

# 2005 年软件工程试卷(B)

注意：请使用中文回答问题

## 一. 解释下列名词的含义（每个小题 2 分，共 20 分）

1. 过程 (Process)
2. 迭代式开发 (Iterative development)
3. 开发模式 (Development Paradigm)
4. 项目进度 (Project schedule)
5. 抛弃式原型 (Throw-away prototype)
6. 耦合 (Coupling)
7. 用例 (Use case)  
描述一个系统应当展示的部分功能
8. 能力缺陷 (capacity fault)  
统活动到达指定的极限时系统性能会变得不可接受
9. 容量测试 (volume test)  
验证系统处理巨量数据的能力
10. 软件可靠性 (Reliability)

## 二. 判断（填写 × 或 √）（每个小题 1 分，共 10 分）

1. ( ) 尽可能推迟程序的逻辑实现是按照瀑布模型开发软件的一条重要的指导思想。
2. ( ) 结构化程序设计方法能改善程序结构，提高程序的运行效率。
3. ( ) 一般来说，模块的内聚程度越高，他们之间的耦合程度也就越高。
4. ( ) 信息隐蔽原则禁止在模块外使用在模块接口说明的关于该模块的信息。
5. ( ) 面向对象方法更适合于软件重用的根本原因在于它是软部件唯一的合成技术。
6. ( ) 如果测试数据满足条件覆盖，则必然满足判定覆盖。
7. ( ) 桩模块的编写比驱动模块更困难。
8. ( ) 原型开发方法中的原型可以不抛弃，而逐步进化为最终的系统。
9. ( ) 开发人员和客户对软件质量因素的认可是完全一致的。
10. ( ) 软件运行正确，可见软件中没有缺陷 (fault)。

## 三. 填空（每个空 0.5 分，共 12 分）

1. 计算机软件不仅仅是程序，还应该有一整套 ( )。
2. ( ) 是指软件生存周期中的一系列相关活动，包括软件开发活动所需未酿成的任务序列和完成这些任务的工作步骤。

3. 确定需求过程一般分为四部分：( )、( )、( )、( )。
4. 在单元测试中，使用( )模拟被测试单元的调用和数据传递动作，( )模拟被测试模块的子程序。
5. 软件投入使用后，用户提出功能变动或增加，就要对软件进行( )。
6. 在类层次中，子类只继承一个父类的数据和方法，称为( )，子类继承了多个父类的数据和方法，称为( )。
7. 数据流图中，使用了四种基本符号，它们分别是( )、( )、( )、( )。
8. 引导测试又分为两种，分别是( )、( )。
9. 请列举出软件缺陷的三种类型：( )、( )、( )。
10. 请列举出性能测试的三种类型：( )、( )、( )。
11. 开发原型的目的是( )。

#### 四. 从供选择的答案中，选出正确的答案填入( )内。(每个小题 1 分，共 12 分)

1. 下列选项不属于瀑布模型的优点的是( )。  
A. 可迫使开发人员采用规范的方法  
B. 严格的规定了每个阶段必须提交的文档  
C. 要求每个阶段交出的所有产品都必须经过质量保证小组的仔细验证  
D. 支持后期的变动
2. 软件维护工程越来越受到重视。因为它的花费常常要占软件生命周期全部花费的( )。  
A. 10%~20%      B. 20%~40%      C. 60%~80%      D. 90%以上
3. 以下关于数据流图的说法错误的是( )  
A. 数据流图舍去了具体的物质，只剩下数据的流动、加工处理和存储  
B. 数据流图是用结构化分析建模的一种工具  
C. 传统的数据流图中主要由加工、数据源点/终点、数据流、控制流、数据存储组成  
D. 数据流图的绘制采用自上而下、逐层分解的方法
4. 需求分析阶段的研究对象是( )  
A. 系统分析员要求      B. 用户要求      C. 软硬件要求      D. 系统要求
5. 耦合度最高的是( )耦合。  
A. 环境      B. 内容      C. 控制      D. 数据
6. 软件测试是软件质量保证的重要手段，下述( )是软件测试的最基础的环节。

