

1.采用工程的概念、原理、技术和方法来
()和()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
好的技术方法结合起来,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。

- A、开发 测试 B、运行 管理
C、开发 管理 D、开发 维护

2.软件是程序、数据及相关文档的完整集合,
其中“开发、使用和维护程序所需要的图文
资料”指的是()

- A、程序 B、数据 C、文档 D、以上都不是

3.计算机系统经历了几个不同的发展阶段,
具有以下特点指的是()发展时期:小
规模团队开发,但仍然沿用早期形成的个
体化开发软件的方法。

- A、个体软件 B、软件作坊
C、软件危机 D、软件工程

4.以下()不是软件危机的表现形式。

- A、开发的软件不满足用户的需要
B、开发的软件可维护性差
C、开发的软件价格便宜
D、开发的软件可靠性差

5.软件产品与物质产品有很大区别,软件产
品是一种()产品

- A、有形 B、消耗 C、逻辑 D、文档

6.以文档作为驱动,适合于软件需求很明确
的软件项目的生存周期模型是()。

- A、喷泉模型 B、增量模型
C、瀑布模型 D、螺旋模型

7.下列软件开发模型中,适合于那些不能预
先确切定义需求的软件系统的开发的模型
是()。

- C、原型模型 B、瀑布模型
C、基于知识的智能模型 D、变换模型

8.下列软件开发模型中,以面向对象的软件
开发方法为基础,以用户的需求为动力,以
对象来驱动的模式是()。

- A、原型模型 B、瀑布模型
C、喷泉模型 D、螺旋模型

9.下列软件开发模型中,()只有在开发人
员具有风险分析和排除风险的经验及专门
知识时,才能获得成功。

- A、原型模型 B、瀑布模型
C、喷泉模型 D、螺旋模型

10.()是用户和设计交换最频繁的方法。

- A、原型模型 B、瀑布模型方法
C、螺旋模型方法 D、构件组装模型

11.传统的()要求,必须等前一阶段
的工作完成之后,才能开始后一阶段的工
作;前一阶段的输出文档,就是后一阶段的

输入文档。

- C、原型模型 B、瀑布模型
C、螺旋模型 D、喷泉模型

12.下列叙述中不属于软件生命周期模型
的是()

- A、瀑布模型 B、快速原型模型
C、风险模型 D、原子模型

13.“为了解决这个问题,目标系统必须做
什么?”这是生命周期中()阶段要确定
的事。

- A、需求分析 B、总体设计
C、详细设计 D、测试

14.()是为了获得高质量软件所需要完
成的一系列任务的框架,他规定了完成各项
任务的工作步骤。

- C、软件工程 B、软件过程
D、质量管理 D、软件方法学

15.下面不属于软件工程的3个要素是()

- A、工具 B、过程 C、方法 D、环境

16.软件工程方法学的目的是:使软件生产
规范化和工程化,而软件工程方法得以实施
的主要保证是()。

- A、硬件环境 B、软件开发的环境
C、软件开发工具和软件开发的环境
D、开发人员的素质

17.在软件生命周期法中,用户的参与主要在
()。

- A、软件定义期 B、软件开发期
C、软件维护期

D、整个软件生命周期过程中

18.软件开发的瀑布模型最典型地刻画了软
件生命周期的阶段划分,而与其最相适应的
软件开发的方法是()

