

<https://blog.csdn.net/ChristensonLee/article/details/125320041>  
[https://blog.csdn.net/weixin\\_43929303/article/details/111371711](https://blog.csdn.net/weixin_43929303/article/details/111371711)  
[https://blog.csdn.net/weixin\\_43398418/article/details/121498466](https://blog.csdn.net/weixin_43398418/article/details/121498466)  
[https://blog.csdn.net/weixin\\_45931586/article/details/122332712](https://blog.csdn.net/weixin_45931586/article/details/122332712)  
<https://max.book118.com/html/2017/0216/91879391.shtml>  
<https://wenku.baidu.com/view/090cb1f218e8b8f67c1cfad6195f312b3169ebf2.html?wkts=1676185352587>

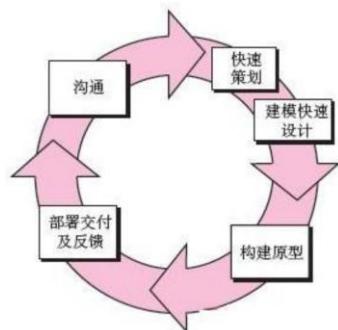
## 判断

### 第一章

1. 软件就是程序，编写软件就是编写程序。 (×)  
2. 软件危机的主要表现是软件需求增加，软件价格上升。 (×)
- 软件危机的主要表现：成本和进度估计不准确、用户满意度不高、质量不可靠、文档不完整、难维护、成本高、开发生产率提高的速度慢
3. 软件工程学科出现的主要原因是软件危机的出现。 (✓)
- 20世纪60年代提出了软件工程的概念
4. 与计算机科学的理论研究不同，软件工程是一门原理性学科。 (×)
- 工程性学科

### 第二章

5. 瀑布模型的最大优点是将软件开发的各个阶段划分得十分清晰。 (×)
- 瀑布模型优点：
- A. 它提供了一个模板，使得分析、设计、编码、测试和支持的方法在该模板下有一个共同的指导。
- B. 虽然有不少缺陷但比在软件开发中随意的状态要好得多。
6. 原型化开发方法包括生成原型和实现原型两个步骤。 (×)
- 原型模型：



7. 软件过程改进也是软件工程的范畴。 (✓)  
8. 在软件开发中采用原型系统策略的主要困难是成本问题。 (×)
- 是因为需求不明确等问题

### 第三章

9. 需求规格说明书在软件开发中具有重要的作用，它也可以作为软件可行性分析的依据。(×)

- 10.需求分析的主要目的是解决软件开发的具体方案。 (×)
- 11.需求规格说明书描述了系统每个功能的实现。 (×)
- 12.非功能需求是从各个角度对系统的约束和限制,反映了应用对软件系统质量和特性的额外要求。 (✓)
- 13.需求评审人员主要由开发人员组成,一般不包括用户。 (×)
- 14.分层的 DFD 图可以用于可行性分析阶段,描述系统的物理结构。 (×)
- DFD 数据流图用于需求分析阶段, 描述数据流程**
- 15.信息建模方法是从数据的角度来建立信息模型的, 最常用的描述信息模型的方法是 E-R 图。 (✓)
- 16.用于需求分析的软件工具,应该能够保证需求的正确性,即验证需求的一致性、完整性、现实性和有效性。 (✓)
- 17.需求分析是开发方的工作,用户的参与度不大。 (×)

#### 第四章

- 18.软件设计说明书是软件概要设计的主要成果。 (×) ? ? ?
- 19.软件设计中设计复审和设计本身一样重要,其主要作用是避免后期付出高昂代价。 (✓)
- 20.HIPO 法既是需求分析方法,又是软件设计方法。 (✓)
- HIPO、IPO、H 图: <https://blog.csdn.net/lvshihua/article/details/8545345>**
- 21.划分模块可以降低软件的复杂度和工作量,所以应该将模块分得越小越好。 (×)
- 22.SD 法是一种面向数据结构的设计方法,强调程序结构与问题结构相对应。 (×)
- SD 法: 结构化的设计方法, 把数据流图转化为系统结构图来设计结构**
- 23.判定表的优点是容易转换为计算机实现,缺点是不能够描述组合条件。 (×)
- 判定表可以描述组合条件, 但当输入项过多时, 规则数以 2 的 n 次方剧增**
- 24.模块独立要求高耦合低内聚。 (×)

#### 第五章

- 25.程序设计语言是指编程时表现出来的特点、习惯、逻辑思维等。 (×)
- 程序设计风格**
- 26.进行程序设计语言的选择时,首先考虑的是应用领域。 (✓)
- 27.好程序的一个重要标准是源程序代码的逻辑简明清晰、易读易懂。 (✓)
- 28.软件测试的目的是尽可能多地发现软件中存在的错误,将它作为纠错的依据。 (✓)
- 29.测试用例由输入数据和预期的输出结果两部分组成。 (✓)
- 30.白盒测试是结构测试,主要以程序的内部逻辑为基础设计测试用例。 (✓)
- 31.软件测试的目的是证明软件是正确的。 (×)
- 32.单元测试通常应该先进行“人工走查”,再以白盒法为主,辅以黑盒法进行动态测试。(✓)
- 33.白盒法是一种静态测试方法,主要用于模块测试。 (×)
- 白盒测试方法有两大类: 静态测试方法和动态测试方法。**
- 静态测试:** 不要求在计算机上实际执行所测试的程序,主要以一些人工的模拟技术对软件进行分析和测试,如**代码检查法、静态结构分析法**等;
- 动态测试:** 是通过输入一组预先按照一定的测试准则构造实际数据来动态运行程序,达到发现程序错误的过程。**白盒测试中的动态分析技术主要有逻辑覆盖法和基本路径测试法**
- 34.在等价分类法中,为了提高测试效率,一个测试用例可以覆盖多个无效等价类。 (×)
- 35.发现错误多的模块,残留在模块中的错误也多。 (✓)