A. 功能测试      B. 单元测试      C. 结构测试      D. 确认测试

7. 软件测试方法中，黑盒测试法主要是用于测试（      ）

A. 结构合理性      B. 软件的外部功能  
C. 程序的正确性      D. 程序的内部逻辑

8. 软件测试的目的是 A[      ]。为了提高测试的效率，应该 B[      ]。使用白盒测试方法时，测试数据应该根据 C[      ]和指定的覆盖标准。一般来说，与设计测试数据无关的文档是 D[      ]，软件测试工作最好由 E[      ]来承担，以提高集成测试的效果。

A: (1)评价软件的质量      (2)发现软件的错误  
(3)找出软件中的所有错误      (4)证明软件是正确的

B: (1)随机的选取测试数据      (2)取一切可能的输入数据作为测试数据  
(3)在完成编码以后制订测试计划      (4)选择发现错误的的可能性大的数据作为测试数据

C: (1)程序的内部逻辑      (2)程序的复杂程度  
(3)使用说明书      (4)程序的功能

D: (1)需求规格说明书      (2)总体设计说明书  
(3)源程序      (4)项目开发计划

E: (1)该软件的设计人员      (2)该软件开发组的负责人  
(3)该软件的编程人员      (4)不属该软件开发组的软件设计人员

填入答案：(A:      B:      C:      D:      E:      )

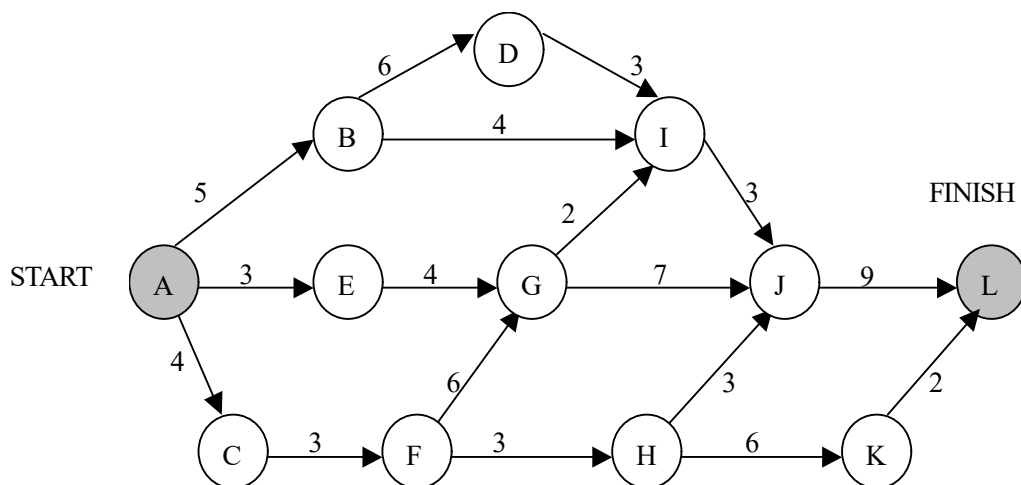
## 五. 简述题（共 24 分）

1. 说明错误、缺陷、失败的含义与联系。
2. 简述现代软件工程大致的阶段划分，以及各阶段的文档。
3. 请列举出用以衡量软件质量的三方面因素及其基本含义。
4. 简述获取需求时的三种类型。
5. 影响软件工程开发实践的关键要素是什么？
6. 举例说明模块间的各种内聚（Cohesion）关系。
7. 简述编程应考虑三方面原则。
8. 说明软件系统测试过程的主要步骤及含义。

## 六. 综合应用题（共 22 分）

1. (6 分) 下图是一个软件开发项目的活动图。其中各个节点表示活动的里程碑，各条边所标数字表示各项活动所需要的天数。对每项活动，列出其最早、最迟开始时间。然后，指出本活动图的关键路径。

注意：使用（<最早开始时间>,<最迟开始时间>）的形式直接标在图上。



关键路径是： ( )

