

1. 结构化分析方法的分析策略是自顶向下逐步求精。
2. 衡量模块独立性的两个定性标准是耦合性与内聚性。
3. 软件集成测试的方法主要有两种，它们是渐增式与非渐增式测试。
4. 可行性研究的目的是用最小的代价在尽可能短的时间内确定该软件项目是否值得开发。
5. 需求分析阶段，分析人员要确定对问题的综合需求，其中最主要的是功能需求。
6. 软件生存周期中时间最长、花费的精力和费用最多的一个阶段是维护阶段。
7. 计算机辅助设计工程这一术语的英文缩写为CASE。
8. McCall提出的软件质量模型包括11个软件质量特性。
9. 软件结构是以模块为基础而组成的一种控制层次结构。
10. 软件概要设计的主要任务是软件结构的设计。
11. 结构化程序设计方法是使用三种基本控制结构（条件、控制、循环）构造程序。
12. 在建立对象的功能模型时，使用的数据流图中包含有处理、数据流、动作对象和数据存储对象。
13. 增量模型适合于软件需求不明确、设计方案有一定风险的软件项目。
14. 结构化分析方法的分析策略是自顶向下逐步求精。
15. 可行性研究实质上是要在较高层次上以较抽象的方式进行需求分析和设计的过程。
16. 衡量模块独立性的两个定性标准是耦合性和内聚性。

17. Jackson方法是一种面向数据结构的设计方法。
18. 软件集成测试的方法主要有两种，它们是渐增式和非渐增式测试。
19. 维护阶段是软件生存周期中花费精力和费用最多的阶段。
20. 继承性是子类自动共享其父类的数据结构和方法的机制。
21. 在面向对象方法中，人们常用状态图描述类中对象的动态行为。
22. 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。
23. 文档是一种数据媒体和其上所记录的数据。
24. 机器语言、汇编语言：依赖于机器，面向机器。
25. 软件有效性，指软件系统的时间和空间效率。
26. 软件定义过程可通过软件系统的可行性研究和需求分析两个阶段来完成。
27. 软件定义的基本任务是确定软件系统的工程需求，也就是要搞清“做什么”。
28. 为了使用户需求逐步精细化、完全化、一致化，通常采用需求建模技术。
29. 一个软件的深度是指其控制的层数。
30. 以详细设计说明书为输入，将该输入用某种程序设计语言翻译成计算机可以理解并最终可运行的代码的过程叫编码过程。
31. 软件维护是软件生命周期的最后一个阶段。
32. 软件工程由方法、工具和过程三部分组成，称软件工程的三要素。
33. 软件可理解性，指系统具有清晰的结构，能直接反映问题的需求。
34. 在实体-关系图简称E-R图中，数据对象实体用长方形，关系用菱形，属性用椭圆表示。
35. 从工程管理的角度软件设计可分为概要设计和详细设计两大步骤。