- A、构件化方法 B、面向对象方法
C、结构化方法 D、原型化方法

19.在软件生存周期中,()阶段必须要回
答的问题是“要解决的问题是什么?”。

- C、详细设计 B、问题定义
C、概要设计 D、软件测试

20.数据存储和数据流都是(),仅仅是所
处的状态不同。

- A、分析结果 B、事件
C、动作 D、数据

21.数据流图和()共同组成系统的功能
模型。

- A、HIPO图 B、PDLC、数据字典 D、层次图

22.数据流图用于抽象描述一个软件的逻辑
模型,数据流图由一些特定的图形符号构成。
下列图符名标识的图符不属于合法数据流
图图符的是()。A、控制流

- B、加工 C、数据存储 D、数据源点/终点

23.结构化分析方法就是面向()的自顶向下逐步求精进行需求分析的方法。

A、目标 B、数据流 C、功能 D、对象

24.数据字典是对数据定义信息的集合,它所定义的对象都包含在()中。

A、数据流图 B、程序框图

C、软件结构 D、方框图

25.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()。

A、系统质量 B、系统目标

C、系统功能 D、系统性能

26.软件可行性一般不考虑()。

A、是否有足够的人员和相关的技术来支持系统开发

B、待开发的软件是否有质量问题

C、待开发的软件是否有市场,经济上是否合算

D、是否有足够的工具和相关的技术来支持系统开发

27.数据流图是进行软件需求分析常用的工具,其中最基本的图符是:处理,数据流,数据存储和数据源点/终点,其中“圆角矩形”可用来代表流图中的()。

A、处理 B、数据流

C、数据存储 D、数据源点/终点

28.数据流图是描述数据在软件中流动和变换的过程,而对数据流图所包含的元素的定义则是()

A、数据存储 B、数据加工

C、数据字典 D、数据定义

29.可行性研究的目的是不是解决问题,而是确定问题是否值得去解决,下列不是可行性研究的内容是()

A、需求分析 B、风险分析

C、资源分析 D、技术分析

31.一个软件项目是否进行开发的结论是在()文档中作出的。

C、软件开发计划 B、可行性分析报告

C、需求分析说明书 D、测试报告

32.下列叙述中, _____ 不属于数据字典的作用。

A、作为编码阶段的描述工具

B、为用户与开发人员之间统一认识

C、作为概要设计的依据

D、为需求分析阶段定义各类条目

33.不适于作为数据流图处理(加工)框名称的是_____。

A、人工校核工资数据 B、工资支票数据

C、打印工资支票

D、计算职工工资

34.系统流程图是概括地描述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系统组成系统的每个部件,其表达的是()

A、控制过程

B、对数据进行加工和处理的过程

C、程序的功能

D、数据在各部件之间的流动情况

35.虽然可以使用自然语言描述由数据元素组成数据的关系,但为了更加清晰简洁,建议采用符号来表示,其中 1{a}5 所代表的含义是()

A、a 重复出现,最少出现 1 次,最多出现 5 次

B、a 重复出现,只能出现 1 次,或 5 次

C、a 重复出现,其可以出现 1 次,也可以出现 5 次

D、a 是可选的,可以出现 2、3、4 次

36.()是可行性研究的一项重要内容,是客户组织负责人从经济角度判断是否继续投资于这项工程的主要依据。

A、纯收入计算 B、投资回收期分析

C、成本/效益分析 D、投资回收率分析

37.下面哪一项不属于可行性研究阶段的典型步骤()

A、研究目前正在使用的系统

B、草拟开发计划

C、书写文档提交审查

D、梳理模块间关系

38.复杂的数据流图()。

A、只有变换型 B、只有事务型

C、无规律 D、既有变换型又有事务型

39.()需求制定系统必须提供的服务,是对软件系统的一项基本需求,但却并不是唯一的需求。

C、功能需求 B、性能需求

C、可靠性和可用性需求 D、接口及约束

40.“系统每周 7 天、每天 24 小时都可以使用且运行无误”,代表的是()

C、功能需求 B、性能需求

C、可靠性和可用性需求 D、接口及约束

41.“系统应在 20 秒之内响应所有的请求”,代表的是()

C、功能需求 B、性能需求

C、可靠性和可用性需求 D、接口及约束

42.“系统出现异常时,应能为用户给出指导性的提示,并及时保存当前的数据”代表的是()

A、出错处理需求 B、性能需求

C、可靠性和可用性需求

D、接口及约束

43.与用户沟通获取需求的方式有很多,其中()不属于获取需求的方式。

A、正式或非正式的访谈

B、自底向上的求精方法

C、快速建立软件原型

D、建议的应用规格说明技术

44.“快速建立软件原型”是一种比较准确、有效、强大的需求分析技术,为了快速的构建和修改原型,通常采用3种方法和工具,以下哪项是无效的()