2. (10 分) 高考录取统分子系统有如下功能：

- i. 计算标准分：根据考生原始分计算，得到标准分，存入考生分数文件。
- ii. 计算录取分数线：根据标准分，招生计划文件中的招生人数，计算录取线，存入录取线文件。
- iii. 考生信息查询：根据考生考号查询考生的考试成绩。

(1) 使用 DFD 图，画出数据加工流程（单层、多处理）。

(2) 使用 Use Case，画出用例图。

3. (6 分) 某系统要处理的类或数据如下：

公司：公司名称，主要产品，公司地址；

部门：部门名称；

员工：姓名，地址，身份证号码；

项目：项目名称；

产品：产品名称；

该公司有许多部门，部门名唯一地确定一个部门，每个部门生产多种产品，每种产品仅由一个部门生产。公司的员工分为经理和工人两类，每个工人可参加多个工程项目，每个项目需要多个员工，每位经理可以主持多个项目，每个项目仅有一人主持，同时一个经理至多管理一个部门。

画出该系统类和类关系的 UML 模型表示。

# 2007 年软件工程试卷(A 版)

注意：请使用中文回答问题

## 一. 解释下列名词的含义（每个小题 2 分，共 20 分）

1. 原型 (Prototype)
2. 增量式开发 (Incremental Development)
3. 演化型原型 (Evolutionary Prototype)
4. 面向对象 (Object Orientation)
5. 过载缺陷 (overload fault)
6. 强度测试 (stress tests)
7. 软件可维护性 (maintainability)
8. 由底向上测试 (Bottom-Up Test)
9. 极限编程(XP) (Extreme Programming)
10. 项目进度 (Schedule)

## 二. 判断（填写 × 或 √）（每个小题 1 分，共 10 分）

1. ( ) 软件测试只能证明程序有错误，不能证明程序没有错误。
2. ( ) 开发初始原型的目的是为了确定用户的基本需求。
3. ( ) 尽可能推迟程序的逻辑实现是按照瀑布模型开发软件的一条重要的指导思想。
4. ( ) 在软件开发的各个阶段进行过程中，增加人员肯定会对整个项目提前完成有好处。
5. ( ) 如果测试数据满足条件覆盖，则必然满足判定覆盖。
6. ( ) 一般来说，模块的内聚程度越高，他们之间的耦合程度也就越高。
7. ( ) 在程序调试时，改正该错误比找出错误的位置和性质更难。
8. ( ) 因为详细设计已经把模块的内部逻辑描述的相当清晰，所以编程时不必再采用统一的标准和约定。
9. ( ) 信息隐蔽原则禁止在模块外使用在模块接口说明的关于该模块的信息。
10. ( ) 开发人员和客户对软件质量因素的认可是完全一致的。

## 三. 填空（每个空 0.5 分，共 10 分）

1. 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 ( )。
2. 在单元测试中，使用 ( ) 模拟被测试单元的调用和数据传递动作，( ) 模拟被测试模块的子程序。
3. 请列举出用以衡量软件质量的三方面因素：( )、( )、( )。

4. 确定需求过程一般分为五部分：( )、( )、( )、( )、( )。
5. 软件投入使用后，用户提出功能变动或增加，就要对软件进行 ( )。
6. 引导测试又分为两种，分别是 ( )、( )。
7. 请列举出性能测试的三种类型：( )、( )、( )。
8. 螺旋模型中四重循环的结果分别是 ( )、( )、( )、( )。

