

软件工程期末考试复习题及答案

一、填空题

1. 软件是计算机系统中与硬件相互依存的另一部分，它是包括（程序）、（数据）及其（相关文档）完整集合。其中，（程序）是按事先设计的功能和性能要求执行的指令序列，（数据）是使程序能够正确操纵信息的数据结构，（相关文档）是与程序开发、维护和使用有关的图文资料。
2. 软件危机是指计算机软件（开发）和（维护）过程中所遇到的一系列严重问题。
3. 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、技术和方法来开发和维护软件，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。
4. 常见的软件开发模型有（瀑布模型）、（快速原型模型）、（增量模型）和（螺旋模型）等；
5. 软件工程学方法的三要素是：（方法）、（工具）和（过程）。
6. 瀑布模型是一个（文档）驱动模型。
7. 可行性研究一般可以从（经济）可行性、（技术）可行性、法律可行性和法律可行性等方面来研究。
8. 数据流图从数据传输和（数据加工）角度，以（图形）的方式描述数据流从输入到输出的传输变换过程。
9. 数据流图的四个基本符号分别表示（数据的源点或终点）、（数据的加工）、（数据存储）和数据流；
10. 实体-联系图中，数据对象之间的关系有（一对一）、（一对多）和（多对多关系）。
11. 需求分析阶段应该建立三种模，即：（数据模型）、（功能模型）和（行为模型）。
12. 一般从（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。
13. HIPO 图是由一组（层次图）图加上一张（IPO）图组成。
14. 模块独立性的两个度量标准为：（耦合）和（内聚）。
15. 一般说来，模块之间的耦合程度（低），则单个模块的内聚程度（高）。
16. 经验表明，设计好的软件结构，通常顶层模块的扇出（比较高），中层模块扇出（较少），底层模块（高）扇入
17. 三种基本的程序控制结果是（顺序结果）、（选择（分支）结构）和（循环结构）。
18. 把程序从一个硬件或软件环境中转移到另一种配置环境称为软件的（可移植性）。
19. 如果一个模块完成一组任务，这些任务彼此之间的关系是很松散的，这种内聚称为（偶然（或巧合）类聚）。
20. 白盒法的具体设计程序测试方法有语句覆盖、（判定覆盖）、（条件覆盖）和路径覆盖。
21. 组成 UML 的三种基本建筑块是（事物）、关系和（图）。
22. 面向对象的软件工程方法包括面向对象的分析、（面向对象的设计）、（面向对象的编程）、（面向对象的测试）和面向对象的软件维护等主要内容。
23. 保证软件质量的措施主要有（审查）、复查和管理复审、（测试）。
24. 提高软件质量和可靠性的技术大致分为两类，即（避开错误技术）和（容错技术）。
25. 常用的估算软件规模的方法有（代码行技术）和（功能点技术）。
26. 功能点技术的信息域包含（输入项数）（输出项数）、（查询数）、（主文件数）、（外部接口数）等 5 个方面的信息。
27. 软件的工作量计算模型是（软件规模）的函数，软件工作量的单位通常是（人月），常用的计算软件工作量的计算模型有（静态单变量模型）、（动态多变量模型）、（COCOMO2 模型）等。
28. 软件过程是软件（生存期）中的一系列相关软件工程（活动）的集合。每个软件过程又是由一组（工作任务）、项目（里程碑）、软件工程产品和交付物以及软件质量保证（SQA）等组成。
29. 软件的度量包括直接度量和间接度量、软件产品的直接度量包括（产生的代码行数）、（执行速度）、（存储量大小）、在某种时间周期中所报告的差错数。软件产品的间接度量则包括（功能性）、（复杂性）、（效率）、（可靠性）、（可维护性）和许多其他的质量特性。

30. 软件配置管理是在软件的整个生命周期内管理变化的一组活动。这组活动包括：①标志变化，②控制变化，③实现变化，④报告变化。
31. 软件的质量可以定义为（用户对软件的满意程度），具体地说，软件质量是软件与（功能和性能需求），（开发标准）以及所有软件产品都应该具有的（隐含特性）相一致的程度
32. 软件维护可以分为（改正性维护）、（适应性维护）、（完善性维护）、（预防性维护）等四类。
33. 把程序从一种计算环境（硬件配置和操作系统）转移到另一种计算环境的难易程度叫软件的（移植性）。
34. 对象是封装了数据结构及可以施加在这些数据结构上的（操作）的封装体，这个封装体可以唯一地标识它的名字，而且向外界提供一组（服务）。