A、可重用的软件构件

B、形式化需求规格说明和环境原型

C、高级语言选择

D、第四代技术

45.通常用“范式”定义消除数据冗余的程度,其中第三范式要求符合第二范式的条件,而且()

A、每个属性值都必须是原子值

B、每个非关键字属性值都由整个关键字决定

C、非关键字不是由关键字的一部分来决定

D、一个非关键字属性值不依赖于另一个非关键字属性值

46.结构化分析阶段要建立中模型,分别是()

A、功能模型、数据模型、状态模型

B、功能模型、行为模型、数据模型

C、功能模型、数据模型、动态模型

D、对象模型、动态模型、功能模型

47,()是对引起系统做动作或(和)从一个状态转换到另一个状态的外界事情的抽象。

A、事件 B、状态 C、初态 D、终态

48.进行需求分析可使用多种工具,但()是不适用的。

A、数据流图 B、E-R图

C、PAD图 D、数据字典

49.E-R图是用于建立数据模型的图形,其中包括3种相互关联的信息,分别是()

A、数据、对象、实体

B、控制、联系、对象

C、实体、联系、属性

D、实体、属性、操作

50.描述复杂的事物时,图形远比文字叙述优越得多,在需求分析阶段可以使用()等图形工具。

A、层次方框图和判定表

B、数据流图和程序流程图

C、IPO图和层次方框图

D、warnier图和PDL

51.结构化分析方法以数据流图、()和加工说明等描述工具,即用直观的图和简洁的语言来描述软件功能模型。

A、DFD图 B、PAD图

C、HIPO图 **D、数据字典**

52.软件需求说明书中包括多方面的内容,下述()不是软件说明书中应包括的内容。

A、数据描述 B、功能描述

C、性能描述 **D、结构描述**

53.软件需求分析阶段建立原型的主要目的是()。

A、确定系统的功能和性能要求

B、确定系统的性能要求

C、确定系统是否满足用户要求

D、确定系统是否开发人员需要

54.需求分析是由分析人员经了解用户的需求,认真仔细的调研、分析,最终应建立目标系统的逻辑模型并写出()。

A、模块说明书

B、需求规格说明书

C、项目开发设计

D、合同文档

55.使用结构化分析方法时,采用的基本手段是()。

C、**分解和抽象** B、分解和综合

C、归纳与推导 D、试探与回溯

56.分层数据流图是一种比较严格又易于理解的描述方式,它的顶层描述了系统的_____。

A、总貌 B、细节 C、抽象 D、软件的作者

57.下列关于软件设计准则的描述,错误的是()。

A)提高模块的独立性

B)深度、宽度、扇出和扇入适当

C)使模块的控制域在该模块的作用域内

D)设计成单入口、单出口的模块

58.下面关于“深度、宽度、扇出和扇入”描述错误的是()

A、深度表示软件结构中控制的层数

B、宽度是软件同一层次上的模块总数的最大值

C、扇出是一个模块调用的模块数目

D、扇入是直接调用这个模块的上级模块的数目

59.结构化设计(SD)是软件设计阶段所使用的方法,()不是SD所使用的工具。

A、DFD图

B、PAD图

C、结构化的过程设计语言

D、判定树

60.结构化设计方法采用变换分析和事务分析技术实现_____。

- A、从数据结构导出程序结构
- B、从模块结构导出程序结构
- C、从模块结构导出数据结构
- D、从数据流图导出软件结构

61.把现实世界中的事物相似的地方集中和概括起来,暂时忽略他们之间的差异,这就是()。

- A、抽象 B、模块化
- C、自底向上 D、信息隐蔽

62.()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,是构成程序的基本构件。

- A、模块化 B、抽象 C、精化 D、模块

63.()是指让一些关系密切的软件元素在物理上彼此靠近。

- A、信息隐蔽 B、内聚
- C、局部化 D、模块独立

64.“为了能集中精力解决主要问题而尽量推迟对问题细节的考虑”,这是对()的定义。

- A、信息隐蔽 B、逐步求精
- C、局部化 D、模块独立

65.软件设计中划分模块的一个准则是()。

- A、低内聚低耦合 B、低内聚高耦合
- C、高内聚低耦合 D、高内聚高耦合

66.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()