## 第六章

- 36.类是指具有相同或相似性质的对象的抽象，类的具体化就是对象。 (✓)  
37.继承性是父类和子类之间共享数据结构和消息的机制，这是类之间的一种关系。 (✗)

### 子类继承父类的属性和方法

- 38.多态性增强了软件的灵活性和重用性，允许用更为明确、易懂的方式去建立通用软件，多态性和继承性相结合使软件具有更广泛的重用性和可扩充性。 (✗)  
39.类的设计过程包括`确定类，确定关联类，确定属性，识别继承关系。` (✓)  
40.用面向对象方法开发的软件系统，可维护性好。 (✓)  
41.模型是对现实的简化，建模是为了更好地理解所开发的系统。 (✓)  
42.多态性防止了程序相互依赖而带来的变动影响。 (✗)

### 封装防止了程序相互依赖而带来的变动影响

- 43.类封装比对象封装更具体、更细致。 (✗)  
44.面向对象的继承性是`子类自动共享父类数据结构和方法的机制。` (✓)

## 第七章

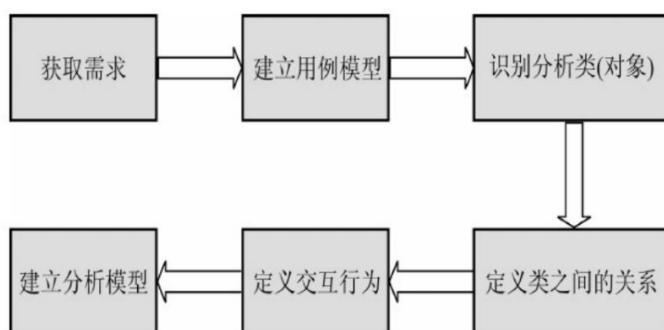
- 45.面向对象的开发方法将过程作为最基本的元素，是分析问题、解决问题的核心。 (✗)  
46.用例需求分析方法采用的是一种结构化的情景分析方法，即是一种基于场景建模的方法。  
(✗)? ? ? ?  
47.面向对象分析方法认为`系统是对象的集合，是以功能和数据为基础的。` (✓)  
48.面向对象分析的核心在于建立一个描述软件系统的模型。 (✗)  
49.面向对象的分析过程主要包括三项内容:`理解、表达和验证。` (✓)  
50.面向对象分析就是`抽取和整理用户需求并建立问题域精确模型的过程。` (✓)  
51.面向对象的方法是以类作为最基本的元素，它是分析问题、解决问题的核心。 (✗)  
52.功能模型是类似编译器之类系统的主要模型。 (✓)

### 软件工程面向对象分析：

<https://wenku.baidu.com/view/33f0feeb57270722192e453610661ed9ac5155a0.html?wkts=1676185945870>

面向对象的分析主要以用例模型为基础

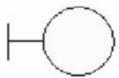
分析流程：



边界类：界面控制类、系统和设备的接口类

控制类：系统的逻辑控制，每个用例一个控制类

实体类：



边界类示意图



控制类示意图



实体类示意图

类之间的关系：依赖（非结构化的短暂的关系）、关联（结构化关系）、聚合、组合、泛化（类似继承关系）、实现关系（针对类与接口的关系）

建模：对象模型、动态模型、功能模型

#### • 7.2.1 建立对象模型

- \_\_\_\_\_ 主题层
- \_\_\_\_\_ 类与对象层
- \_\_\_\_\_ 结构层
- \_\_\_\_\_ 属性层
- \_\_\_\_\_ 服务层

复杂问题（大型系统）的对象模型通常由下述5个层次组成：主题层（也称为范畴层）、类与对象层、结构层、属性层和服务层，如图所示。

上述5个层次对应着在面向对象分析过程中建立对象模型的5项主要活动：划分主题；找出类与对象；识别结构；定义属性；定义服务。

面向对象中大型系统的层次（括号中为对应的主要活动）：

主题层（划分主题）、类与对象层（找出类与对象）、结构层（识别结构）、属性层（定义属性）、服务层（定义服务）

## 第八章

53. 面向对象设计是在分析模型的基础上，运用面向对象技术生成软件实现环境下的设计模型。 (✓)

54. 软件设计一般分为总体设计和详细设计，它们之间的关系是全局和局部。 (✗)

55. 在面向对象的设计中，应遵循的设计准则除了模块化、抽象、低耦合、高内聚以外，还有信息隐藏。 (✓)

56. 一个设计得好的软件系统具有低内聚、高耦合的特征。 (✗)

57. 面向对象分析和设计活动是一个多次反复迭代的过程。 (✓)

58. 关系数据库可以完全支持面向对象的概念，面向对象设计中的类可以直接对应到关系数据库中的表。 (✗)

59. 用户界面设计对于一个系统的成功是至关重要的，一个设计得很差的用户界面可能导致用户拒绝使用该系统。 (✓)

## 第九章

60. 面向对象设计的结果，既可以用面向对象语言，也可以用非面向对象语言实现。 (✓)

61. 一般来说，应该尽量选用面向对象语言来实现面向对象分析、设计的结果。 (✓)

62. 良好的面向对象程序设计风格，既包括传统的程序设计风格准则，也包括为适应面向对象方法所特有的概念（如继承性）而必须遵循的一些新准则。 (✓)

63. 面向对象测试用例的设计由软件的输入--处理--输出视图或单个模块的算法细节驱动。 (×)

64. 面向对象系统的确认测试是面向白盒的，并且可以应用传统的白盒方法完成测试工作。 (×)

- 面向对象单元测试：测试对象是类  
    传统的单元测试：侧重模块接口、局部数据结构、重要执行通路、出错处理通路、边界条件，目的是测出模块中的错误
- 传统的集成测试：测试对象是模块间接口，全局数据结构、模块间数据传输等  
    面向对象的集成测试：

