

# 软件工程往年真题

2019解析

## 一、单选题(每小题2分,共40分)

1. 通过对软件的测试,可以证明( )  
A. 程序正确性      B. 错误不存在  
C. 错误存在      D. 不含有隐患
2. 某程序功能说明中列出“规定每个运动员参赛项目为1~3项”,应用黑盒法中的等价分类法确定等价类是( )  
A.  $1 \leq \text{项目数} \leq 3$       B. 项目数 $<1$   
C. 项目数 $>3$       D. 以上都是
3. 协作图反映收发消息的对象的结构组织,它与( )是同构的  
A. 用例图      B. 类图      C. 活动图      D. 时序图
4. 软件维护,可按不同的维护目的而分类,为了适应硬件环境或软件环境的变更对软件作修改是( )  
A. 纠正性维护      B. 适应性维护  
C. 完善性维护      D. 预防性维护
5. 要计算软件开发项目的关键路径,应选择( )  
A. Gantt图      B. 工程网络图  
C. COCOMO模型      D. 数据流图
6. 在面向对象软件开发方法中,能够表示类与类之间“部分与整体”的关系是( )  
A. 关联      B. 继承      C. 聚集      D. 依赖
7. 在使用UML设计分析类图时,经理类与雇员类的“管理”关系应该是( )  
A. 关联关系      B. 继承关系  
C. 依赖关系      D. 细化关系
8. 适于软件检验的手段有多种,通过人工来评审文档或程序,借以发现其中错误,该手段是( )  
A. 黑盒法      B. 正确性证明  
C. 动态检查      D. 静态检查
9. 以下不是软件工程的三个要素是( )  
A. 方法      B. 文档      C. 过程      D. 工具

1. 软件测试目标（P110）

- 1. 为了发现程序中的错误而执行程序
- 2. 好的测试方案极可能发现迄今为止尚未发现的错误
- 3. 成功的测试方法是发现了迄今为止尚未发现的错误

B

2. 等价划分方法（P121）

- 1. 如果规定输入范围，可以划分出一个有效的等价类（输入值在此范围内），两个无效的等价类（输入值小于最小值或输入值大于最大值）
- 2. 如果规定输入数据的个数，可类似操作，划分出一个有效的等价类和两个无效的等价类。
- 3. 如果规定输入数据的一组值，程序对不同输入处理不同，每个允许的输入值是一个有效的等价类，任一个不允许的输入构成无效类集合。
- 4. 如果规定输入数据必须遵循的规则，则可以划分出一个有效的等价类（符合规则）和若干个无效的等价类（从各种不同角度违反规则）
- 5. 如果规定了输入数据为整型，则可以划分出正整数、零、负整数3个有效类。

D

3. 协作图（P248）

协作图	顺序图（时序图）	关系
用于描述相互协作的对象间的交互关系和连接关系。	描述对象间的动态交互关系，着重表现对象间消息传递的时间顺序。	顺序图着重表现时间顺序，协作图着重表现交互对象的静态连接关系。

D

4. 维护分类（P303）

类别	作用
纠错性维护	识别并纠正潜藏的错误，改进性能缺陷。
适应性维护	适应软硬件环境变更。
完善性维护	针对用户对软件产品提出的新需求所进行的维护。
预防性维护	为了提高软件的可维护性和可靠性。

B

5. 进度计划 & 工作量估算（P260）

模型	作用
COCOMO2	估算工作量和成本。
Gantt图	形象描绘任务分解情况，以及每个子任务的开始时间和结束时间。
工程网络	通过 <u>关键路径</u> 显式描述各个作业彼此间的依赖关系。

B

## 6. 类间关系（P158）

类间关系	子关系	作用	举例
关联	普通关联	存在连接关系，最为常见。	一个作家可以使用1到多台计算机，一台计算机可被0至多个作家使用。
	关联类	引入关联类记录附加信息。	一个电梯控制器控制4部电梯，每个队列存储来自控制器和电梯内部的请求服务信息。队列是关联类。
	聚集	类与类之间的关系是整体与部分的关系，包括共享和组合两种聚集关系。	共享：一个课题组包含许多成员，每个成员又可以是另一个课题组的成员；组合：在屏幕上打开一个窗口，它由组件组成。一旦关闭了窗口，各个组成部分也消失。
泛化（继承）	-	继承关系。	汽车和船继承交通工具。
依赖	依赖	其中一个模型元素是独立的，另一个模型元素不是独立的，它依赖于独立的模型元素。	友元依赖关系：使B类的操作可以使用A类私有或保护的成员；
	细化	细化是UML中的术语，表示对事物更详细一层的描述。	设计类是对分析类的细化。

C

## 7. 类间关系（P158）

一个经理管理多个雇员，每个雇员可以由多个经理，可以是普通关联关系。

A

## 8. 静态测试 & 动态测试 (CSDN)

静态测试：

1. 代码检查：代码会审、代码走查、桌面检查；
2. 静态结构分析；
3. 代码质量度量。

动态测试：

1. 黑盒测试：又称功能测试。这种方法把被测软件看成黑盒，在不考虑软件内部结构和特性的情况下测试软件的外部特性。
2. 白盒测试：又称结构测试。这种方法把被测软件看成白盒，根据程序的内部结构和逻辑设计来设计测试实例，对程序的路径和过程进行测试。