- A、顺序内聚 B、过程内聚
- C、逻辑内聚 D、功能内聚

67.如果一个模块内部的处理元素和同一个功能密切相关,而且这些处理必须顺序执行(通常一个处理元素的输入作为下一个处理元素的输出),则称为()。

- A、逻辑内聚 B、时间内聚
- C、过程内聚 D、顺序内聚

68.如果一个模块完成的任务在逻辑上属于相同或相似的一类,则称为()

- A、逻辑内聚 B、时间内聚
- C、过程内聚 D、顺序内聚

69.如果模块中所有元素都是用同一个输入数据和(或)产生同一个输出数据,则称为()。

- A、通信内聚 B、时间内聚
- C、过程内聚 D、顺序内聚

70.耦合是软件各个模块间连接的一种度量。一组模块都访问同一公有环境数据结构应属于()方式。

- A、内容耦合 B、公共耦合
- C、外部耦合 D、控制耦合

71.耦合是软件中各种模块间相互联系的一种度量,耦合的强弱取决于模块间的复杂程度,在下列耦合中,耦合度最低的是()

- A、内容耦合 B、公共耦合
- C、数据耦合 D、控制耦合

72.把整个数据结构作为参数传递而被调用的模块只需要使用其中一部分数据元素,称之为()

- A、内容耦合 B、公共耦合
- C、特征耦合 D、控制耦合

73.模块的功能是打印录取通知书或不录取通知书,调用模块可通过向此模块传递布尔标志信息决定本次调用哪种通知书的打印,这种模块之间的耦合称之为()。

- C、数据耦合 B、混合耦合
- C、公共耦合 D、控制耦合

74.数据沿着输入通路到达一个处理,这个处理根据输入数据的类型在若干动作序列中选出一个来执行,这类数据流称为()

- C、变换流 B、变换中心
- C、事务中心 D、事务流

75.信息沿输入通路进入系统,同时由外部形式变换成内部形式,进入系统的信息经过变换中心,经过加工处理后再沿输出通路变换成外部形式离开软件系统,这种信息流称为()。

- C、变换流 B、变换中心
- C、事务中心 D、事务流

76.软件的结构化设计(SD)方法中,一般分为总体设计和详细设计两个阶段,其中总体设计主要是要建立()。

- A、软件结构 B、软件流程
- C、软件模型 D、软件模块

77.面向数据流的设计方法把()映射成软件结构。

- A、软件结构 B、系统结构
- C、控制结构 D、信息流

78.详细设计的基本任务是确定每个模块的()设计。

- C、功能 B、调用关系
- C、输入输出数据 D、算法

79.结构化程序设计思想的核心是要求程序只有顺序,循环和()三种结构组成。

- A、分支 B、单入口
- C、单出口 D、有规则 GOTO

80.()的结果基本上决定了最终的程序代码的质量。

- C、概要设计 B、详细设计
- C、编码 D、测试

81. Jackson 设计方法是由英国的 M. Jackson 提出的, 它是一种面向 () 的软件设计方法。

- C、对象 B、数据流
C、数据结构 D、控制结构

82. () 阶段是为每个模块完成的功能进行具体的描述, 要把功能描述转变为精确的、结构化的过程描述。

- A、概要设计 B、详细设计
C、编码 D、测试

83. 结构化程序从理论上可以不使用 () 语句。当然程序是否使用该语句是无关紧要的, 重要的是程序的结构是否严谨、清晰。

- A、AND B、IF
C、GOTO D、FOR

84. PAD (Problem Analysis Diagram) 图是一种 () 工具。

- C、系统描述 B、详细设计
C、测试 D、编程辅助

85. 下列系统设计工具中, () 不属于详细设计工具。

- C、N-S 图 B、数据流图
C、程序流程图 D、PAD

86. () 应考虑对模块相联和资源共享问题进行描述和制约。

- A、系统设计 B、详细设计
C、接口控制 D、结构化编辑工具

87. 程序流程图是一种传统的程序设计表示工具, 有其优点和缺点, 使用该工具时应注意 ()