36.通过向用户提供视觉和听觉上的反馈，保持用户与界面间的双向通信，是提高界面可交互性的措施之一。

37.测试用例中，不仅要选择合理的输入数据，还要选择不合理的输入数据。

38.软件工程由方法、工具和过程三部分组成，称软件工程的三要素。

39.基于计算机系统的软件要素中软部件由程序、数据和文档组成。

40.流程图也称为程序框图是常用的一种表示法，它有顺序、选择和循环三个基本控制构件。

41.机器语言和汇编语言，也称为低级语言。

42.软件测试是执行程序发现并排除程序中潜伏的错误的过程。

43.动态测试有两种测试方法：黑盒测试和白盒测试。

44.软件可维护性，是指软件产品交付使用后，在实现改正潜伏的错误、改进性能等属性、适应环境变化等方面工作的难易程度。

45.黑盒测试主要是测试软件是否满足功能需求。

46.调试的目的是确定错误的位置和引起错误的原因，并加以改正。因此，又称为排错或纠错。

47.软件工程由方法、工具和过程三部分组成，称软件工程的三要素。

48.软件项目管理的范围主要集中于3个P上，即：People人员、Problem问题和（过程）process项目project。

49.软件工程三要素方法，工具，过程。

50.类图不能用于动态建模的有，用例模型的基本组成部件是用例、角色和用例之间的联系。

51.数据流图的组成元素包括数据的源点或终点，数据的处理，数据流和数据存储。

52.数据字典的组成元素包括数据流，数据流分量，

数据存储和处理。

53. 技术可行性的主要内容：研究现有技术、资源及限制能否支持和实现系统的功能、性能，主要是技术风险问题。

54. 软件测试的基本原则：所有的测试都应追溯到用户需求、应该在测试工作真正开始前的较长时间内就进行测试计划、Pareto原则应用于软件测试、测试应该从“小规模”开始，逐步转向“大规模”穷举测试是不可能的、为了达到最佳效果，应该由独立的第三方来构造测试。

55. 模型元素有关联、聚集、依赖。

56. 维护：所谓软件维护就是在软件已经交付使用之后，为了改正错误或满足新的需要而修改软件的过程。维护的过程：维护组织，维护报告。软件的可理解性，可测试性，可修改性，可移植性和可重用性是决定软件可维护性的基本因素。