    基于线程的测试：集成响应系统的一个输入或事件所需的一组类，每个线程被集成并分别测试，应用回归测试以保证没有产生副作用

    基于使用的测试：先测试几乎不使用服务器类的类(称为独立类)，再测试使用独立类的类，称为依赖类，直到构造完整整个系统

- 确认测试：在传统软件和面向对象软件间没有明显差别，验证软件的功能和性能及其它特性是否与用户要求一致

    步骤：

    1. 有效性测试（黑盒测试）

    2. 软件配置复查

    3. α 测试（一个用户在受控环境下进行的测试）和 β 测试（多个用户在实际使用环境下的测试）

    4. 验收测试：由用户用实际数据进行测试。考虑软件的功能、性能、可移植性、兼容性、可维护性

- 系统测试：基于实际应用环境对计算机系统的一种多方位的测试，包括：恢复测试、安全测试、压力测试、性能测试、部署测试

## 第十章

65. UML 是一种建模语言，是一种标准的表示，是一种方法。 (×)

UML (Unified Modeling Language) 为面向对象软件设计提供统一的、标准的、可视化的建模语言。适用于描述以用例为驱动，以体系结构为中心的软件设计的全过程。

66. 洪道是一种分组机制，它描述了状态图中对象所执行的活动。 (×)

67. 类图中的角色是用于描述该类在关联中所扮演的角色和职责的。 (✓)

68. 类图用来表示系统中类与类之间的关系，它是对系统动态结构的描述。 (×)

69. 用例模型的基本组成部件是用例、角色和用例之间的联系。 (✓)

70. 用例之间有扩展、使用、组合等几种关系。 (✓)

71. 顺序图描述对象之间的交互关系，重点描述对象间消息传递的时间顺序。 (✓)

72. 活动图显示动作及其结果，着重描述操作实现中所完成的工作，以及用例实例或类中的活动。 (×)

73. UML 语言支持面向对象的主要概念，并与具体的开发过程相关。 (×)

74. 部署图描述系统硬件的物理拓扑结构以及在此结构上执行的软件。 (✓)

## 第十一章

75. 代码行技术是比较简单的定量估算软件规模的方法。 (✓)

76. 功能点技术依据对软件信息域特性和软件复杂性的评估结果，估算软件规模。 (✓)

    基于过程的估算、基于用例的估算

## 基于问题的估算：代码行估算和功能点估算

### 1 LOC(Lines of Code,代码行)估算代码尺寸

把项目划分为若干个功能，分别计算每个功能的代码长度，所有功能代码行之和即项目的代码长度。

LOC估算表包括：

每个功能的代码长度估算值=(乐观值+4\*可能值+悲观值)/6

估算工作量=代码总估算长度/估算生产率

估算总成本=日薪\*估算工作量

估算行成本=估算总成本/估算代码长度

估算生产率由经验获得

### 2 FP(功能点)估算代码尺寸

项目的功能点数是几个测量参数(用户输入数、用户输出数、用户查询数、文件数、外部接口数)的功能点之和。

用户输入数：计算每个用户输入，它们向软件提供面向应用的数据。输入应该与查询区分开来，分别计算。

用户输出数：计算每个用户输出，它们向软件提供面向应用的信息。这里，输出是指报表、屏幕、出错信息，等等。一个报表中的单个数据项不单独计算。

用户查询数：一个查询被定义为一次联机输入，它导致软件以联机输出的方式产生实时的响应。每一个不同的查询都要计算。

文件数：计算每个逻辑的主文件(如数据的一个逻辑组合，它可能是某个大型数据库的一部分或是一个独立的文件)。

外部接口数：计算所有机器可读的接口(如磁带或磁盘上的数据文件)，利用这些接口可以将信息从一个系统传送到另一个系统。

FP估算表包括：

每个测量参数的估算FP计数=估算值\*加权因子

项目估算FP=各参数FP计数之和\*复杂度调整因子

估算生产率由经验获得

估算工作量=项目估算FP/估算生产率

估算总成本=日薪\*估算工作量

单个FP估算成本=估算总成本/估算FP

77.常用的制定进度计划的具主要有Word 和 Excel 两种。 (×)

## 第十二章

78.民主制程序员组的一个重要特点是，小组成员完全平等，享有充分民主，通过协商做出技术决策。 (✓)

79.主程序员组的两个关键特性是专业化和层次性。 (✓)

80.现代程序员组中，技术组长既对技术工作负责，又负责非技术事务。 (×)

## 第十三章

81.风险有两个显著特点，一是不确定性，另一个是损失。 (✓)

82.回避风险指的是：风险倘若发生，就接受后果。 (×)

83.软件质量保证的措施主要有，基于非执行的测试(也称为复审)、基于执行的测试和程序正确性证明。 (✓)

## 第十四章

84.总体上说，软件工程文档可以分为用户文档、开发文档和管理文档三类。 (✓)

85.文档是影响软件可维护性的决定因素。 (✓)

86.适应性维护是在软件使用过程中，用户会对软件提出新的功能和性能要求，为了满足这些新的要求而对软件进行修改，使之在功能和性能上得到完善和增强的活动。 (×)

**适应性维护：**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%。

87.进行软件维护活动时，直接修改程序，无需修改文档。 (×)

88.软件生命周期的最后一个阶段是书写软件文档。 (×)

**软件生命周期：**问题定义、可行性研究、需求分析、设计编码、测试、运行与维护

## 选择题

第一章

1.在下列选项中，( )不是软件的特征。

- A.系统性与复制性 B.可靠性与一致性
- C.抽象性与智能性 D.有形性与可控性

2.软件危机的主要原因是( )。

- A.软件工具落后 B.软件生产能力不足
- C.对软件的认识不够 D.软件本身的特点及开发方法

3.下列说法中正确的是( )。

- A.20世纪50年代提出了软件工程的概念
- B.20世纪60年代提出了软件工程的概念
- C.20世纪70年代出现了客户机/服务器技术
- D.20世纪80年代软件工程学科达到成熟