- A、支持逐步求精 B、考虑控制流程
C、遵守结构化设计原则 D、数据结构表示

88. 软件设计中设计复审是和设计本身一样重要的环节, 其主要目的和作用是 ()

- A、减少测试工作量
B、减免后期付出高代价
C、保证软件质量
D、缩短软件开发周期

89. 软件设计中, 可应用于详细设计的工具有 ()

- A、程序流程图、PAD、方框图和伪码
B、数据流程图、PAD、结构图和伪码
C、业务流程图、N-S 图和伪码
D、数据流程图、PAD、N-S 图和伪码

90. 详细设计是软件总体设计滞后的工作。下列可选内容中, () 不属于详细设计内容。

- A、算法设计
B、程序流程设计
C、界面设计
D、功能设计

91. 结构化程序设计的一种基本方法是 ()

- A、筛选法 B、递归法
C、迭代法 D、逐步求精法

92. “实现”在软件生命周期中横跨两个阶段, 分别是 ()。

- C、编码和维护 B、测试和维护
C、编码和测试 D、调试和维护

93. 20 世纪 50 年代以来, 出现了许多不同的程序设计语言, 下列语言中哪个语言是低级语言 ()。

- C、PASCAL B、VISUAL BASIC
C、C++ D、汇编语言

94. 把软件设计结果翻译成用某种程序设计语言书写的程序的过程, 称之为 ()。

- A、需求分析 B、详细设计
C、编码 D、测试

95. 软件测试是为了 () 而执行程序的过程。

- A、证明软件的正确性
B、发现程序中的错误
C、证明软件系统中存在错误
D、表明程序是正确的

96. 使用白盒测试方法时, 确定测试数据应根据 () 和指定的覆盖标准。

- C、程序的内部逻辑 B、程序的复杂程度
C、该软件的编辑人员 D、程序的功能

97. 检查软件产品是否符合需求定义的过程称为 ()

- A、确认测试 B、集成测试
C、系统测试 D、模块测试

98. 单元测试又称为 (), 一般用白盒法测试

- A、集成测试 B、模块测试
C、系统测试 D、白盒法

99. 对软件是否能达到用户所期望的要求的测试称为 ()

- A、集成测试 B、系统测试
C、有效性测试 D、验收测试

100. 以下 () 不属于白盒测试技术

- A、基本路径测试 B、边界值分析
C、条件覆盖测试 D、逻辑覆盖测试

101. () 方法需要考察模块间的接口和各个模块之间的关系。

- A、单元测试 B、集成测试
C、确认测试 D、验收测试

102. 软件测试是软件开发过程中重要的和不可缺少的阶段, 其包含的内容和步骤甚多, 而测试过程的多种环节中最基础的是 ()

- A、集成测试 B、单元测试
C、系统测试 D、验收测试

103.可以提高软件测试效率的是()

A、随意选取测试的数据

B、制定测试计划

C、选取边界数据作为测试用例

D、取尽可能多的数据进行测试

104.软件测试采用白盒法的是()

A、单元测试 B、集成测试

C、确认测试 D、系统测试

105.集成测试有两个具体办法,它们是()

A、非渐增式方式和渐增式方式

B、白盒法和黑盒法

C、确认测试和系统测试

D、归纳法和演绎法

106.软件测试方法一般有黑盒测试和白盒测试,而其中黑盒测试主要是依据程序的()

A、结构 B、流程 **C、功能** D、逻辑

107.程序测试是一项复杂的工作,一般测试者在进行测试中都需要设计()