57. 软件复杂性度量：McCabe方法，Halstead方法。

58. 软件的开发模型有瀑布模型、快速原型模型、增量模型、螺旋模型、喷泉模型。

59. 将下面的关系按继承关系、聚集关系或普通关联进行分类。

小汽车---红旗轿车（继承）

小汽车---驾驶员（普通关联）

班级---学生（聚集）

60. 将下列各项分为类或类的实例

我的汽车（实例） 交通工具（类）

61. 可行性研究的目的是不是去开发一个软件项目，而是研究这个项目是否值得去开发，其中的问题能否解决。

62. 软件结构的设计是以模块为基础的。

63. 软件设计阶段，是把软件“做什么”的逻辑模型变换为“怎么做”的物理模型。

64. 需求分析的关键是必须准确理解用户的要求，同

时满足用户的要求。最终解决目标系统“做什么”的问题。

65.数据流图一般可分为变换型和事务型两类。

66.HIPO图中H图用于描述软件总的模块层次结构IPO图用于说明模块间的信息传递及模块内部的处理。

67.详细设计也称模块设计，它是软件设计的第二阶段，主要确定每个模块的具体执行过程。

68.处理过程设计中最典型的方法是结构化程序设计（SP）方法，其基本要点是采用的程序设计方法是自顶向下、逐步求精。

69.测试用例由输入数据和预期的输出数据两部分组成。

70.白盒法必须考虑程序的内部结构和处理过程，以检查处理过程的细节为基础，对程序中尽可能多的逻辑路径进行测试。

71.软件测试过程中需要3类信息：软件配置、测试配置和测试工具。

72.在详细设计阶段，需要对处理过程的模块和数据库的物理结构做评审。

73.对象具有状态，对象用数据值来描述它的状态。

74.数据流图描述系统的分隄，即描述系统由哪几部分组成，各部分之间有什么联系等。

75.任何程序都可以由顺序、选择和重复3种基本控制结构构造。这3种基本结构的共同点是单入口、单出口。

76.测试的目的就是发现程序中的错误，而不是调试程序没有错误。

77.对象之间进行的通信的构造叫做消息，类中操作实现的过程叫做方法。

78.在面向对象方法中，信息隐蔽通过对象的封装性来实现。类结构分离了细节与实现，从而支持了信息隐藏。

79.面向数据流的设计又称结构化程序设计（SP），

它是以需求分析阶段产生的数据流图为基础，按一定的步骤映射成软件结构。

80.可行性分析包括技术可行性、经济上的可行性和社会因素可行性分析。

81.对象模型表示了静态、结构化的系统数据性质。

82.动态模型描述了系统的控制结构。

83.对象实现了数据和操作的结合，使它们封装于对象的统一体中。

84.在客观世界中又若干类，这些类之间有一定的结构关系。通常有两种主要的结构关系，即一般-具体结构关系，整体-部分结构关系。

85.具有相同或相似性质的对象的抽象就是类。类的实例就是对象，也可以说类的具体化是对象。

86.类具有属性，它是状态的抽象，用数据结构来描述类的属性。

87.类具有操作，它是行为的抽象，用操作名和方法实现来描述。

88.现实世界中，各对象之间相互触发，一个触发行为就是一个事件。对事件的响应取决于接受该触发的对象的状态，响应包括状态的改变或形成一个新的触发。

89.链表示对象间的物理与概念联结，关联表示类之间的一种关系，就是一些可能的链的集合。链是关联的实例，关联是链的抽象。两个类之间的关联称为二元关联，三个类之间的关联称为三元关联。

90.聚集关系是一种整体-部分关系。在这种关系中，有整体类和部分类之分。聚集最重要的性质是传递性，也具有逆对性。

91.在结构化分析中，数据字典用于详细地定义数据流图中的成分。

92.耦合性和内聚性是衡量模块独立性的两个定性的标准。

93.结构化程序设计是一种程序设计技术，它采用自

顶向下、逐步求精的设计方法和顺序、选择、循环三种基本控制结构构造常态程序。

94. 结构化分析方法中，数据流图的四种基本成分为：处理、源点或终点、数据存储和数据流。

95. 从模块内外的联系来看，结构化设计的要求是高内聚低耦合。

96. 在需求分析中，可从有关问题的简述中提取组成数据流图的基本成分。通常问题简述中的动词将成为数据流图中的处理成分。

97. 如果两模块间的通信信息是若干个参数，而每个参数是单一的数据项，则这种耦合性为数据耦合。

98. 模块内部的算法设计在结构化方法的详细设计阶段进行。

99. 定性度量模块的两个标准是内聚性和耦合性。

100. 在模块结构图中，直接调用某一模块的其他模块数称为该模块数的扇入数。

101. 在软件开发过程中要产生大量的信息，要进行大量的修改，软件配置管理能协调软件开发，并使混乱减到最低程度。

102. 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，则称此软件具有容错功能的软件。

103. McCall提出的软件质量模型包括11个软件质量特性。

104. 软件可维护性度量的七个质量特性是：可理解性、可测试性、可修改性、可靠性、可移植性、可使用性和效率。

105. 为了便于对照检查，测试用例应由输入数据和预期的输出结果两部分组成。

106. 程序设计语言的心理特性主要表现在歧义性、简洁性、传统性、局部性和顺序性。

107. 软件结构是以模块为基础而组成的一种控制层次结构。

- 108.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。
- 109.结构化语言是介于自然语言和形式语言之间的一种半形式语言。
- 110.软件是计算机程序及其说明程序的各种文档。文档是有关计算机程序功能、设计、编制、使用的文字或图形资料。
- 111.软件生存周期是指一个软件从提出开发要求开始直到该软件报废不止的整个时期。通常，软件生存周期包括问题定义、可行性研究、需求分析、总体设计、详细设计、编码测试、软件维护活动，可以将这些活动以适当方式分配到不同阶段去完成。
- 112.可行性研究的目的是用最小的代价在尽可能短的时间内确定该软件项目是否能够开发，是否值得去开发。
- 113.模块的耦合性分为四种类型：数据耦合、控制耦合、公共耦合、内容耦合。
- 114.内聚性有七种类型，最强的是功能内聚，最弱的是偶然内聚。
- 115.结构图的深度指结构图控制的层数，结构图宽度是指回一层个数。
- 116.对于变换型的数据流图，按照软件设计思想，要将一个大型复杂的软件进行分解，要确定数据流图的变换中心、逻辑输入、逻辑输出。
- 117.软件编码阶段要善于积累编程的经验，使得编出的程序清晰易懂，易于测试与维护，从而提高软件质量。
- 118.Jackson方法是一种面向数据结构的开发方法。
- 119.需求分析的基本任务是要准确回答系统必须“做什么”的问题。
- 120.数据流图有四种基本成分：源、终点、处理、存储、数据流。
- 121.通过功能分解可以完成数据流图的细化。