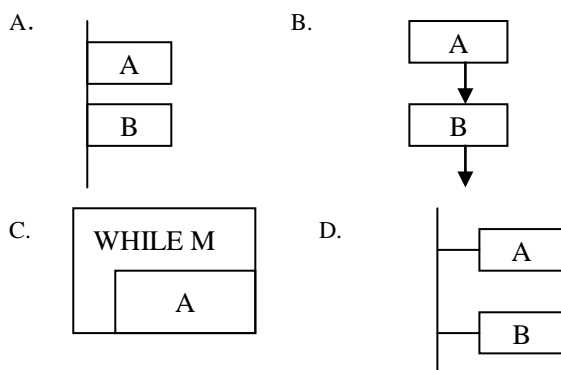
二、单项选择题

1. 软件设计中划分程序模块通常遵循的原则是使各模块间的耦合尽可能（C）
 - a) 强
 - b) 适中
 - c) 弱
2. 若一个模块直接引用另一个模块中的数据，这种类型的耦合称为（E）。
 - a) 公共耦合
 - b) 数据耦合
 - c) 逻辑耦合
 - d) 外部耦合
 - e) 内容耦合
 - f) 控制耦合
3. 瀑布模型是（ ）驱动模型
 - a) 功能
 - b) 文档
 - c) 数据
 - d) 时间
4. 若一个模块把开关量作为参数传送给另一个模块，这种类型的耦合称为（F）。
 - a) 公共耦合
 - b) 数据耦合
 - c) 逻辑耦合
 - d) 外部耦合
 - e) 内容耦合
 - f) 控制耦合
5. 若一个模块通过公共数据结构把数据传送给另一个模块，这种类型的耦合称为（A）。
 - a) 公共耦合
 - b) 数据耦合
 - c) 逻辑耦合
 - d) 外部耦合
 - e) 内容耦合
 - f) 控制耦合
6. 以下各种类型的耦合中，（E）的耦合性最强。
 - a) 公共耦合

- b) 数据耦合
 - c) 逻辑耦合
 - d) 外部耦合
 - e) 内容耦合
 - f) 控制耦合
7. 为了高质量地开发软件项目，在软件结构设计时必遵循（C）的原则
- a) 质量控制
 - b) 程序优化
 - c) 信息隐蔽
 - d) 数据共享
 - e) 模块通信
8. 在建立软件系统的模块结构。应根据（C）评价系统模块划分的质量。
- a) 数据独立性
 - b) 程序独立性
 - c) 模块独立性
 - d) 可修改性
 - e) 可理解性
9. 结构化程序设计方法要求每个（A）结构应是单入口和单出口。
- a) 程序块
 - b) 公共块
 - c) 数据块
 - d) 记录块
 - e) 通信块
10. 开发软件时，对于提高软件开发人员工作效率至关重要是（A）。
- a) 程序开发环境
 - b) 操作系统的资源管理功能
 - c) 开发程序人员数量
 - d) 计算机的并行处理能力
11. 软件工程中描述软件生存周期的瀑布模型一般包括计划、（D）、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为（概要设计）和（详细设计）两个步骤。
- a) 问题定义
 - b) 需求调查
 - c) 可行性分析
 - d) 需求分析
12. 各种需求分析方法都有它们共同适用的（C）。
- a) 说明方法
 - b) 描述方式
 - c) 准则
 - d) 基本原则
13. 软件需求分析阶段的工作，可以分为以下几个方面；对问题定义、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
14. 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) 结构化英语
 - d) 判定表与判定树
15. 在结构化分析方法中，用状态迁移图表达系统或对象的行为。在状态迁移图中，由一个状态和一个事件所决定的下一个状态可能会有（C）个。
- a) 1
 - b) 2
 - c) 多个
 - d) 不确定
16. 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
17. 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
18. 1960 年底 Dijkstra 提倡的（D）是一种有效的提高程序设计效率的方法。
- a) 标准化程序设计
 - b) 模块化程序设计
 - c) 多道程序设计
 - d) 结构化程序设计
19. 汇编程序是指（C）。
- a) 用汇编语言写的程序 B、
 - b) 符号程序
 - c) 汇编语言的处理程序
20. 为了实现递归子程序的正确调用，人们必须用（A）来保存（返回地址）及有关信息。
- a) 堆栈
 - b) 线性表
 - c) 队列
 - d) 入口点
21. UNIX 操作系统是（）研制的，它是用程序语言（C）书写实现的。
- a) Bell 实验室
 - b) DEC 公司
 - c) IBM 公司
 - d) Microsoft 公司
22. 允许用户建立、修改、存储正文的计算机程序是（B）。
- a) Bootstrap
 - b) Editor
 - c) Loader
 - d) TextFormater
23. FORTRAN 语言的源程序是（A）结构。