A、数据结构 **B、测试用例**

C、测试阶段 D、测试方法

108.调试是指在测试发现错误之后排除错误的过程,()不属于调试途径。

A、蛮干法 B、回溯法

C、原因排除法 **D、运用测试实例法**

109.所谓逻辑覆盖测试是对一系列测试过程的总称,其中:被测程序中每个语句至少执行一次,称之为()。

A、语句覆盖 B、判定覆盖

C、条件覆盖 D、条件组合覆盖

110.所谓逻辑覆盖测试是对一系列测试过程的总称,其中:每个判定的每个分支都至少执行一次,称之为()。

A、语句覆盖 **B、判定覆盖**

C、条件覆盖 D、条件组合覆盖

111.所谓逻辑覆盖测试是对一系列测试过程的总称,其中:不仅每个语句至少执行一次,而且使判定表达式中的每个条件都取到各种可能的值,称之为()。

A、语句覆盖 B、判定覆盖

C、条件覆盖 D、条件组合覆盖

112.所谓逻辑覆盖测试是对一系列测试过程的总称,其中:每个判定表达式中条件的各种可能组合都至少出现一次,称之为()。

A、语句覆盖 B、判定覆盖

C、条件覆盖 D、**条件组合覆盖**

113.()是一种黑盒测试技术,这种技术把程序的输入域划分成若干个数据类,据此导出测试用例。

A、等价划分 B、边界值分析

C、错误推测 D、路径测试

114.所谓()是指在测试发现错误之后排除错误的过程。

A、调试 B、纠错

C、评估 D、维护

115.所谓()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。

A、调试 B、纠错

C、测试 **D、维护**

116.为了和变化了的环境适当地配合而进行的修改软件的活动称之为(),是既必要又经常的维护活动。

A、改正性维护 **B、适应性维护**

C、完善性维护 D、预防性维护

117.在任何大型程序的使用期间,用户必然会发现程序错误,并且把他们遇到问题报告给维护人员。把诊断和改正错误的过程称为()。

A、改正性维护 B、适应性维护

C、完善性维护 D、预防性维护

118.在使用软件的过程中用户往往提出增加新功能或修改已有功能的建议,还可能提出一般性的改进意见。为了满足这类要求,需要进行()。

A、改正性维护 B、适应性维护

C、完善性维护 D、预防性维护

119.为了改进未来的可维护性或可靠性,或为了给未来的改进奠定更好的基础而修改软件而进行的维护活动称之为()。

A、改正性维护 B、适应性维护

C、完善性维护 **D、预防性维护**

120.()指的是外来读者理解软件的结构、功能、结构和内部处理过程的难易程度。

A、可测试性 B、可修改性

C、可理解性 D、可移植性

121.()指的是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。

A、可测试性 B、可修改性

C、可理解性 **D、可移植性**

122.软件容易修改的程度称之为(),耦合、内聚、信息隐藏、局部化.....等都会对其产生影响。

A、可测试性 **B、可修改性**

C、可理解性 D、可移植性

123.()是指同一事物不做修改或稍加改动就在不同环境中多次重复使用。

A、可测试性 **B、可重用性**

C、可理解性

D、可移植性

124. 软件维护工作的最主要部分是()。
A、校正性维护 B、适应性维护
C、完善性维护
D、预防性维护

125. 软件调试的目的是()
A、发现错误 B、改正错误
C、改善软件的性能
D、挖掘软件的潜能

126. 如果软件配置的唯一成分是()，那么这种维护属于非结构化维护。
A、需求文档 B、设计文档
C、程序代码
D、测试用例

127. 以下文档，() 不属于用户文档包括的内容。
A、功能描述 B、安装文档
C、需求说明
D、使用手册

128. 以下文档，() 不属于系统文档包括的内容。
A、软件需求规格说明 B、软件使用手册
C、软件详细设计
D、软件测试用例

129. 面向对象开发方法不需要建立的模型是()。
A、对象模型 B、动态模型
C、功能模型
D、结构模型

130. 对象和类之间存在着关联关系，利用对象的()特征可以实现对象的多态性。
A、惟一性
B、分类性
C、继承性
D、封装性

131. 面向对象的主要特征除对象惟一性、封装、继承外，还有()。
A、多态性
B、完整性
C、可移植性
D、兼容性

132. 描述对象模型静态特征的工具是()。
A、DFD B、类图 C、状态图
D、DD

133. 每个对象可用它自己的一组属性和它可以执行的一组()来实现。
A、状态
B、功能
C、操作
D、数据

134. 一个类的上层可以有父类，下层可以有

子类，这种层次结构系统的一个重要性质是(): 一个类获得其父类的全部描述(数据和操作)。

A、传递性 B、继承性
C、复用性
D、并行性

135. ()模型表示了对象的相互行为。

A、对象 B、动态
C、功能
D、分析

136. 软件部分的内部实现与外部可访问性分离，这是指软件的()