4.( )是将系统化的、规范的、可定量的方法应用于软件的开发、运行和维护的过程，它包括方法、工具和过程三个要素。

- A.软件生命周期 B.软件测试 C.软件工程 D.软件过程

5.在下列选项中，( )不属于软件工程学科所要研究的基本内容。

- A.软件工程材料 B.软件工程目标 C.软件工程原理 D.软件工程过程

6.软件工程的三要素是( )。

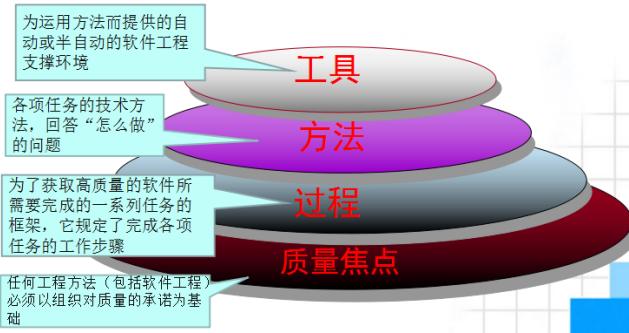
- A.技术、方法和工具 B.方法、对象和类
- C.方法、工具和过程 D.过程、模型和方法

**软件工程三要素：**

(1) 工具：为过程和方法提供自动化或半自动化的支持

(2) 方法：为构建软件提供技术上的解决方法（如何做）

(3) 过程：过程定义了一个框架，构建该框架是有效实施软件工程技术必不可少的



7.用来辅助软件开发、运行、维护、管理、支持等过程中的活动的软件称为软件开发工具，

通常也称为( )工具。

- A. CAD B. CAI C. CAM D. CASE

## 第二章

8. 软件生命周期模型不包括(B)。

- A. 瀑布模型 B. 用例模型 C. 增量模型 D. 螺旋模型

9. 包含风险分析的软件工程模型是(D)。

- A. 喷泉模型 B. 瀑布模型 C. 增量模型 D. 螺旋模型

10. 软件过程是( )。

- A. 特定的开发模型 B. 一种软件求解的计算逻辑

- C. 软件开发活动的集合 D. 软件生命周期模型

11. 在下面的软件开发方法中，哪一个对软件设计和开发人员的开发要求最高( )

- A. 结构化方法 B. 原型化方法 C. 面向对象的方法 D. 控制流方法

12. 与设计测试数据无关的文档是( )。

- A. 需求说明书 B. 设计说明书 C. 源程序 D. 项目开发设计

11. 软件工程中描述生命周期的瀑布模型一般包括计划、需求分析、设计、编码、(D)、维护等几个阶段。

- A. 产品发布 B. 版本更新 C. 可行性分析 D. 测试

12. 软件开发的瀑布模型，一般都将开发过程划分为：分析、设计、编码和测试等阶段，一般认为可能占用人员最多的阶段是( )。

- A. 分析阶段 B. 设计阶段 C. 编码阶段 D. 测试阶段

13. 增量模型本质上是一种( )。

- A. 线性顺序模型 B. 整体开发模型 C. 非整体开发模型 D. 螺旋模型

14. 螺旋模型综合了( )的优点，并增加了风险分析。

- A. 增量模型和喷泉模型 B. 瀑布模型和演化模型

- C. 演化模型和喷泉模型 D. 原型和喷泉模型

螺旋模型特点：

A. 结合了原型的迭代性质和瀑布模型的系统性和可控性特点

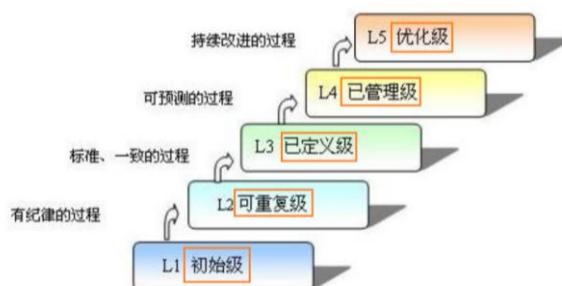
B. 风险驱动，引入非常严格的风险识别、风险分析和风险控制

C. 早期迭代中可能是一个理论模型或原型

15. CMM 模型将软件过程的成熟度分为 5 个等级。在( ) 使用定量分析来不断地改进和管理软件过程。

- A. 管理级 B. 优化级 C. 定义级 D. 可重复级

CMM：



初始级

可重复级：基本的项目管理

已定义级：过程标准化

已管理级：定量管理

优化级：持续的过程改进

### 第三章

16.需求工程的主要目的是( )。

- A.系统开发的具体方案 B.进一步确定用户的需求
- C.解决系统是“做什么的问题” D.解决系统是“如何做”的问题”