122. 模块化是指解决一个复杂问题时自顶向下逐层把软件系统划分成若干模块的过程。每个模块完成一个特定的子功能，所有的模块按照某种方法组装起来。成为一个整体，完成整个系统所要求的功能。

123.在面向对象方法中，对象实现了数据和操作的结合，使数据和操作封装于对象的统一体中。

124.如果知道软件的规格说明书描述的程序功能，那么就可以采用确认测试进行测试。

125.软件可维护性度量的七个质量特性是可理解性、可测试性、可修改性、可靠性、可移植性、可使用性和效率。

126.当A模块调用B模块时，若两个模块之间传递的是数值型参数，则这两个模块的耦合方式是数据耦合。

127.向对象发出请求服务为接口，类对象上的各种操作称为方法。

128.提高软件质量和可靠性的技术主要有两类：避开错误技术和容错技术。

129.大型软件测试包括单元测试、集成测试、确认测试和系统测试四个步骤。

130.软件结构图的宽度是指一层中最大的模块个数。

131.软件生存周期是指一个软件从提出开发要求开始直到该软件报废为止的整个时期。

132.McCall提出的软件质量模型包括11个软件质量特性。

133.对大信息系统的需求，DFD应与数据库技术中的ER图结合起来。

134.瀑布模型是以文档为驱动、适合于需求很明确的软件项目的模型。

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

136.数据项是指不可再分的数据单元。

137.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底向上或相反顺序集成的过程。

138.软件维护的内容包括纠正性维护，适应性维护、完善性维护和预防性维护。

139.CMM将软件过程的成熟度分为5个等级是初始级、可重复级、已定义级、已管理级和优化级。

140.软件结构是以模块为基础而组成的一种控制层次结构。

141.结构化语言是介于自然语言和形式语言之间的一种半形式的语言。

142.在面向对象方法中，对象实现了数据和操作的结合，使数据和操作封装于对象的统一体中。

143.划分模块时尽量做到高内聚、低耦合，保持模块的独立性。

144.瀑布模型是以文档为驱动、适合于需求很明确的软件项目的模型。

145.软件可维护性度量的七个质量特性是可理解性、可测试性、可修改性、可靠性、可移植性、使用性和效率。

146.软件开发模型用于指导软件的开发。演化模型是在快速开发一个可运行的软件原型的基础上，逐步演化成最终的软件。

147.软件工程学包括的三个要素是方法、工具和过程。

148.模块的分解和合并应力求降低耦合提高内聚。

149.在需求分析时，常采用数据流图来描绘系统的行为。

150.实体联系图包括：实体关系属性3种相互关联的信息。

151.基本路径测试是在程序流程图的基础上，通过分析控制构造的环路复杂性，从而导出基本路径的集合，从而设计测试用例，保证这些路径至少通过一次。

152. 软件维护活动主要包括完善性维护、适应性维护、改正性维护和预防性维护等四种。
153. 继承是子类自动共享父类属性和操作的机制。
154. IEEE定义基线是：已经通过了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。
155. 瀑布模型的核心思想是按工序将问题化简，将功能的实现与设计分开，便于分工协作，即采用结构化的分析与设计方法将逻辑实现与物理实现分开。
156. 软件生存周期是指软件从定义、开发、维护直到最终被废弃的全过程。
157. 软件生命周期由软件定义、软件开发和软件维护三个时期组成。
158. 一般来说，可行性研究应该从技术、经济和社会三个方面研究每种解法的可行性。
159. 软件需求分析时，应建立数据模型、功能模型和行为模型。
160. 软件可靠性时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。
161. 如果一个程序的代码块仅仅通过顺序、选择、和循环这三种基本控制结构进行连接，并且每个代码只有一个入口和一个出口，则称这个程序是结构化的。
162. 软件的维护一般经过提出维护或修改要求、领导复审并作出答复，如同意修改则列入维护计划、领导分配人物维护人员执行修改三个步骤。
163. 采用UML进行软件设计时，可用泛化关系来表示两类事物之间存在的特殊/一般关系，用聚集关系来表示事物之间存在的整体/部分关系。
164. 继承与多态机制，是面向对象程序中实现重用的主要手段。
165. 成本效益分析的目的是从经济角度评价开发一个软件项目是否可行。
166. 对象之间通过消息进行通信。