#### 四. 从供选择的答案中，选出正确的答案填入 ( ) 内。(每个小题 1 分，共 13 分)

1. 软件工程的出现是由于 ( )。  
A. 软件危机的出现      B. 计算机硬件技术的发展  
C. 软件社会化的需求      D. 计算机软件技术的发展
2. 以下关于数据流图的说法错误的是 ( )  
A. 数据流图舍去了具体的物质，只剩下数据的流动、加工处理和存储  
B. 数据流图是用结构化分析建模的一种工具  
C. 传统的数据流图中主要由加工、数据源点/终点、数据流、控制流、数据存储组成  
D. 数据流图的绘制采用自上而下、逐层分解的方法
3. 需求分析阶段的研究对象是 ( )  
A. 系统分析员要求      B. 用户要求      C. 软硬件要求      D. 系统要求
4. 采用 UML 分析用户需求时，用例 UC1 可以出现在用例 UC2 出现的任何位置，那么 UC1 和 UC2 的关系是 ( )。  
A: include;      B: extends;      C: generalize;      D: call。
5. 传统的软件生命周期不包含 ( )。  
A. 问题定义      B. 性能优化      C. 测试      D. 实现
6. 系统技术可行性研究涉及的技术应该是 ( ) 技术。  
A. 现在已提出的      B. 现在在研究的  
C. 不一定可以获得的      D. 一定可以获得的
7. 软件测试是软件质量保证的重要手段，下述 ( ) 是软件测试的最基础的环节。  
A. 功能测试      B. 单元测试      C. 结构测试      D. 确认测试
8. 软件测试方法中，黑盒测试法主要是用于测试 ( )  
A. 结构合理性      B. 软件的外部功能

C. 程序的正确性

D. 程序的内部逻辑

9. 软件测试的目的是 A[ ]。为了提高测试的效率, 应该 B[ ]。使用白盒测试方法时, 测试数据应该根据 C[ ] 和指定的覆盖标准。一般来说, 与设计测试数据无关的文档是 D[ ], 软件测试工作最好由 E[ ] 来承担, 以提高集成测试的效果。

A: (1)评价软件的质量

(2)发现软件的错误

(3)找出软件中的所有错误

(4)证明软件是正确的

B: (1)随机的选取测试数据

(2)取一切可能的输入数据作为测试数据

(3)在完成编码以后制订测试计划 (4)选择发现错误的的可能性大的数据作为测试数据

C: (1)程序的内部逻辑 (2)程序的复杂程度

(3)使用说明书 (4)程序的功能

D: (1)需求规格说明书 (2)总体设计说明书

(3)源程序

(4)项目开发计划

E: (1)该软件的设计人员

(2)该软件开发组的负责人

(3)该软件的编程人员

(4)不属该软件开发组的软件设计人员

填入答案: (A: B: C: D: E: )

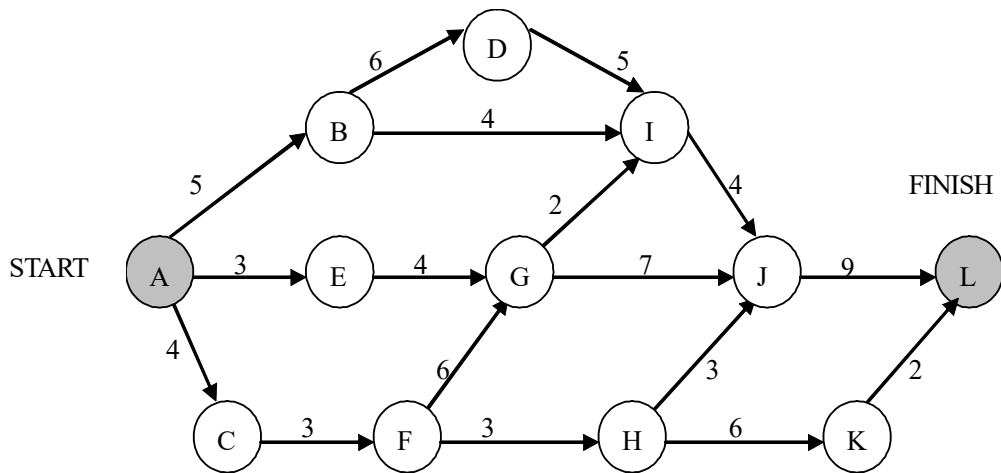
## 五. 简述题 (共 27 分)

1. 软件开发与程序设计有何不同?
2. 影响软件工程开发实践发生变化的关键要素是什么?
3. 试述 COCOMOII 模型的工作原理。
4. 考虑到需求的优先级, 举例说明获取需求时的需求基本分类。
5. 试述设计用户界面应考虑的问题
6. 举例说明模块间的各种内聚 (Cohesion) 关系。
7. 列出类图中各个类之间的各种基本关系
8. 传统测试与 OO 测试有何不同?
9. 说明系统测试过程的主要步骤及含义。

