题

1.软件的开发与运行经常受到硬件的限制和制约。(T)

2.模块内的高内聚往往意味着模块间的松耦合。(T)

3.Jackson 图只能表达程序结构，不能表达数据结构。(F)

4.软件生命周期中需我进行需求分析，再进行可行性研究。(F)

5.软件的质量好坏主要由验收人员负责，其他开发人员不必关心。(F)

6.判定覆盖不一定包含条件覆盖，条件覆盖也不一定包含判定覆盖。(T)

7.应该尽量使用机器语言编写代码，提高程序运行效率，而减少高级语言的使用。
(F)

8.UML 只能应用于软件系统模型的建立。(F)

9.容错就是每个程序采用两种不同的算法编写。(F)

10.软件测试的目的是为了无一遗漏的找出所有的错误。(F)

11.在进行总体设计时应加强模块间的联系。(F)

12.系统结构图是精确表达程序结构的图形表示法。因此，有时也可以将系统结构图当作流程图使用。(F)

13.用黑盒法测试时，测试用例是根据程序内部逻辑设计的。(F)

14.在程序调试时，找出错误的位置和性质比改正该错误更难。(T)

15.以对象、类、继承和通信为基础的面向对象设计方法(OOD)也是常见的软件概要设计方法之一。(T)

16.如果通过软件测试没有发现错误，则说明软件是正确的。(F)

17.快速原型模型可以有效地适应用户需求的动态变化。(T)

18.模块化，信息隐藏，抽象和逐步求精的软件设计原则有助于得到高内聚，低耦合度的软件产品(T)

19.集成测试主要由用户来完成。(F)

20.软件危机完全是由硬件问题引起的。(F)

1.一般来说，可以从 一致性 、完整性__、现实性____和有效性__四个方面验证软件需求的正确性。

2.IPO 图是 _输入、处理和输出图_ 的简称，它是美国 IBM 公司发展完善起来的一种图形工具。

3.系统流程图是描述__物理模型__的传统工具，用图形符号表达了系统中各种元素之间的__信息流动__情况。

4.数据流图中的箭头表示数据流，椭圆或圆形表示数据处理，矩形表示数据的源点/终点。

5.集成测试的方法主要有 渐增式和非渐增式 。

6.软件生命周期一般可分为**问题定义、可行性研究、需求分析、设计编码、测试、运行与维护**阶段。

7.可行性研究主要集中在以下四个方面**经济可行性**、技术可行性__、__法律可行性__和抉择。

8.在结构化设计中，HIPO 图应用在总体设计阶段，由 IPO 图和_层次图_两部分组成。

9.实施精化设计的目标是基于模块的“高内聚低耦合”的原则，提高模块的_独立性。

10.按软件的功能进行划分，软件可以划分为 系统软件 ， 支撑软件 ， 应用软件

11.单元测试一般以**白盒**测试为主，__黑盒__测试为辅。

12.面向对象的数据存储管理模式分为__文件__、关系数据库和面向对象数据库三种。

13.通用体系结构模式有：分层模式、管道与过滤器模式和黑板模式。

14.成本效益分析的目的是从(经济)角度评价开发一个项目是否可行。

15.自顶向下结合的渐增式测试法，在组合模块时有两种组合策略：深度优先策略和（ 宽度优先策略 ）。

16.汇编语言是面向(机器) 的，可以完成高级语言无法完成的特殊功能，如与外部设备之间的一些接口工作。

17.详细设计的任务是确定每个模块的内部特性，即模块的算法、(使用的数据)。

18.所有软件维护申请报告要按规定方式提出，该报告也称(软件问题)报告。

19.有两类维护技术：在开发阶段使用来减少错误、提高软件可维护性的面向维护的技术；在维护阶段用来提高维护的效率和质量的(维护支援)技术。

20.用户界面的(可使用性) 是用户界面设计最重要的也是最基本的目标

21.(数据流图) 和(数据字典)共同构成系统的逻辑模型

22.常见的软件概要设计方法有三大类：以数据流图为基础构造模块结构的 结构化设计方法 ， 以数据结构为基础构造模块的 Jackson 方法 ， 以对象、类、继承和通信为基础的 面向对象设计方法。

23.实用的白盒测试覆盖技术有 4 种：语句覆盖、条件覆盖、分支 (或判定) 覆盖和组合覆盖。

24.传统软件维护一般分为 4 大类，分别是纠错性维护、适应性维护、完善性维护和预防性维护。

25.传统软件维护一般分为 4 大类，分别是纠错性维护、适应性维护、完善性维护和预防性维护。

26 用于表示分析模型的 UML 图形机制主要是类图、活动图、交互图与状态图。

26.内聚性有 6 种类型 偶然内聚、逻辑内聚、时间内聚、通信内聚、顺序内聚、功能内聚。

27.一般而言，在面向对象建模过程中，人们最常用的是建立系统的用例图、类图和顺序图。

28.常用的软件开发方法有结构化方法、面向对象方法和形式化开发方法

29.内聚性 是模块独立性的衡量标准之一，它是指模块的功能强度的度量，即一个模块内部各个元素彼此结合的紧密程度的度量。

30.在结构化分析中，用于描述加工逻辑的主要工具有三种，即：结构化语言、判定表、判定树。

1. 软件的开发与运行经常受到硬件的限制和制约。(√)