D

## 9. 软件工程三要素 (CSDN)

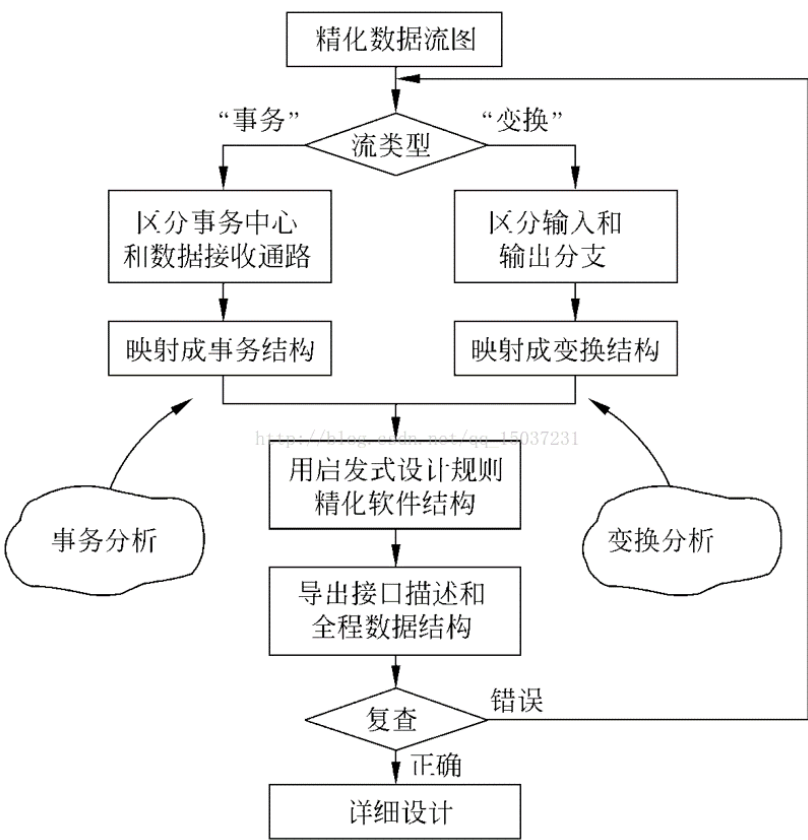
方法、工具、过程。

B

10. 面向数据流的设计方法描述错误的是( )  
A. 实际设计中常常把变换分析和事务分析应用到同一个数据流图的不同部分  
B. 变换数据流中有明确的事务中心  
C. 需要区分变换分析和事务分析  
D. 能够由数据流图生成软件结构
11. UML 的用例图中用例之间可以出现下列哪个关系( )  
A. <<aggregate>>  
B. <<inherit>>  
C. <<extend>>  
D. <<composite>>
12. 把一组具有相似特性的对象组合在一起,称之为( )  
A. 多态性  
B. 继承性  
C. 类  
D. 消息
13. 结构化程序之所以具有易于阅读,并且有可能验证其正确性,这是由于( )  
A. 它强调编程风格  
B. 选择良好的数据结构和算法  
C. 有限制地使用 GOTO 语句  
D. 只有三种基本结构
14. 如果一个模块包括并仅包括为完成某一个具体任务所必需的的所有成分,则这种模块的内聚性属于( )  
A. 顺序型  
B. 通讯型  
C. 逻辑型  
D. 功能型
15. 以下不是敏捷软件开发方法的是( )  
A. 极限编程(XP)  
B. 微软过程(MSF)  
C. Rational 统一过程(RUP)  
D. 快速原型模型
16. 对象实现了数据和操作的结合,使数据和操作( )于对象的统一体中。  
A. 结合  
B. 隐藏  
C. 封装  
D. 抽象
17. 瀑布模型的存在问题是( )  
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
18. 软件生命周期中所花费用最多的阶段是( )  
A. 详细设计  
B. 软件编码  
C. 需求分析  
D. 软件维护
19. 以下不是程序的三种基本控制结构是( )  
A. 过程  
B. 顺序  
C. 选择  
D. 循环
20. 架构模式描述了软件体系结构,下面属于架构模式的是( )  
A. 适配器(Adapter)模式  
B. 模型/视图/控制器(MVC)模式  
C. 抽象工厂模式(Abstract Factory)  
D. 桥(Bridge)模式

10. 面向数据流的设计方法（CSDN）

面向数据流的设计方法的目标是给出设计软件结构的一个系统化的途径。面向数据流的设计方法定义了一些不同的“映射”，利用这些映射可以把数据流图变成软件结构。



变换流的中心被称作变换中心。事务流的中心被称作事务中心。

B

11. 用例之间的关系（P242）

用例间关系	描述	符号	举例
拓展	向一个用例中添加一些动作后构成另一个用例，后者继承前者的一些行为。	«extend»	自动售货机系统中，“售货”是一个基本用例，如果顾客购买灌装饮料，售货功能顺利完成；但是如果顾客后买用纸杯装的散装饮料，则不能执行该用例提供的常规操作，需要拓展售货用例。
包含	一个用例包含另一个用例的全部过程。	«include»	自动售货机系统中，“供货”和“取货款”中，这两个用例的开始动作都是去掉保险并打开它，这两个用例可以把开始的动作抽象成“打开机器用例”。