- a) 块状
 - b) 分程序嵌套
 - c) 既是块状，又是嵌套
 - d) 既不是块状，又不是嵌套的
24. 国际上最广泛使用的商用及行政管理语言是 (A)。
- a) COBOL
 - b) BASIC
 - c) FORTRAN
 - d) PL / 1
25. 国际上最流行的数值计算的程序设计语言是 (C)。
- a) BASIC
 - b) Algol
 - c) FORTRAN
 - d) C
26. 美国国防部主持开发了高级程序设计语言 Ada，在它研制开始时，经反复比较，确定以高级语言 (C) 作为 Ada 研究的出发点。
- a) LISP
 - b) ALGOL
 - c) ALGOL 68
 - d) PL / 1
27. 在人工智能领域，目前最广泛使用的高级语言是 (D)。
- a) Ada
 - b) FORTRA
 - c) COBOL
 - d) LISP
28. 下列 (A) 属于系统软件。
- a) WINDOWS 2000
 - b) Word
 - c) Flash
 - d) 3D MAX
29. 下列哪个图是 N-S 图的构件 (C)。



30. 某次程序调试没有出现预计的结果，下列 (B) 不可能是导致出错的原因。
- a) 变量没有初始化
 - b) 编写的语句书写格式不规范
 - c) 循环控制出错
 - d) 代码输入有误

31. 下列关于程序效率的描述错误的是 (C)。
- a) 提高程序的执行速度可以提高程序的效率
 - b) 降低程序占用的存储空间可以提高程序的效率
 - c) 源程序的效率与详细设计阶段确定的算法的效率无关
 - d) 好的程序设计可以提高效率

32. 下列伪码片段的程序流程图是 (), 其环路复杂度是 ()

33. 下列伪码片段的环路复杂度是 ()

- a) 2
- b) 3
- c) 4
- d) 5

34. 软件过程能力的提高首先需要对当前的软件过程状况进行科学的 (C)。

- a) 度量
- b) 估算
- c) 评估
- d) 管理

36. 软件工程学的目的应该是最终解决软件生产的(B)问题。

- A. 提高软件的开发效率
- B. 使软件生产工程化
- C. 消除软件的生产危机
- D. 加强软件的质量保证.

37. 目前存在有若干种软件生存周期模型。例如瀑布模型、增量模型、螺旋模型等。其中规定了由前至后、相互衔接的固定次序的模型是 (A)。

- A. 瀑布模型
- B. 增量模型
- c. 喷泉模型
- D. 螺旋模型

38. 瀑布模型的问题是(B)。

- A. 用户容易参与开发
- B. 缺乏灵活性
- C. 用户与开发者易沟通
- D. 适用可变需求

39. 系统定义明确之后, 应对系统的可行性进行研究。可行性研究包括(B)

- A. 软件环境可行性、技术可行性、经济可行性、社会可行性
- B. 经济可行性、技术可行性、操作可行性
- C. 经济可行性、社会可行性、系统可行性
- D. 经济可行性、实用性、社会可行性

40. 数据流图是常用的进行软件需求分析的图形工具, 其基本符号是(C)。

- A. 输入、输出、外部实体和加工
- B. 变换、加工、数据流和存储
- c. 加工、数据流、数据存储和外部实体
- D. 变换、数据存储、力 IT-和数据流

41. 数据流图是用于表示软件模型的一种图示方法, 在下列可采用的绘图方法中, (B)

是常采用的。

①自顶向下

②自底向上

③分层绘制

④逐步求精

A. 全是 B. ①③④

c. ②③④ D. ①④

42、在软件开发过程中常用图作为描述工具，DFD 就是面向(B)分析方法的描述工具。

A 数据结构 B 数据流

C 对象 D 构件(component)