167.在面向对象的软件中，类是对具有相同数据和相同操作的一组相似对象的定义；实例是由某个特定的类所描述的一个具体对象。

168.需求分析阶段产生的最重要的文档是需求规格说明书。

169.软件开发中的原型是软件的一个早期可运行的版本，它反映了最终系统的重要特性。

170.软件开发过程来自用户方面的干扰和阻力主要有：不积极配合，求快求全和功能变化。

171.软件文档按照其产生和使用的范围可分为开发文档、管理文档和用户文档，其中可行性研究报告属于开发文档。

172.可行性研究实质上是要进行一次简化、压缩了的需求分析、设计过程，是在较高层次上以较抽象的方式进行需求分析和设计过程。

173.一个软件开发过程描述了“谁做”，“做什么”，“怎么做”和“什么时候做”，RUP用角色来表述“谁做”。

174.在面向对象的软件中，类是对具有相同数据和相同操作的一组相似对象的定义；实例是由某个特定的类所描述的一个具体对象。

175.软件工程的原则包括抽象、信息隐蔽、模块化、局部化、确定性、一致性、完备性和可验证性。

176.软件结构是以模块为基础而组成的一种控制层次结构。

177.软件能力成熟度模型（CMM）将软件能力成熟度自低到高依次划分为初始级、可重复级、定义级、管理级和优化级。

178.关联是建立类之间关系的一种手段，而链则是建立对象之间关系的一种手段。

179.软件测试方法中的计算机辅助静态分析属于静态测试方法。

180.继承与多态机制是面向对象程序中实现重用的主要手段。

- 181.系统详细设计阶段最后产生的文档是详细设计说明书。
- 182.DFD利用图形符号表示系统中各个元素，表达了系统中各个元素之间的信息流动。
- 183.在设计阶段，要充分体现软件工程的“模块化”、“抽象”、“信息隐蔽”等基本原则。
- 184.在面向对象分析和OOD中，通常将对象的操作称为方法或服务。
- 185.单元测试一般以白盒法为主，测试的依据是系统的模块功能规格说明。
- 186.当A模块调用B模块时，若两个模块之间传递的是数值型参数，则这两个模块的耦合方式是数据耦合。
- 187.软件测试用例主要由输入数据和预期输出结果两部分组成。
- 188.为适应软硬件环境变化而修改软件的过程是适应性维护。
- 189.提高程序效率的根本途径在于选择良好的设计方法、良好的数据结构和算法，而不是靠编程时对程序语句做调整。
- 190.一个模块把数值作为参数传送给另一个模块的耦合方式称为数据耦合。
- 191.成本效益分析的目的是从经济角度评价开发一个软件项目是否可行。
- 192.软件结构是以模块为基础而组成的一种控制层次结构。
- 193.在单元测试时，需要为被测模块设计驱动模块和桩模块的作用是返回被测模块所需的信息。
- 194.继承与多态性机制，是面向对象程序中实现重用的主要手段。
- 195.程序设计语言的发展大致分为四代：机器语言、汇编语言、高级语言和第四代语言。
- 196.程序设计算法的特点包括：有穷性，确定性，

有0个或多个输入，有一个或多个输出，有效性。

197.静态测试的目的是检查代码与设计的一致性，代码的可读性、代码的正确性、代码结构的合理性。

198.CMM指软件能力成熟度模型，其等级分为初始级、可重复级、已定义级、已管理级和优化级。

199.技术可行性是对要开发的项目的功能、性能、限制条件进行分析，确定在现有资源条件下，技术风险有多大，项目是否能实现。

200.模块之间的调用关系主要有直接调用、选择调用和循环调用。