C

## 12. 类与对象关系 (P150)

类是对象的集合，对象是类的实例。

C

## 13. 结构程序设计定义 (P92)

结构程序设计的经典定义如下：

如果一个程序的代码块仅仅通过顺序、选择和循环这3中控制结构进行连接，并且每个代码块只有一个入口和一个出口，则称这个程序是结构化的。

结构程序设计本质如下：

结构程序设计的本质并不是无GOTO语句的编程方法，而是一种使程序代码容易阅读、容易理解的编程方法。由于GOTO语句有时可以起到增强可读性的左右，所以只需限制GOTO语句的使用即可。

C

## 14. 模块的内聚性 (P73)



内聚类型	描述	内聚性 (正比于分数)	举例
功能内聚	一个模块内所有处理元素属于一个整体，完成一个单一的功能。	10	最高程度内聚。
顺序内聚	一个模块内的处理元素与同一个功能密切相关，并且这些处理必须顺序执行。	9	数据流图确定的模块划分。
通信内聚	一个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。	7	-
过程内聚	一个模块内的处理元素相关，而且必须以特定次序执行。	5	流程图确定的模块划分。
时间内聚	一个模块包含的任务必须在同一时间段内执行。	3	模块完成各种初始化工作。
逻辑内聚	一个模块完成的任务在逻辑上属于相同或相似的一类。	1	一个模块产生各种类型的全部输出。
偶然内聚	松散的关系	0	在写完一个程序之后，发现一组语句在两处或多处出现，于是把这些语句作为一个模块以节省内存。

C

## 15. 敏捷软件开发方法（CSDN）

敏捷软件开发方法如下：

1. 极限编程（XP）
2. Rational统一过程（RUP）
3. 微软过程（MSF）
4. 精益软件开发过程（Lean）
5. Scrum
6. DevOps（华为）

D

## 16. 对象 (P150)

对象定义：

面向对象方法学中的对象是由描述该对象属性的数据以及可以对这些数据施加的所有操作封装在一起构成的统一体。

对象特点：

1. 以数据为中心
2. 实现数据封装
3. 本质上具有并行性
4. 模块独立性好

C

## 17. 瀑布模型的问题 (P16)

瀑布模型几乎完全依赖与书面的规格说明，很有可能导致最终开发出的软件产品不能真正满足用户的需要，即不欢迎变化，缺少灵活性。

B

## 18. 软件生命周期的基本任务 (P11, P301)

软件生命周期由软件定义、软件开发和运行维护3个时期组成，每个时期又可进一步分成若干阶段：

软件定义时期：确定软件开发工程必须完成的总目标

1. 问题定义

“要解决的问题是什么？”

2. 可行性研究

“上一个阶段所确定的问题是否有行得通的解决方法？”

3. 需求分析

“目标系统必须做什么？”

软件开发时期：具体设计和实现前一个时期定义的软件

1. 概要设计

“怎样实现目标系统？”

2. 详细设计

“应该怎样具体的实现这个系统？”

3. 编码

写出正确的，容易理解和维护的程序模块。

4. 单元测试

## 5. 综合测试

通过各种类型的测试使软件达到预定的要求。

运行维护时期：使软件持久的满足用户的需要

### 1. 软件维护

通过各种必要的维护活动是系统持久的满足用户的需要。

通常，软件维护是软件生命周期中延续时间最长、工作量最大到的阶段。软件开发机构 60% 以上的精力都用在维护已有的软件产品，维护费用有时会高达开发费用的 4~5 倍。

D

## 19. 过程设计（P91）

过程设计是详细设计阶段应该完成的主要任务。其主要任务不是具体的编写程序而是要设计出程序的“蓝图”，以后程序员将根据这个蓝图编写实际的代码，并且代码应该尽可能的简明易懂。

顺序、选择、循环。

A

## 20. 软件架构风格与设计模式（教材P53、教材P133）

软件结构风格是描述某一特定应用领域中系统组织方式的惯用模式。

数据流风格：

1. 批处理序列
2. 管道与过滤器

调用/返回风格：

1. 层次结构
2. 正交软件结构
3. 客户机/服务器结构
4. 浏览器/服务器结构

独立构架风格：

1. 进程通信
2. 事件系统
3. MVC结构

虚拟机风格：

1. 解释器

数据中心风格：

1. 数据库系统
2. 超文本系统
3. 仓库/黑板系统

设计模式提供了相似的程序设计任务中经常出现的相同问题的解决方案。

创建模式：

1. 抽象工厂模式
2. 单例模式

结构模式：

1. 适配器模式
2. 桥模式
3. 装饰模式
4. 门面模式
5. 代理模式

行为模式：

1. 观察者模式
2. 策略模式
3. 状态模式

B