43、结构化分析方法使用的描述工具(B)定义了数据流图中每一个图形元素。

A. 数据流图 B. 数据字典

c. 判定表 D. 判定树

44、软件需求规格说明书的内容不应包括对(B)的描述。

A. 主要功能 B. 算法的详细过程

C. 用户界面及运行环境 D. 软件的性能

45、SA 方法的分析步骤是首先调查了解当前系统的工作流程，然后(B)。

A. 获得当前系统的物理模型，抽象出当前系统的逻辑模型，建立目标系统的逻辑模型

B. 获得当前系统的物理模型，抽象出目标系统的逻辑模型，建立目标系统的物理模型

C. 获得当前系统的逻辑模型，建立当前系统的物理模型，抽象出目标系统的逻辑模型

D. 获得当前系统的逻辑模型，建立当前系统的物理模型，建立目标系统的物理模型

46、在面向数据流的软件设计方法中，一般将信息流分为(A)。

A. 变换流和事务流 B. 变换流和控制流

c. 事务流和控制流 D. 数据流和控制流

47 模块独立性是软件模块化所提出的要求，衡量模块独立性的度量标准是模块的(C)。

A. 抽象和信息隐蔽 B. 局部化和封装化

c. 内聚性和耦合性 D. 激活机制和控制方法

49. Jackson 方法是一种面向(B)的方法。

A. 对象 B. 数据结构

C. 数据流 D. 控制流

50. 为了提高测试的效率，应该(D)。

A. 随机地选取测试数据

B. 取一切可能的输入数据作为测试数据

C. 在完成编码以后制定软件的测试计划

D. 选择发现错误可能性大的数据作为测试数据

51. 使用白盒测试方法时，确定测试数据应根据(A)和指定的覆盖标准。

- A. 程序的内部逻辑 B. 程序的复杂结构
c. 使用说明书 D. 程序的功能

三、多项选择题

- 软件危机可以表现为（A、B、C、D）。
A. 产品不符合用户的需要
B. 软件产品的质量差
C. 软件的可维护性差
D. 软件的价格昂贵
- 优秀的程序员经常使用（A、B、C、D）显示他们的经验。
A、空格 B、空行 C、注释 D、缩进
- 下列（B、D）属于主动式防错程序设计。
A. 栈的深度是否合适
B. 内存检查
C. 数组界限是否正确
D. 时间检查
- 关于数据流图有关描述中正确的是（A、B、C）。
A. 数据流图中不夹带控制流
B. 数据流图的主图必须包括数据流图中的四种基本元素
C. 数据流图不是系统的执行程序，不是程序流程图
D. 加工可以只有输入数据流，没有输出数据流
- 软件维护工作量的数学模型如下为： $M=P+K \cdot e^{(c-d)}$ ，其中P是生产性工作，K为一常数，c为软件的复杂程度，d为维护人员对软件的熟悉程度。从这个模型可以看出：（AD）
A. 不使用软件工程方法指导开发的软件维护起来工作量较大；
B. 不使用软件工程方法指导开发的软件维护起来工作量较小；
C. 软件开发人员参加维护工作，维护的工作量将反而加大；
D. 软件开发人员参加维护工作，维护的工作量将相对较低；
- 用于需求分析的结构化分析方法很多，这些分析方法大都遵循以下准则（ABCD）
A. 必须理解并描述问题的信息域；
B. 必须定义软件的功能域；
C. 必须描述软件的行为；
D. 用层次的方法展示各种模型的细节
- 从下列叙述中选出5条关于好的程序设计风格的正确叙述（①④⑤⑦⑩）
① 使用括号以改善表达式的情晰性。
② 对递归定义的数据结构不要使用递归过程。
③ 尽可能对程序代码进行优化。
④ 不要修补不好前程序，要重新写。
⑤ 不要进行浮点数的相等比较。
⑥ 应尽可能多地输出中间结果。
⑦ 利用数据类型对数据值进行防范。
⑧ 用计数方法而不是用文件结束符或输入序列结束符来判别输入的结束。
⑨ 程序中的注释是可有可无的。

⑩ 使用有意义的标识符。

8. 从下列叙述中选出 5 条关于软件测试的正确叙述。(①②③④⑧⑩)

- ① 为了使得软件容易测试, 应该使用高级的程序设计语言编制程序。
- ② 软件测试是一个程序的执行过程, 目的是为了发现软件中隐藏的错误。
- ③ 如果程序中连锁式连接了 8 个判定 (IF) 结构, 则程序中总的路径数可达 2^8 。
- ④ 白盒测试仅与程序的内部结构有关, 完全可以不考虑程序的功能要求。
- ⑤ 为了快速完成集成测试, 采用一次性集成方式是适宜的。
- ⑥ 对一批模块进行测试, 被发现的错误多的模块中残留的错误将比其他的模块少。
- ⑦ 好的测试用例应能证明软件是正确的。
- ⑧ 边界值分析方法是取输入/输出等价类的边界值作为测试用例。
- ⑨ 等价类划分方法考虑了各等价类之间取值的组合情况下可能的结果。
- ⑩ 判定覆盖法可能查不出在判定中逻辑运算符使用有误时产生的错误。

9. 从下列叙述中选出 5 条有利于软件可维护性的正确叙述。(①③⑤⑦⑧⑩)