2. 模块内的高内聚往往意味着模块间的松耦合。(√)

3. Jackson 图只能表达程序结构，不能表达数据结构。(X)

上述数据流图表示数据 A 和 B 同时输入变换成 C。(X)

5. 软件的质量好坏主要由验收人员负责，其他开发人员不必关心。(X)

6. 判定覆盖不一定包含条件覆盖，条件覆盖也不一定包含判定覆盖。(√)

7.应该尽量使用机器语言编写代码，提高程序运行效率，而减少高级语言的使用。(X)

8. UML 只能应用于软件系统模型的建立。(X)

9. 容错就是每个程序采用两种不同的算法编写。(X)

10. 软件测试的目的是为了无一遗漏的找出所有的错误。(X)

判断题二

向滞后的项目中增加人手会使得项目更加滞后。(T)

过程描述语言可以用于描述软件的系统结构。（F）

按照瀑布模型开发软件的一条指导思想是清楚地区分逻辑设计与物理设计，以便尽早开始程序的物理实现。（ F ）

只有质量差的软件产品才需要维护。（ F ）

层次图用来描述软件的层次结构。层次图中的一个矩形框代表一个模块，方框间的连线表示模块的组成关系。在层次图中除最顶层的方框之外，给每个方框都加编号，即成为带有编号的层次图。（ F ）

建立动态模型的第一步，是编写典型交互行为的脚本。（ T ）

软件错误可能出现在开发过程的早期，越早修改越好。（T）

软件模块的耦合是越低越好。（ T ）

一个好的测试用例在于能发现至今未发现的错误。（ T ）

面向对象设计的结果只能采用面向对象语言来实现（ F ）

一个成功的项目唯一提交的就是运行程序。（ F ）

用例之间的关系有 3 种。（ T ）

面向对象设计准则也要遵循弱耦合的原则，但是继承耦合则应该提高，紧密地继承耦合与高度的一般-特殊内聚是一致的。（ T ）

软件工作的考虑范围主要是程序设计和实现。（ F ）

环形复杂度定量度量程序的逻辑复杂度，可以用这个公式来计算环形复杂度：
流图 G 的环形复杂度 $V(G) = N - E + 2$ 。（改成 $E - N + 2$ ）（ F ）

等价类划分方法将所有可能的输入数据划分成若干部分，然后从每一部分中选取少数有代表性的数据作为测试用例。（ T ）

在进行总体(改成概要)设计时应加强模块间的联系。（ F ）

系统结构图是精确表达程序结构的图形表示法。因此，有时也可以将系统结构图当作系统流程图使用。(T)

建立动态模型的第一步，是编写典型交互行为的脚本。(T)

在程序调试时，找出错误的位置和性质比改正该错误更难。(F)

如果通过软件测试没有发现错误，则说明软件是正确的。(F)

快速原型模型可以有效地适应用户需求的动态变化。(T)

模块化，信息隐藏，抽象和逐步求精的软件设计原则有助于得到高内聚，低耦合度的软件产品。(T)

集成测试主要由用户来完成。(F)

面向对象 = 对象 + 类 + 继承 + 消息传递。(T)

面向对象的分析是面向计算机系统建立软件系统的对象模型。(F)

软件的开发与运行经常受到硬件的限制和制约。(T)

模块内的高内聚往往意味着模块间的松耦合。(T)

Jackson 图只能表达程序结构，不能表达数据结构。(F)

软件测试的目的是为了无一遗漏的找出所有的错误。(F)

软件的质量好坏主要由验收人员负责，其他开发人员不必关心。(F)

判定覆盖不一定包含条件覆盖，条件覆盖也不一定包含判定覆盖。(T)

应该尽量使用机器语言编写代码，提高程序运行效率，而减少高级语言的使用。(F)

UML 只能应用于软件系统模型的建立。(F)

容错就是每个程序采用两种不同的算法编写。(F)

判断题：

1. 软件是一种抽象的逻辑产品。(√)
2. 按照软件的服务对象可将软件划分为系统软件和应用软件两类。(×)
3. 开发软件所需高成本和产品的低质量之间有着尖锐的矛盾，这种现象称作软件危机。(√)
4. 软件危机就是大量软件出现严重问题不能继续再使用。(×)
5. 软件工程有 7 个基本要素。(×)
6. 结构化软件开发期将软件开发划分为 3 个阶段，其中关键的是需求的准确性。(√)
7. 瀑布模型存在的最主要的问题是缺乏灵活性。(√)
8. 在软件工程中，软件是指程序+数据+文档。(×)
9. 软件就是指程序。(×)
10. CASE 即计算机辅助软件工程的英文缩写。(√)
11. 瀑布模型也称为是基于文档驱动模型。(√)
12. 快速原型模型也称为是基于文档驱动模型。(×)
13. 软件生存周期是指开发软件的全部时间。(×)
14. 软件工程在软件设计时把软件的效率放在第一位。(√)