17.需求分析的主要方法有( )。

- A.形式化分析方法 B.PAD 图描述
- C.结构化分析 SA 方法 D.程序流程图

18.SA 法的主要描述手段有( )。

- A.系统流程图和模块图 B.DFD 图、数据词典、加工说明
- C.软件结构图、加工说明 D.功能结构图、加工说明

19.画分层 DFD 图的基本原则有( )。

- A.数据守恒原则 B.分解的可靠性原则
- C.子、父图平衡的原则 D.数据流封闭的原则

20.在 E-R 模型中，包含以下基本成分( )。

- A.数据、对象、实体 B.控制、关系、对象
- C.实体、关系、控制 D.实体、属性、关系

21.在下面的叙述中哪一个不是软件需求分析的任务( )。

- A.问题分解 B.可靠性与安全性要求
- C.结构化程序设计 D.确定逻辑模型

22.需求规格说明书的作用不应包括(D)。

- A.软件设计的依据 B.用户与开发人员对软件要做什么的共同理解
- C.软件验收的依据 D.软件可行性研究的依据

23.软件需求规格说明书的内容不应该包括(B)。

- A.对重要功能的描述 B.对算法的详细过程描述
- C.对数据的要求 D.软件的性能

24.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

25.下述任务中，不属于软件工程需求分析阶段的是(D)。

- A.分析软件系统的数据要求 B.确定软件系统的功能需求
- C.确定软件系统的性能要求 D.确定软件系统的运行平台

26.进行需求分析可使用多种工具,但(B)是不适用的。

- A.数据流图 B. PAD 图 C.状态转换图 D. 数据词典

27.在需求分析之前有必要进行(B) 工作。

- A.程序设计 B.可行性分析 C.ER 分析 D.2NF 分析

28.数据流图是进行软件需求分析的常用图形工具，其基本图形符号是(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、加工和数据流

29.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是(A)。

- A.数据流图 B.数据字典 C.结构化语言 D.判定表与判定树

#### 第四章

30.为了提高模块的独立性，模块之间最好是()。

- A.公共环境耦合 B.控制耦合 C.数据耦合 D.特征耦合

耦合：7个

非直接耦合、数据耦合（值传递）、标记耦合（引用传递）、控制耦合（控制变量）、外部耦合（全局简单变量）、公共耦合（全局数据结构）、内容耦合（一个模块直接访问/通过不正常入口访问另一个模块内部、两模块代码重迭（只允许出现在汇编语言中）、一个模块有多个入口）

原则：尽量数据耦合、少用控制耦合、限制公共耦合的范围、不用内容耦合

内聚：7个

功能内聚、信息内聚（同一数据结构的操作）、通信内聚（相同的输入输出）、过程内聚（按一定顺序的过程，如一个流程中的循环部分、判定部分、计算部分是三个过程内聚）、时间内聚、逻辑内聚（由传进来的判定参数决定执行哪部分）、巧合内聚

31.在面向数据流的软件设计方法中，一般将信息流分为()。

- A.数据流和控制流 B.变换流和控制流 C.事务流和控制流 D.变换流和事务流

32.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()。

- A.内聚性和耦合性 B.局部化和封装化 C.抽象和信息隐藏 D.逐步求精和结构图

33.模块的独立性是由内聚性和耦合性来度量的，其中内聚性是(C)。

- A.模块间的联系程度 B.信息隐藏程度 C.模块的功能强度 D.接口的复杂程度

34.当算法中需要用一个模块去计算多种条件的复杂组合，并根据这些条件完成适当的功能时，从供选择的答案中，选出合适的描述工具。()

- A.程序流程图 B.N-S 图 C.PAD 图 D.判定表

35.面向数据流的软件设计方法可将(C)映射成软件结构。

- A.控制结构 B.模块 C.数据流 D.事物流

36.Jackson 方法根据()来导出程序结构。

- A.数据流图 B.数据间的控制结构 C.数据结构 D. IPO 图

#### 第五章

37.程序语言的特性包括()。

- A.习惯特性 B.算法特性 C.工程特性 D.技术特性

38.软件实现是软件产品由概念到实体的一个关键过程，它将( )的结果翻译成用某种程序设计语言编写的并且最终可以运行的程序代码。虽然软件的质量取决于软件设计，但是规范的程序设计风格将会对后期的软件维护带来不可忽视的影响。

- A.软件设计 B.详细设计 C.架构设计 D.总体设计

39.成功的测试是指运行测试用例后( )

- A.发现了程序错误 B.未发现程序错误 C.证明程序正确 D.改正了程序错误

40.白盒测试法是根据程序的()来设计测试用例的方法。

- A.输出数据 B.内部逻辑 C.功能 D.输入数据

13. 白箱测试方法重视( )的度量。

- A. 测试覆盖率 B. 测试数据多少 C. 测试费用 D. 测试周期

41.软件的集成测试工作最好由()承担，以提高集成测试的效果。

- A.该软件的设计人员 B.该软件开发组的负责人

- C.不属于该软件开发组的软件设计人员 D.该软件的编程人员
- 42.黑盒测试是从(用户)观点的测试，白盒测试是从(开发人员)观点的测试。  
A.开发人员、管理人员 B.用户、管理人员  
C.用户、开发人员 D.开发人员、用户
- 43.软件测试可能发现软件中的(错误)，但不能证明软件(没有错误)。  
A.所有错误、没有错误 B.设计错误、没有错误  
C.逻辑错误、没有错误 D.错误、没有错误
- 44.软件测试的目的是(D)。  
A.证明软件的正确性 B.找出软件系统中存在的所有错误  
C.证明软件系统中存在错误 D.尽可能多的发现软件系统中的错误
- 45.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.程序的复杂程度  
C.程序的难易程度 D.程序的功能
- 46.黑盒测试方法根据(B)设计测试用例。  
A.程序的调用规则 B.软件要完成的功能  
C.模块间的逻辑关系 D.程序的数据结构
- 47.在软件测试中，逻辑覆盖标准主要用于(A)。  
A.白盒测试方法 B.黑盒测试方法  
C.灰盒测试方法 D.回归测试方法
- 48.集成测试的主要方法有两个，一个是()一个是()。  
A.白盒测试方法、黑盒测试方法 B.等价类划分方法、边缘值分析方法  
C.渐增式测试方法、非渐增式测试方法 D.因果图方法、错误推测方法.
- 49.验收测试的任务是验证软件的()。  
A.可靠性 B.正确性 C.移植性 D.有效性
- 验收测试：由用户用实际数据进行测试。考虑软件的功能、性能、可移植性、兼容性、可维护性**
- 50.软件测试的目的是尽可能发现软件中的错误，通常(D)是代码编写阶段可进行的测试，它是整个测试工作的基础。  
A.集成测试 B.系统测试 C.验收测试 D. 单元测试