A、继承性 B、共享性
C、封装性
D、抽象性

137. 面向对象方法学中，功能模型的描述工具是()

A、动态图 B、结构图
C、用例图
D、程序流图

138. 面向对象方法学中，动态模型的描述工具是()

A、对象图 B、结构图
C、状态图
D、设计图

139. 汽车有一个发动机，汽车与发动机之间的关系是()关系。

A、一般与具体 B、整体与部分
C、分类关系
D、关联关系

140. 火车是一种陆上交通工具，火车和陆上交通工具之间的关系是()关系。

A、组装整体与部分 B、具体与一般
C、共享聚集
D、普通关联

141. ()就是对具有相同数据和操作的一组相似对象的定义。

A、继承 B、封装
C、多态 D、类

142. 一个面向对象的系统的体系结构通过它的()的关系确定。

A、类与对象 B、对象与对象
C、过程与对象 D、类与界面

143. 所谓管理就是通过()、()和()等一系列活动，合理地配置和使用各种资源，以达到既定目标的过程。

A、计划、组织、控制
B、讨论、计划、执行
C、分析、设计、执行
D、分析、计划、组织

144. 软件项目管理先于任何技术活动之前开始, ()。

- A、贯穿于整个需求分析阶段
- B、贯穿于整个详细设计阶段
- C、贯穿于整个可行性分析阶段
- D、贯穿于软件整个生命周期

145. () 技术依据以往开发类似项目的经验和历史数据, 估计实现一个功能所需要的源程序行数。

- A、代码行技术
- B、专家估计法
- C、PERT 方法
- D、功能点法

146. 功能点技术定义了信息域的 5 个特性, 分别是: 输入项数, 输出项数, 查询, 主文件数和 ()。

- A、内部接口数
- B、内部查询数
- C、外部数据库数
- D、外部接口数

147. “软件编码完成之后, 我才可以对它进行软件测试”, 这句话说明了编码和测试属于那种逻辑关系? ()。

- A、结束-开始
- B、开始-结束
- C、结束-结束
- D、开始-开始

148. “硬件开始安装了, 才能进行软件安装”, 这句话说明了硬件开始和软件安装属于那种逻辑关系? ()。

- A、结束-开始
- B、开始-结束
- C、结束-结束
- D、开始-开始

149. 赶工一个任务时, 应该关注 ()。

- A、尽可能多的任务
- B、非关键任务
- C、加速执行关键路径上的任务
- D、通过成本最低化加速执行任务

150. 项目质量管理的最终责任由谁来承担 ()。

- A、项目开发人员
- B、采购经理
- C、质量经理
- D、项目经理

150. 质量与等级的区别是什么 ()。

- A、质量是对需求的满足程度, 而等级是对质量的排序。
- B、质量是与用户需求的一致程度, 而等级是对服务的排序。
- C、在项目管理过程中, 质量和等级没有明显的区别。
- D、质量是项目的最终结果, 而等级是项目进展过程中对质量的排序。

1. 软件的本质特性: 1. 软件工程关注于大型程序的构造; 2. 软件工程的中心课题是控制复杂性; 3. 软件经常变化; 4. 开发软件的效率非常重要; 5. 和谐地合作是开发软件的关键; 6. 软件必须有效地支持它的用户; 7. 在软件工程领域中通常由具有一种文化背景的人替具有另一种文化背景的人创造产品。