## 六. 综合应用题 (共 20 分)

1. (5 分) 下图是一个软件开发项目的活动图。其中各个节点表示活动的里程碑, 各条边所标数字表示各项活动所需要的天数。对每项活动, 列出其最早、最迟开始时间。然后, 指出本活动图的关键路径。

注意: 使用 (<最早开始时间>, <最迟开始时间>) 的形式直接标在图上。



关键路径是：( )

2、(10 分) 宠物商店 PetStore 是一个集客户购物、订单处理、销售统计等功能于一体的电子商务网站。以下是简要的功能需求：

客户分为匿名客户和注册客户，对于注册客户以会员方式管理，登记并管理其个人信息，根据其消费积分分为金牌会员、银牌会员、普通会员，以享受不同程度的优惠。

通过浏览器，匿名客户可以查询宠物；创建账户并登录后才能够使用购物车、创建订单、提交订单、通过信用卡支付等购物活动。每个客户可以同时拥有多个订单，但只能有一个购物车，订单基于购物车内的商品创建。

宠物商店的工作人员能够接受或拒绝客户提交的订单、处理订单、发订单给供应商、接受供应商返回的配送结果；还能够统计销售情况。

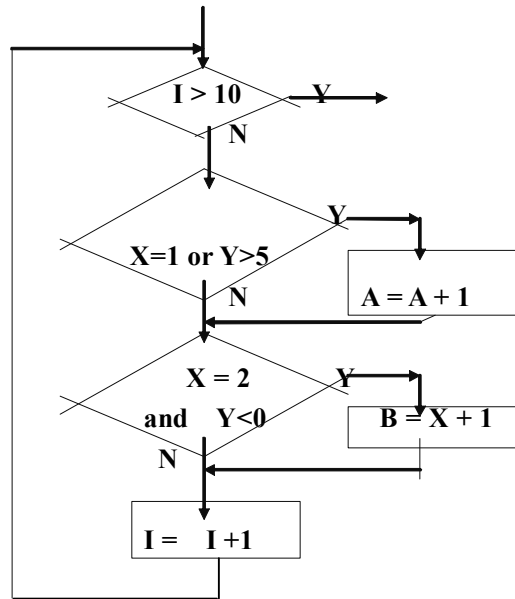
注册客户可以随时查询其订单状态，当订单尚未处理时，可以取消订单；订单一旦开始处理，不允许客户取消。

- (1) 使用 UML 用例图表达以上需求。
- (2) 使用 UML 类图描述此系统的对象模型。



3. (5 分) 针对如右程序流程图所示的程序结构：

假设程序中变量均为整形变量，为该程序设计测试用例，满足路径覆盖 (Path Testing) 要求。



# 2018 年

概念题

- 1.项目进度
- 2.重用
- 3.beta 测试
- 4.测试用例
- 5.生命周期
- 6.功能性需求
- 7.结对编程
- 8.迭代开发
- 9.文档缺陷
- 10.设计模式

判断题：

其余的和其他往年题一样

- 1.面向对象设计方法和面向对象分析方法产生结果类似，比如都可以用 UML 图表示（）
- 2.测试人员发现错误，并且好改，自己能改，就可以直接改（）

选择题：

其余基本和往年题差不多。

- 1.复审目的是为了什么 B:降低风险 C:提高质量

简答题：

- 1.叙述影响产品质量的因素
- 2.叙述依赖关系，并给出解释（类图或者代码例子）
- 3.题目给出类图，询问类图的设计是否合理。（比如：本来是组合关系，图中画的是继承）说明不合理的原因
- 4.一个老板认为迭代开发是，第一次迭代是需求分析，第二次迭代是系统设计，第三次。。。他的认为是否正确呢？说明理由
- 5.简述 COCOMO 模型原理

