

## 判断题：

1. 软件是一种抽象的逻辑产品。( T )
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。( F )
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。( T )
4. 软件危机就是大量软件出现严重问题不能继续再使用。( F )
5. 软件工程有 7 个基本要素。( F ) 3 个 方法、工具、过程
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。( T )
7. 瀑布模型存在的最主要的问题是缺乏灵活性。( T )
8. 在软件工程中，软件是指程序+数据+文档。( T )
9. 软件就是指程序。( F )
10. CASE 即计算机辅助软件工程的英文缩写。( T )
11. 瀑布模型也称为是基于文档驱动模型。( T )
12. 快速原型模型也称为是基于文档驱动模型。( F )
13. 软件生存周期是指开发软件的全部时间。( F )
14. 软件工程在软件设计时把软件的效率放在第一位。( T )
15. 可行性分析研究的目的是规划如何开发项目。( F )
16. 可行性分析研究的目的是探讨项目值得开发否。( T )
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。( T )
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。( F )
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。( T )
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。( F ) 数据流是动态数据，数据存储是静态数据
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。( T )
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。( T )
23. 数据流图中可以有控制流。( F )
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。( T )
25. DFD 中的每个加工至少有一对输入流和输出流。( T )
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。( T )
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。( F )
28. 需求分析最终结果是产生 SRS。( T )
29. 需求分析最终结果是产生项目开发计划。( F )
30. 软件设计基本原理中不包括自底向上的设计思路。( T )
31. 软件设计基本原理中包括自底向上的设计思路。( F )
32. 模块的独立性高意味着模块的内聚弱，耦合高。( F )
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。( T )
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。( T )

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。( F )
36. 程序流程图(PDF)中的箭头代表的是程序的控制流程。( T )
37. 模块本身的内聚是模块独立性的一个重要度量因素之一。在 7 类内聚中，具有最强内聚的一类是过程性内聚。( F ) 功能内聚
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。( T )
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。( T )
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。( F )
41. 内聚和耦合都是用于度量模块复杂性的两个定性指标。( F ) 度量模块独立程度
42. 判定所在的模块称为判定的作用范围。( F )
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。( T )
44. 软件详细设计主要采用的方法是流程图设计法。( T )
45. 程序的三种基本控制结构是顺序、选择和重复。( T )
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。( T )
47. 结构化程序设计主要强调的是程序的清晰易读性 ( T )。
48. 软件详细设计的主要任务是确定每个模块的功能。( F )。算法和使用的数据结构
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。( F ) 确认测试
50. 测试是为了证明程序中没有错误而执行程序的过程。( F )
51. 软件调试的目的是改正软件中存在错误。( T )
52. 测试用例就是一组测试用的数据。( F ) 测试数据+预期结果
53. 测试用例就是一组测试用的数据及其预期结果。( T )
54. 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。( F )  
指定的覆盖标准、程序的内部逻辑
55. 测试的关键问题是如何选择测试用例。( T )
56. 测试的关键问题是用测试用例来驱动测试被测对象。( F )
57. 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。( F )
58. 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。( F ) 适应性维护
59. 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。( T )
60. UML 是统一建模语言的英文简写。( T )
61. 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。( F )
62. 在面向对象技术中，对象之间利用数据进行通信。( F ) 消息
63. 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。( F )  
版本管理
64. 软件能力成熟度模型 CMM 分为 5 个等级。( T )
65. 螺旋模型是在瀑布模型和增量模型的基础上增加了风险分析活动。( T )
66. 软件是指用程序设计语言编写的程序，软件开发实际上就是编写程序代码。( F )
67. 过程描述语言可以用于描述软件的系统结构。( F )
68. 数据库设计说明书是一个软件配置项 ( T )
69. 软件模块之间的耦合性越弱越好。( T )
70. 如果通过软件测试没有发现错误，则说明软件是正确的。( F )
71. 软件概要设计包括软件系统结构设计以及数据结构和数据库设计。( T )
72. 数据流图(DFD)和程序流程图没有区别。( F )
73. 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 )

74. 白盒测试无需考虑模块内部的执行过程和程序结构，只要了解模块的功能即可。( F )

75. 软件危机是指在软件开发和维护过程中遇到的一系列严重问题。( T )

76. 结构化分析方法就是用抽象模型的概念，按照软件内部数据传递、变换的关系，自顶向下逐层分解，直到找到满足功能要求的所有可实现的软件为止。( T )

77. 程序设计语言中应绝对禁止使用 GOTO 语句。( F )

78. 模块越小，模块的优点越大。( F )

79. 集成测试主要由用户来完成。( F )

80. 确认测试计划应该在可行性研究阶段制定 ( F )

81. 快速原型模型可以有效地适应用户需求的动态变化。( T )

82. 需求规格说明书不是一个软件配置项。( F )

83. 黑盒测试无需考虑模块内部的执行过程和程序结构，只要了解模块的功能即可。( T )

84. UML 只能应用于软件系统模型的建立。( F )

85. 在进行需求分析时需同时考虑维护问题。( T )

86. 完成测试作业后，为了缩短源程序的长度应删去源程序的注解。( F )

87. 尽可能在软件生产过程中保证各阶段文档的正确性。( T )

88. 编码时应尽可能使用全局变量。( F )

89. 选择时间效率和空间效率尽可能高的算法。( F )

90. 尽可能使用硬件的特点。( F )

91. 重视程序结构的设计，使程序具有较好的层次结构。( T )

92. 使用维护工具或支撑环境。( T )

93. 在进行概要设计时应加强模块间的联系。( F )

94. 为了提高程序的易读性，尽可能使用高级语言编写程序。( T )

95. 为了加快软件维护作业的进度，尽可能增加维护人员的数目。( F )