- ① 在进行需求分析时需同时考虑可维护性问题。
- ② 完成测试作业后, 为了缩短源程序的长度应删去程序中的注释。
- ③ 尽可能在软件生产过程中保证各阶段文档的正确性。
- ④ 编程时应尽可能使用全局变量。
- ⑤ 选择时间效率和空间效率尽可能高的算法。
- ⑥ 尽可能利用硬件的特点。
- ⑦ 重视程序结构的设计, 使程序具有较好的层次结构。
- ⑧ 使用维护工具或支撑环境。
- ⑨ 在进行概要设计时应加强模块间的联系。
- ⑩ 提高程序的易读性, 尽可能使用高级语言编写程序。

10. 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。(②③④⑦⑧)

- 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。(可能是数据文件)
- 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。

11. 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。(①③⑤⑦⑨)

- 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。

12. 由于软件工程有如下哪些的特点，使得软件的管理比其他工程的管理更为困难：（A、B、D）

- a) 软件产品不可见；
- b) 软件生产不存在标准的过程；
- c) 软件的开发成本往往非常高；
- d) 有些大型软件项目往往是“一次性”的项目；

13. 软件维护可以分为改正性维护和（）

- a) 适应性维护
- b) 文档性维护
- c) 完善性维护
- d) 预防性维护

14. OO 软件系统可维护性好体现在以下几个方面（ABCD）

- a) OO 的软件的稳定性比较好
- b) OO 的软件比较容易修改
- c) OO 的软件比较容易理解
- d) OO 的软件易于测试和调试

15. OO 方法建模需要建立以下哪几种模型（A、B、D）。

- a) 对象模型
- b) 功能模型
- c) 数据模型
- d) 动态模型

四、判断题

- 1. 软件生存周期是从软件开始开发到开发结束的整个时期。（F）
- 2. 系统流程图是一个典型的描述逻辑系统的传统工具。（F）
- 3. 数据流图和数据字典共同构成系统的逻辑模型。（T）
- 4. 扇出是一个模块直接调用的模块数目，一般推荐的扇出为 3 或 4。（T）
- 5. 耦合用于衡量一个模块内部的各个元素彼此结合的紧密程度。（F）
- 6. 判定覆盖不一定包含条件覆盖，但是条件覆盖一定包含判定覆盖。（F）
- 7. 在代码的编写中要尽量避免使用 ELSE GOTO 或者 ELSE RETURN 结构。（T）
- 8. 软件测试的目的就是证明软件没有错。（F）
- 9. 语句覆盖的含义是使被测试程序中的每个语句都至少被执行一次。（T）
- 10. 软件开发的成本主要是人的劳动消耗。（T）
- 11. 从层次图和结构图中能看出模块之间的调用次序（F）
- 12. 结构化程序设计就是要彻底在程序中消除 GOTO 语句（F）
- 13. 自顶向下的集成测试方法需要驱动（存根）程序（F）
- 14. 为了加快软件维护作业的进度，应尽可能增加维护人员的数目（F）
- 15. 软件交付给用户使用后，随即进入软件的维护阶段（T）
- 16. CMM 定义的 5 个等级能引导软件机构不断识别出其软件过程的缺陷，指出应该做哪些改进，并（不）提供做这些改进

的具体措施。(F)

17. 开发软件时投入的人员越多, 开发时间就越短 (F)

18. 程序正确性证明能证明程序的功能和动态特性符合设计要求 (F)

五、简答题

1. 层次图和层次方框图之间区别何在? P92

- ①层次图描绘软件的层次结构, 层次方框图描绘数据结构
- ②层次图中方框代表一个功能模块, 层次方框图方框代表数据的子集或数据元素
- ③层次图中连线表示模块间调用关系, 层次方框图连线表示组成关系。

2. 2、软件设计过程中为什么要采用模块化设计方法? P85

参考答案:

- (1)、采用模块化原理可以使软件结构清晰, 不仅容易设计, 也容易阅读和理解;
- (2)、模块化使软件容易测试和调试, 因而有助于提高软件的可靠性;
- (3)、模块化可以提高软件的可修改性;
- (4)、模块化有助于开发过程的组织和管理;

3. 用某种软件复杂性度量算法来度量不同类型的程序时。得出的度量值是否真正反映了它们的复杂性? 如果对同类型的程序进行度量, 其结果是否就比较有价值?