## 第六章

- 51.汽车有一个发动机，汽车和发动机之间的关系是(B)关系。  
A.组装 B.整体部分 C.分类 D.一般具体
- 52.(D)是把对象的属性和操作结合在一起，构成一个独立的对象，其内部信息对外界是隐藏的，外界只能通过有限的接口与对象发生联系。  
A.多态性 B.继承 C.消息 D.封装
53. (B)意味着一个操作在不同的类中可以有不同的实现方式。  
A.多继承 B.多态性 C.消息 D.封装
- 54.每个对象可用它自己的一组属性和它可以执行的一组(A)来表征。  
A.操作 B.功能 C.行为 D.数据
- 55.应用执行对象的操作可以改变该对象的()。  
A.行为 B.功能 C.属性 D.数据
- 56.面向对象的主要特征除了对象唯一性、封装性、继承性外,还有(多态性)。  
A.兼容性 B.完整性 C.可移植性 D.多态性

57. 关联是建立(B)之间关系的一种手段。

- A. 对象 B. 类 C. 功能 D. 属性

58. 面向对象软件技术的许多强有力的功能和突出的优点，都来源于把类组织成一个层次结构的系统，一个类的上层可以有父亲，下层可以有子类，这种层次结构系统的一个重要性质是 (B)，一个类可获得其父亲的全部描述(数据和操作)。

- A. 兼容性 B. 继承性 C. 复用性 D. 多态性

59. 所有的对象可以成为各种对象类， 每个对象类都定义了一组()。

- A. 说明 B. 类型 C. 过程 D. 方法

60. 通过执行对象的操作改变对象的属性，但它必须通过()的传递。

- A. 操作 B. 消息 C. 信息 D. 继承

61. 下列不属于面向对象的要素有(C)。

- A. 继承 B. 抽象 C. 分类性 D. 封装

## 第七章

62. 面向对象分析的首要工作是建立(D)。

- A. 系统的动态模型 B. 系统的功能模型

- C. 基本的 E-R 模型 D. 问题的对象模型

63. 面向对象的分析方法主要是建立三类模型，即(B)。

- A. 系统模型、ER 模型、应用模型 B. 对象模型、动态模型、功能模型

- C. ER 模型、对象模型、功能模型 D. 对象模型、动态模型、应用模型

64. 软件开发过程中，抽取和整理用户需求并建立问题域精确模型的过程叫()。

- A. 生命周期 B. 面向对象设计 C. 面向对象分析 D. 面向对象程序设计

65. 面向对象分析阶段建立的 3 个模型中，核心的模型是() 模型。

- A. 功能 B. 动态 C. 对象 D. 分析

66. 面向对象分析是对系统进行()的一一种方法。

- A. 设计评审 B. 程序设计 C. 需求建模 D. 测试验收

67. 应该在()，确定对象类中应有的服务。

- A. 建立对象模型之时 B. 建立动态模型和功能模型之后

- C. 建立功能模型之后 D. 建立动态模型之后

68. 面向对象的动态模型中，每张状态图表示(A) 的动态行为。

- A. 某一个类 B. 有关联的若干个类 C. 一系列事件 D. 一系列状态

69. 在考察系统的一些涉及时序和改变的状况时，要用动态模型来表示。动态模型着重于系统的控制逻辑，它包括两个图：一个是事件追踪图，另一个是(A)。

- A. 顺序图 B. 状态图 C. 系统结构图 D. 数据流图

70. 动态模型的描述工具是()。

- A. 设计图 B. 结构图 C. 状态图 D. 对象图

71. 对象模型的描述工具是()。

- A. 状态图 B. 数据流图 C. 结构图 D. 对象图

72. 功能模型中所有的( )往往形成一个层次结构，在这个层次结构中一个数据流图的过程可以由下一层数据流图做进一步的说明。

- A. 事件追踪图 B. 物理模型图 C. 状态迁移图 D. 数据流图

## 第八章

73. 只有类的共有界面的成员才能成为使用类的操作，这是软件设计的()原则。

A.过程抽象 B.信息隐藏 C.功能抽象 D.共享性

软件设计的七大原则：

开闭原则 (OCP)

单一职责原则 (SRP)

Liskov 替换原则 (LSP)

依赖倒置原则 (DIP)

接口隔离原则 (ISP)

合成/聚合复用原则 (C/ARP): 尽量使用对象组合，而不是继承来达到复用的目的

迪米特法则 (LOD)/最少知识原则 (LKP)

74. ( ) 是表达系统类及其相互联系的图示，它是面向对象设计的核心，是建立状态图、协作图和其他图的基础。

A.部署图 B.类图 C.组件图 D.配置图

75.下面所列的性质中，(D)不属于面向对象程序设计的特性。

A.继承性 B.重用性 C.封装性 D.可视化

76.下列是面向对象设计方法中有关对象的叙述，其中(B)是正确的。

A.对象在内存中没有它的存储区 B.对象的属性集合是它的特征表示

C.对象的定义与程序中类型概念相当 D.对象之间不能相互通信

77.面向对象程序设计中，基于父类创建的子类具有父类的所有特性(属性和方法),这一特点称为类的(C)。

A.多态性 B.封装性 C.继承性 D.重用性

78.面向对象设计 OOD 模型的主要部件中，通常不包括(A)