2. 软件工程的基本原理: 1. 用分阶段的生命周期计划严格管理; 2. 坚持进行阶段评审; 3. 实行严格的产品控制; 4. 采用现代程序设计技术; 5. 结果应能清楚地审查; 6. 开发小组的人员应该少而精; 7. 承认不断改进软件工程实践的必要性。

3. 获取需求的方法: 1. 访谈; 2. 面向数据流自顶而下求精; 3. 简易的应用规格说明技术; 4. 快速建立软件原型。

4. 需求分析的任务: 1. 确定对系统的综合要求; 2. 分析系统的数据要求; 3. 导出系统的逻辑模型; 4. 修正系统开发计划。

5. 面向数据流的设计方法步骤: 1. 复查基本系统模型; 2. 复查并精化数据流程图; 3. 确定数据流程图具有变换特性还是事务特性; 4. 确定输入流和输出流的边界, 从而孤立出变换中心; 5. 完成“第一级分解”; 6. 完成“第二级分解”; 7. 使用设计度量和启发式规则对第一次分割得到的软件结构进一步精化。

6. 出错信息应具备的属性: 1. 信息应该用用户可以理解的术语描述问题; 2. 信息应该提供有助于从错误中恢复的建设性意见; 3. 信息应该指出错误可能导致哪些负面后果, 以使用户检查是否出现了这些问题, 并在确定出现问题时及时解决; 4. 信息应该伴随着听觉上或视觉上的提示; 5. 信息不能带有指责色彩, 也就是说, 不能责怪用户。

7. 用户帮助设施要解决的问题: 1. 在用户与系统交互期间, 是否在任何时候都能获取关于系统任何功能的帮助信息? 有两种选择: 提供部分功能的帮助信息和提供全部功能的帮助信息; 2. 用户怎样请求帮助? 有 2 种选择: 帮助菜单, 特殊功能键和 HELP 命令; 3. 怎样显示

帮助信息？有 3 种选择：在独立的窗口中，指出参考某个文档（不理想）和在屏幕固定位置显示简短提示；4.用户怎样返回到正常的交互方式中？有 2 种选择：屏幕上的返回按钮和功能键；5.怎样组织帮助信息？有 3 种选择：平面结构（所有信息都通过关键字访问），信息的层次结构（用户可在该结构中查到更详细的信息）和超文本结构。

8. 软件维护过程：维护组织；维护报告；维护的事件流。

9.软件再工程过程：库存目录分析；文档重构；逆向工程；代码重构；数据重构；正向工程。

10.面向对象方法学的定义：尽可能模拟人类习惯的思维方式，使开发软件的方法与过程尽可能接近人类认识世界解决问题的方法与过程，也就是使描述问题的问题空间与实现解法的解空间在结构上尽可能一致。

优点：与人类习惯的思维方法一致；稳定性好；可重用性好；较易开发大型软件产品；可维护性好。

11.什么是建模？为什么要建模？

12.软件质量是：就是“软件与明确地和隐含地定义的需求相一致的程度”，更具体说，它是软件与明确地叙述的功能和性能需求、文档中明确描述的开发标准以及任何专业的软件产品都应该具有的隐含特性相一致的程度。强调 3 个要点：1.软件需求是度量软件质量的基础，与需求不一致就是质量不高；2.指定的开发标准定义了一组指导软件开发的准则，如果没有遵守这些准则，肯定会导致软件质量不高；3.通常，有一组没有显式描述的隐含需求。如果软件满足明确描述的需求，但不满足隐含的需求，那么软件的质量仍然是值得怀疑的。

它与软件可靠性的关系：

13.一个程序能既正确又不可靠吗？

所谓软件可靠性,是程序在给定的时间间隔内按照规格说明书的规定成功地运行的概率.通常认为,软件可靠性既包含正确性又包含健壮性,也就是说,不仅在预定环境下程序应该能正确地、完成预期功能,而且在硬件发生故障,输入的数据无效或用户操作错误等意外环境下,程序也应该能做出适当的响应.如果一个程序在预定环境下能够正确地、完成预期的功能,但是在意外环境下不能做出适当的响应,则该程序就是既正确又不可靠。