参考答案:

开发规模相同, 但复杂性不同的软件, 花费的成本和时间会有很大的差异。因此到目前为止, 还没有一个软件复杂性度量的方法能够全面、系统地度量任一软件的复杂性, 某一种度量方法只偏重于某一方面。所以, 用某种软件复杂性来度量不同类型的程序, 所得到的度量值不一定真正反映它们的复杂性。但对同一类里的程序, 按某种视点来度量它们的复杂性, 其结果还是比较有价值的。

4. 数据流图的作用是什么? 它有哪些基本成分?

参考答案:

数据流图可以用来抽象地表示系统或软件。它从信息传递和加工的角度, 以图形的方式刻画数据流从输入到输出的移动变换过程, 同时可以按自顶向下、逐步分解的方法表示内容不断增加的数据流和功能细节。因此, 数据流图既提供了功能建模的机制, 也提供了信息流建模的机制, 从而可以建立起系统或软件的功能模型。

基本成分有: 数据原点/终点 数据处理 数据流 数据存储

5. 常用的估算软件规模的方法有哪些? 试比较这些方法的优缺点。

代码行技术的优点, 容易计算, 缺点是太片面, 因为源程序只是软件配置的很少一部分, 不同语言实现同一个软件所需的代码行数不同;

功能点数的特点: 与编程语言无关, 但是判断信息域复杂性级别和技术因素的影响程度时, 存在着相当大的主观因素。

6. CMM 的全称是什么? 其基本思想是什么? 为什么要对 CMM 进行分级? P321

其基本思想是: 软件开发和维护过程中的问题是由于管理软件过程的方法不当引起的, 即使是应用新的软件技术, 也不会自动提高软件的生产率和质量。能力成熟度模型有助于软件开发机构建立一个有规律的、成熟的软件过程。改进后的软件过程将有利于开发出高质量的软件, 使更多的软件项目免受时间和费用超支之苦。3 分

对于任何软件开发机构的软件过程的改进都是在完成一个又一个小的改进的基础之上不断进行的渐进过程, 而不是一蹴而就

的彻底革命。CMM 将软件过程从无序到有序的进化过程分为 5 个等级，用于测量软件机构的软件过程成熟度和评价其软件过程能力

什么是软件配置管理？

答：软件配置管理，简称 SCM(Software Configuration Management)，是在软件的整个生命周期内管理变化的一组活动。具体地讲，这组活动包括：①标志变化，②控制变化，③确保适当地实现变化，④向需要指导这类信息的人报告变化。软件配置管理技术可以使软件变更所产生的错误达到最小并最有效地提高生产率。

7. 什么是基线？你是如何理解基线的？ P319

基线：已经通过正式复审的规格说明或中间产品，它可以作为进一步开发的基础，并且只有通过正式的变化控制过程才能改变它。

基线是通过了正式复审的软件配置项。一旦软件配置变成基线，那么，对于软件配置的改变，必须使用特定的、正式的过程来评估、实现和验证每个变化。

基线除了包含软件配置之外，还有可能包含软件工具。

8. 什么是面向对象的方法学？它有哪些优点？ P193

面向对象的方法就是模拟人类认识世界解决问题的思维习惯开发软件的方法，这种方法认为：OO=对象+类+继承+对象之间的通讯。1 分

OO 方法有如下四个方面的要点：①客观世界是由各种对象组成的，任何事物都是对象，复杂的对象可以有比较简单的对象以某种方式组合而成。②把所有对象都划分成各种对象类，每个类定义一组数据（静态属性，状态信息，类实例专有数据）和一组方法（施加于对象上的操作，类实例所共有的）。③按照子类与父类的关系，把若干个类组成一个层次结构的系统。子类继承父类的方法和属性，同时可以重载某些方法。④对象之间仅能通过传递消息相互联系 ..2 分

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较易开发大型软件产品，⑤可维护性好。 2 分

六、分析题

1. 试说明下面的两个程序段的功能是什么？可否用另一些等效的程序段来代替它，以提高其可读性。

(1)

A=A+B

B=A-B

A=A-B

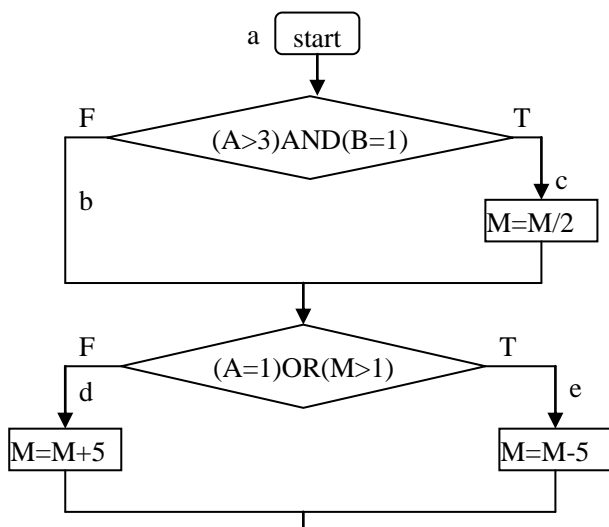
(2)

for(i=0;j<n;i++)

for(j=0;j<n;j++)