A.通信部件 B.人机交互部件 C.任务管理 D.数据管理

79.面向对象设计时，对象信息的隐藏主要是通过()实现的。

A.对象的封装性 B.子类的继承性 C.系统模块化 D.模块的可重用

80.面向对象设计阶段的主要任务是系统设计和(D)。

A.结构化设计 B.数据设计 C.面向对象程序设计 D.对象设计

## 第九章

81.面向对象的实现主要包括(A)。

A.把面向对象设计用某种程序设计语言书写为面向对象程序,测试并调试面向对象的程序

B.面向对象设计

C.选择面向对象语言

D.单元测试

82.面向对象程序设计语言不同于其他语言的最主要特点是()。

A.模块性 B.抽象性 C.继承性 D.内聚性

**基本特征主要有抽象性、继承性、封装性和多态性**

83.面向对象的测试与传统测试方法的主要区别是()。

A.面向对象的测试可在编码前进行，传统测试方法在编码后进行

B.面向对象的测试以需求和设计阶段的测试为主，不需要进行代码测试

C.测试对象不同

D.面向对象的测试不需要设计测试用例，只需要进行会议评审

## 第十章

84.UML 是软件开发中的一个重要工具，它主要应用于()。

A. 基于螺旋模型的结构化方法 B. 基于需求动态定义的原型化方法

C. 基于数据的数据流开发方法 D. 基于对象的面向对象的方法

85.(C)是从用户使用系统的角度描述系统功能的图形表达方法。

A. 类图 B. 活动图 C. 用例图 D. 状态图

86.(A)是表达系统类及其相互联系的图示，它是面向对象设计的核心，是建立状态图、协作图和其他图的基础。

A. 类图 B. 状态图 C. 对象图 D. 部署图

87.(B)描述了一组交互对象间的动态协作关系，它表示完成某项行为的对象和这些对象之间传递消息的时间顺序。

A. 类图 B. 顺序图 C. 状态图 D. 协作图

88.(C)是用于表示构成分布式系统的节点集和节点之间的联系的图示，它可以表示系统中软件和硬件的物理架构。

A. 组件图 B. 类图 C. 部署图 D. 状态图

89.UML 是()的缩写。

A. Unified Module Language B. Universal Module Language

C. Universal Module Locator D. Unified Modeling Language

90.()定义了系统的功能需求，它是从系统的外部看系统功能，并不描述系统内部对功能的具体实现。

A. 用例图 B. 类图 C. 活动图 D. 对象图

91. 状态图包括(D)。

A. 类的状态和状态之间的转换 B. 触发类动作的事件

C. 类执行的动作 D. 所有以上选项

## 第十一章

92. ()的作用是为有效地、定量地进行管理，把握软件工程过程的实际情况和它所产生的产品质量，

A. 估算 B. 度量 C. 风险分析 D. 进度安排

93. LOC 和 FP 是两种不同的估算技术，但两者有许多共同的特征，只是 LOC 和 FP 技术对于分解所需要的()不同。

A. 详细程度 B. 分解要求 C. 使用方法 D. 改进过程

17. 制定软件计划的目的在于尽早对预开发的软件进行合理估价，软件计划的任务是()

A. 组织与管理 B. 分析与估算

C. 设计与测试 D. 规划与调度

18. 软件设计将涉及软件的构造、过程和模块的设计，其中软件过程是指()

A. 模块间的关系 B. 模块的操作细节

C. 软件层次结构 D. 软件开发过程

24. 李先生在公司的项目组中担任项目经理角色，该项目组负责的项目已经处于稳定阶段，在过去的一个星期内，产品中新缺陷产生的数量远远少于缺陷被解决的数量，针对该情况，李先生需要向项目投资方汇报当前的项目状态，以下哪些说法是正确的()

A. 该产品已经可以部署 B. 零缺陷反弹里程碑已经达到

C. 当前的版本可以作为候选版本发布 D. 缺陷收敛里程碑已经达到

25. 以下哪些不属于开发阶段的活动()

A. 内部发布 B. 构造用户体验交付物

C. 测试解决方案 D. 确定解决方案范围

## 第十二章

94.项目团队原来有 6 个成员，现在又增加了 6 个成员，这样沟通渠道增加了多少?()

- A.4.4 倍 B.2 倍 C.6 倍 D.6 条

本来是 6 个人,两两进行沟通,不重复,那就是  $5+4+3+2+1=15$  种沟通方式,那多了六个人就有 12 个人,同上,两两不重复沟通,就有 66 种, $66/15=4.4$

95. Mantei 提出了 3 种通用的项目组组织方式:民主分权式、控制分权式、()。

- A.启发式 B.归纳式 C.总结式 D.控制集权式

## 第十三章

96.下列哪项不是风险管理的过程?(D)

- A.风险规划 B.风险识别 C.风险评估 D.风险收集

97.按照软件配置管理的原始指导思想,受控制的对象应是(C)。

- A.软件过程 B.软件项目 C.软件配置项 D.软件元素

98.下面(D)不是人们常用的评价软件质量的 4 个因素之一。

- A.可理解性 B.可靠性 C.可维护性 D.易用性

## 第十四章

99.在软件维护的内容中,占维护活动工作量比例最高的是()。

- A.纠错性维护 B.适应性维护 C.预防性维护 D.完善性维护

100.使用软件时提出增加新功能就必须进行() 维护。

- A.预防性 B.适应性 C.完善性 D.纠错性

软件维护的内容是:

1、**改正性维护**: 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%;

2、**适应性维护**: 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%;

3、**完善性维护**: 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**。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%。这方面的维护还要注意将相关的**文档资料加入到前面相应的文档中去**。

4、**预防性维护**,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**。占 4%。

101.软件维护的副作用是指()。

- A.运行时误操作 B.隐含的错误 C.因修改软件而造成的错误 D.开发时的错误

102.软件文档是软件工程实施的重要成分,它不仅是软件开发各阶段的重要依据,而且也影响软件的()。

- A.可用性 B.可维护性 C.可扩展性 D.可移植性

103 影响软件可维护性的主要因素不包括()。

- A.可修改性 B.可测试性 C.可用性 D.可理解性

决定软件可维护性的因素有:

A、可理解性

外来读者理解软件的结构、接口、功能和内部过程的难易程度。模块化、详细的设计文档、结构化设计、源代码内部文档和良好的高级程序设计语言等,都对改进软件的可理解性有重要贡献。

## B、可测试性

诊断和测试的难易程度主要取决于软件容易理解的程度。良好的文档对诊断和测试是至关重要的。此外，软件结构、可用的测试工具和调试工具，以及以前设计的测试过程也都是非常重要的。维护人员应该能够得到在开发阶段用过的测试方案，以便进行回归测试。在设计阶段应该尽力把软件设计成容易测试和容易诊断的。

## C、可修改性

软件容易修改的程度和软件设计原理和规则直接相关。耦合，内聚，局部化，控制域的关系等等，都影响软件的可修改性。

## D、可移植性

可移植性表明程序转移到一个新的计算环境的可能性的大小。或者它表明程序可以容易地、有效地在各种各样的计算环境中运行的容易程度。

## E、可重用性

可重用性定义为程序不修改或者稍加改动就在不同环境中可以重新使用的容易程度。

文档：可维护性的决定因素

# 简答题

1. 请简述现代软件工程与传统软件工程显著的区别和改进。

9. 传统的软件工程是基于结构化的软件开发方法，而现代软件工程是以面向对象技术为标志。

- 不仅在编程技术上有很大的改进，而且在分析、设计等整个开发过程中，采用面向对象的思维方式，更加完整、自然地反映客观世界。
- 采用架构技术，开发效率、产品质量得到了极大提高。
- 更注重团队开发和管理，融入更多、更新的管理理念和手段，如 RUP 模型、XP 模型、过程改进、能力成熟度模型、配置管理等。

CSDN @KirinLee\_01

2. 敏捷方法的核心价值观有哪些？它对传统方法的“反叛”体现在哪些方面？

个人和交互 胜过 工具和过程

可以工作的软件 胜过 面面俱到的文档

客户合作 胜过 合同谈判

响应变化 胜过 遵循计划

8. 敏捷思想的核心价值观大体包括以下几方面。

- 注重个人与交互。
- 重点关注可以工作的软件。
- 提高客户参与程度。
- 以积极的心态响应需求变化。

敏捷联盟强调的价值观是这一流派的核心。人是受价值观驱使的，敏捷项目管理因而也是以价值观为推动力的。一个团队可以采用敏捷做法，但如果它不接受敏捷价值观，它将不能得到敏捷方法的潜在好处。

敏捷方法对传统方法的“反叛”体现在以人为中心，不再依靠文档来控制过程，认为传统过程过于僵化、文档繁琐，主张对传统过程瘦身，在上述价值观的支配下，做到够用为准。

3. 假设你要开发一个软件，它的功能是把 73624.9385 这个数开平方，所得到的结果应该精

确到小数点后 4 位。一旦实现并测试完之后，该产品将被抛弃。你打算选用哪种软件生命周期模型？请说明你做出这样选择的理由。

11. 对这个软件的需求很明确，实现开平方功能的算法也很成熟。因此，既无须通过原型模型来分析需求也无须用原型模型来验证设计方案。此外，一旦实现并测试完之后，该产品将被抛弃，因此也无须使用有助于提高软件可维护性的增量模型或螺旋模型来开发该软件。

综上所述，为了开发这个简单的软件，使用大多数人所熟悉的瀑布模型就可<sup>见于 @KirinLee\_01</sup>

#### 4. 需求分析的理论基础有哪些？

8. 需求分析的理论基础：什么是软件需求、软件需求需要量化、需求是一个过程、需求过程中的角色、需求过程是一个迭代的过程、需求来源。  
CSDN @KirinLee\_01

#### 5. 软件测试应该划分几个阶段？各个阶段应重点测试的内容是什么？

确定测试计划，测试设计，设计测试用例，实施测试，进行测试报告 5 个步骤。

- 单元测试是对软件中的基本组成单位进行的测试，如一个模块、一个过程等。它是软件动态测试的最基本的部分，也是最重要的部分之一，其目的是检验软件基本组成单位的正确性。
- 集成测试是软件系统集成过程中所进行的测试，其主要目的是检查软件单位之间的接口是否正确。
- 系统测试是对已经集成好的软件系统进行彻底的测试，以验证软件系统的正确性和性能等满足其规约所指定的要求，检查软件的行为和输出是否正确并非是一项简单的任务，它被称为测试的“先知者问题”。
- 验收测试是向软件的购买者展示该软件系统满足其用户的需求。它的测试数据通常是系统测试的测试数据的子集。
- 回归测试是在软件维护阶段，对软件进行修改之后进行的测试。其目的是检验对软件进行的修改是否正确。

CSDN @KirinLee\_01

#### 6. 请简述软件质量的定义。

3. 所谓软件质量，就是供方提供的软件产品满足用户明确和隐含需求的能力特性的总和。  
CSDN @KirinLee\_01

#### 7. 软件工程中的文档可以分为哪几类？

10. 按照文档产生和使用的范围，软件工程中的文档大致可以分为以下 3 类。

① 开发文档：这类文档在软件开发过程中，作为软件开发人员前一阶段工作成果的体现和后一阶段工作依据的文档。包括：软件需求说明书、数据要求说明书、概要设计说明书、详细设计说明书、可行性研究报告、项目开发计划。

② 管理文档：这类文档是在软件开发过程中，由软件开发人员制定的需提交的一些工作计划或工作报告。管理人员能够通过这些文档了解软件开发项目的安排、进度、资源使用和成果等。包括项目开发计划、测试计划、测试报告、开发进度月报及项目开发总结。

③ 用户文档：这类文档是软件开发人员为用户准备的有关该软件使用、操作、维护的资料。它包括用户手册、操作手册、维护修改建议、软件需求说明书、运行模式建议说明书  
CSDN @KirinLee\_01