15. 可行性分析研究的目的是规划如何开发项目。(×)
16. 可行性分析研究的目的是探讨项目值得开发否。(√)
17. 软件开发的可行性研究，一般涉及经济、技术和操作的可能性，而进行可行性研究的基本依据则是用户提出的软件系统目标。(√)
18. 软件开发的可行性研究，一般涉及经济、技术和操作的可能性，而进行可行性研究的基本依据则是用户提出的软件系统功能。(×)
19. 在可行性研究中最具挑战性的，难度最大是技术可行性分析。(√)
20. 数据存储和数据流都是数据，区别在于数据流是静态数据，而数据存储是动态数据。(×)
21. 数据流图和数据字典共同组成目标系统的逻辑（功能）模型。(√)
22. SA（结构化分析）方法中目标软件的逻辑模型是由数据流图和数据字典共同组成。(√)
23. 数据流图中可以有控制流。(×)
24. 数据字典是对数据定义信息的集合，它是对 DFD 中所有数据对象及加工的详细说明。(√)
25. DFD 中的每个加工至少有一对输入流和输出流。(√)
26. 结构化分析方法是一种面向数据流的自顶向下逐步求精进行需求分析的方法。(√)

27. 结构化分析方法是一种面向数据结构的需求分析的方法。(×)、
28. 需求分析最终结果是产生 SRS (软件需求规格说明书)。(√)
29. 需求分析最终结果是产生项目开发计划。(×)
30. 软件设计基本原理中不包括自底向上的设计思路。(√)
31. 软件设计基本原理中包括自底向上的设计思路。(×)
32. 模块的独立性高意味着模块的内聚弱, 耦合高。(×)
33. 模块是数据说明、可执行语句等程序对象的集合, 它是单独命名的而且可以通过名字来访问。(√)
34. 面向数据流的设计方法简单地说就是把 DFD 中的信息流映射成为软件结构。(√)
35. 面向数据流的设计方法就是把 DFD 中的控制结构映射成为软件结构。(×)
36. 程序流程图(PDF)中的箭头代表的是程序的控制流程。(√)
37. 模块本身的内聚是模块独立性的重要性度量因素之一。在 7 类内聚中, 具有最强内聚的一类是过程性内聚。(×)
38. 面向数据流的软件设计方法, 一般是把数据流图中得数据流化分为变换流和事务流两种类型, 再将数据流图映射为软件结构。(×)
39. 模块的控制域和作用域原则是指作用域应该在控制域之内。(√)

40. 模块化中的信息隐藏原则是指使模块内部的私有信息对软件开发人员隐藏。

(×)

41. 内聚和耦合都是用于度量模块复杂性的两个定性指标。(×)

42. 判定所在的模块称为判定的作用范围。(×)

43. 判定所在的模块及该判定影响的所有模块称为判定的作用范围。(√)

44. 软件详细设计主要采用的方法是流程图设计法。(√)

45. 程序的三种基本控制结构是顺序、选择和重复。(√)

46. 程序的三种基本控制结构的共同特点是只有一个入口和一个出口。(√)

47. 结构化程序设计主要强调的是程序的清晰易读性 (√)

48. 软件详细设计的主要任务是确定每个模块的功能。(×)

49. 检查软件产品是否符合需求定义的测试过程称为集成测试。(×)

50. 测试是为了证明程序中没有错误而执行程序的过程。(×)

56. 软件调试的目的是改正软件中存在错误。(√)

57. 测试用例就是一组测试用的数据。(×)

58. 测试用例就是一组测试用的数据及其预期结果。(√)

59. 使用白盒测试方法时，确定测试数据应根据程序的输入输出关系。(×)

60. 测试的关键问题是如何选择测试用例。(√)

61. 测试的关键问题是用测试用例来驱动测试被测对象。(×)

62. 在白盒法技术测试用例的设计中，语句覆盖是必须要满足的覆盖标准。
(×)

63. 为了适应软硬件环境变化而修改软件的过程是预防性维护。(×)

64. 软件在给定的时间间隔内和给定的条件下，按照规格说明书的规定成功地运行的概率称为软件的可靠性。(√)

65. UML 是统一建模语言的英文简写。(√)

66. 在面向对象技术中，对象之间可以直接调用对方的任何方法。(×)

67. 在面向对象技术中，对象之间利用数据进行通信。(×)

68. 软件项目管理中，需要对软件进行配置，各阶段文档的管理属于资源管理。
(×)

69. 软件能力成熟度模型 CMM 分为 5 个等级。(√)