V[i][j]=(i/j)*(j*i)

2. 已知被测模块流程图如下，按照“条件覆盖”法，在表格中填入满足测试要求的数据。



假设几种可能的条件是：

T1: A>3 F1: A<=3 T2: B=1 F2: B≠1 T3: A=1 F3: A≠1 T4: M>1 F4: M<=1

填写下表（注意：本题答案不唯一）

数据	覆盖路径	覆盖条件	M 值
A=4,B=1,M=2	cd	T1, T2, F3, F4	6
A=1,B=2,M=4	be	F1, F2, T3, T4	-1

请设计路径覆盖用例，并写出各种测试用例下程序的执行路径。

3. 下面是一段用 Seidel 迭代法求解线性方程组的程序。其中 A[n,n]是方程组的系数矩阵。B[n]是方程组的右端项，X[n]是方程组的解向量。eps 是控制迭代精度的较小实数。imax 是控制迭代的最大次数。flag 是标志，值为 0 时表示迭代不收敛，值为 1 时表示迭代收敛。

```
for(k=0;k<n;k++)
    X[k]=0.0;
X[n-1]=1.0;
for(i=1;i<imax;i++)
{
    flag=1;
    for(j=0;j<n;j++)
    {
        s=B[j];
        for(k=0;k<n;k++)
        {
            if(j==k)
                s += X[k];
            s=s-A[j,k]*X[k];
        }
        if(abs(X[j]-s) > (abs(s) + 1.0)*eps)
            flag=0;
        X[j]=s;
    }
    if(flag==1)
        break;
}
```

画出该 C 伪码的程序流程图，N-S 图，程序流图，并计算程序的 McCabe 环形复杂度。

4. 阅读下列关于软件可靠性方面的叙述，回答问题 1 和问题 2。

软件的可靠度可定义为：在假定输入和硬件不发生错误的前提下，对于给定的环境和给定的输入，在指定的时间内能完成规

定任务的概率。

某软件中心的评测部为了评估已开发实现的应用软件 ASP 的可靠性，决定采用软件可靠性的错误播种模型来进行测试和评估。评测部在评估时，作了下列 3 个假设：

(1) 在测试前，单位长度的故障个数 E_T/I_T 为一常数，此常数基本上落在一个固定的范围内。其中 I_T 为被测程序的长度（即机器指令条数）， E_T 为被测程序中故障总数。

(2) 失效率正比于软件中剩余的（潜伏的）故障数，平均无故障时间 MTTF（Mean Time To Failure）与单位长度的剩余故障个数成反比，即 $MTTF=1/(K \cdot \varepsilon_r)$ 。其中 ε_r 为单位长度剩余故障个数，K 的典型值现取为 200。

(3) 测试中发现的错误都得到了及时改正，在测试过程中没有引入新的错误。评测部对 ASP 软件人为地植入了 10 个错误，即 $N_s=10$ ，在开始测试的一小段时间内，发现了 160 个固有故障，即 $n=160$ ，又发现了植入的故障 2 个，即 $n_s=2$ ，被测程序 ASP 的长度（即机器指令条数）为 100000。

问题 1：用故障播种（植入）的数学模型，估算出被测程序 ASP 的固有故障的个数 N 的值。如果通过测试一段时间后，发现的固有错误个数为 $E_C=795$ 时，请估算此程序的平均无故障时间 MTTF 的值。

问题 2：若要求把此 MTTF 提高 4 倍，应至少再排除多少个固有错误？请简要地列出有关计算式。

$$N = \frac{n}{n_s} N_s = \frac{160}{2} \times 10 = 800$$

$$MTTF = \frac{1}{K * \varepsilon_r} = \frac{1}{K * ((E_T - E_C) / I_T)} = \frac{1}{200 * ((800 - 795) / 100000)} = 100$$

$$\text{由 } \frac{1}{200 * ((800 - x) / 100000)} = 500 \text{ 得:}$$

$$x = 799$$

所以应该在排除：799-795=4 个错误。

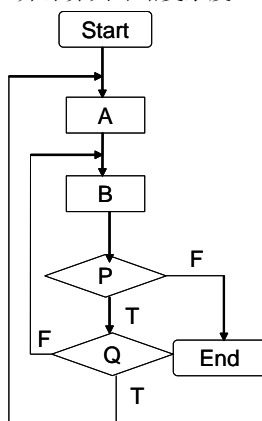
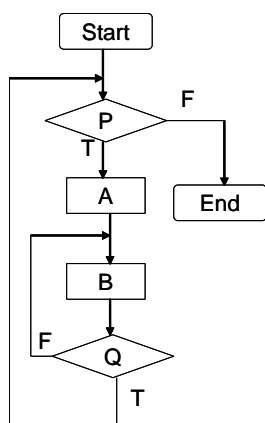
5. 对于包含 10000 条机器指令的程序进行一个月集成测试后，总共改正了 15 个错误，此时 MTTF=10h；经过两个月的测试后，总共改正了 25 个错误（第二个月改正了 10 个错误），MTTF=15h。请问：

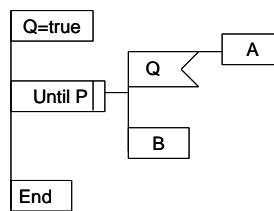
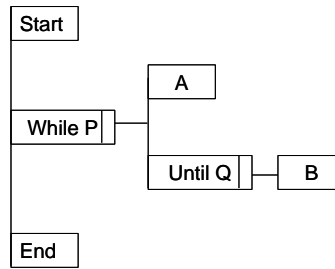
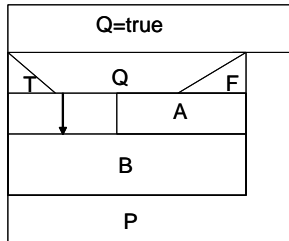
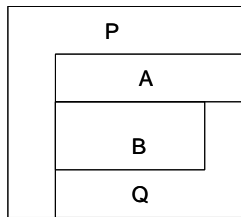
- (1) 在开始测试之前，软件中潜藏着多少个错误？
- (2) 为了达到 MTTF=100h，还应该改正多少个错误？程序中还残留多少个错误？

6. 在测试一个长度为 24000 条指令的程序时，第一个月内由甲、乙两人各自独立地测试这个程序。经过一个月后，甲发现并改正了 20 个错误，使 MTTF 达到 10h。与此同时，乙发现了 24 个错误，其中 6 个错误甲也发现了。以后甲一个人继续独立测试这个程序。请问：

- (1) 刚开始测试这个程序时，程序中总共潜藏有多少个错误？
- (2) 为了使 MTTF 达到 60h，需要再改正多少个错误？此时程序中还残留有多少个错误？

7. 根据下列程序流程图画出相应得 N-S 图和 PAD 图，并计算其环路复杂度。





七、名词解释

1、系统流程图

系统流程图是描绘物理系统的传统工具，用图形符号来表示系统中的各个元素，表达了系统中各个元素之间的信息流动的情况。

2. 数据字典

答：数据字典(Data Dictionary, 简称 DD)就是对数据流图中包含的所有元素的定义的集合。它和数据流图共同构成了系统的逻辑模型。

3. 数据流图(DFD)

答：数据流图，简称 DFD，是 SA 方法中用于表示系统逻辑模型的一种工具，它以图形的方式描绘数据在系统中流动和处理的过程，由于它只反映系统必须完成的逻辑功能，所以它是一种功能模型。

4. 软件工程

答：软件工程是：①把系统的、规范的、可度量的途径应用于软件开发、运行和维护过程，也就是把工程应用于软件；②研究①中提到的途径。

5. 模块

答：在程序中是数据说明、可执行语句等程序对象的集合，或者是单独命名和编址的元素，在软件的体系结构中，模块是可组合、分解和更换的单元。

6. 控制耦合

答：控制耦合指一个模块调用另一个模块时，传递的是控制变量(如开关、标志等)，被调模块通过该控制变量的值有选择地执行块内某一功能。

7. 内聚性

答：内聚性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。

8. PAD 图

答：PAD 图指问题分析图(Problem Analysis Diagram)，是日本日立公司于 1979 年提出的一算法描述工具，它是一种由左往右展开的二维树型结构。

PAD 图的控制流程为自上而下、从左到右地执行。

9. 单元测试

答：单元测试指对源程序中每一个程序单元进行测试，检查各个模块是否正确实现规定的功能，从而发现模块在编码中或算法中的错误。

10. 判定/条件覆盖

答：指设计足够的测试用例，使得判定表达式中的每个条件的所有可能取值至少出现一次，并使每个判定表达式所有可能的结果也至少出现一次。

11. 完善性维护

答：在软件运行时期中，用户往往会对软件提出新的功能要求与性能要求。这种增加软件功能、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。