大题：

- 1.某客户信息管理系统中保存着两类客户的信息。

（1）个人客户。对于这类客户，系统保存了其客户标识（由系统生成）和基本信息（包括姓名、住宅电话和 E-mail）。

（2）集团客户。集团客户可以创建和管理自己的若干名联系人。对于这类客户，系统除了保

存其客户标识（由系统生成）之外，也保存了其联系人的信息。联系人的信息包括姓名、住宅电话、E-mail、办公电话以及职位。

画出类图，类关系和类的属性

2.信用卡有：开放 关闭 正常 透支 4 中状态，按条件转换（非常简单）。

给出某一天（年，月， 日），计算出它的下一天，取值范围为：

年：  $1000 \leq \text{year} \leq 3000$     月：  $1 \leq \text{month} \leq 12$     日：  $1 \leq \text{day} \leq 31$

如 1999 年 3 月 4 日的下一天是：1999 年 3 月 5 日

要求：

输入：三个参数（年，月， 日）

输出：如能正确计算，计算出它的下一天，

否则，输出相应的错误信息。

3.

按照年月份的规定输入，黑色测试。划分等价类并给出测试用例。

# 2012 年软件工程试卷(A 版 01 稿)

注意：请使用中文回答问题

## 一. 解释下列名词的含义（每个小题 1 分，共 10 分）

1. 统一过程 (UP-Unify Process)
2. 项目进度 (Project Schedule)
3. 重用 (Reuse)
4. 项目成本 (Project Cost)
5. 需求 (requirement)
6. 耦合 (Coupling)
7. 关联 (Association)
8. 极限编程 (XP-Extreme Programming)
9. 驱动模块 (Component Driver)
10.  $\beta$  测试 ( $\beta$  / Beta Test)

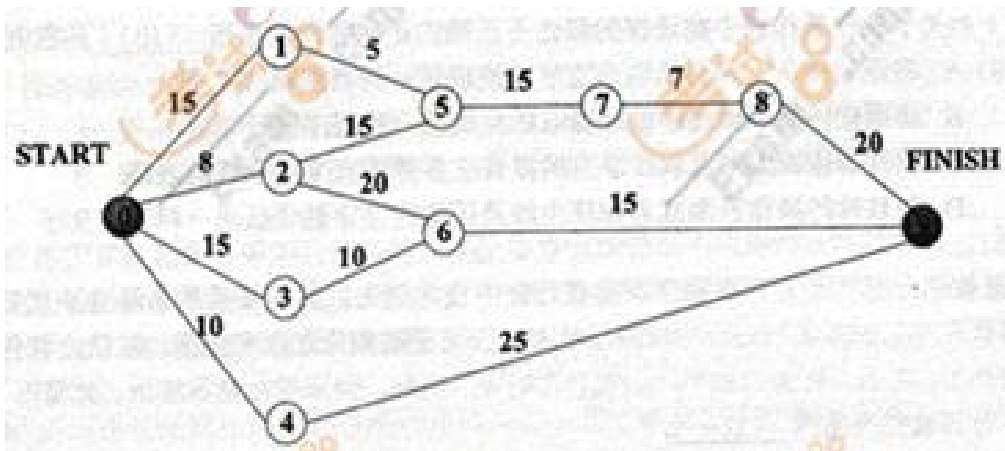
## 二. 判断（填写 $\times$ 或 $\checkmark$ ）（每个小题 1 分，共 10 分）

1. ( ) 原型化可以改善对需求和设计的理解，因此选择原型化过程可以降低项目风险。
2. ( ) 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。
3. ( ) 软件过载缺陷就是当运行程序时，软件内部定长的数据结构被溢出，系统任务无法完成。
4. ( ) 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。
5. ( ) RUP (Rational Unified Process: 统一软件过程) 本质上是轻量级的软件过程规范。
6. ( ) 如果测试数据满足了条件覆盖，则必然满足判定覆盖。
7. ( ) 每一个故障(缺陷)都会对应于一个失效(失败)。
8. ( ) 设计成功的系统不需维护, 维护意味着系统存在问题。
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. ( ) 开发人员和客户对软件质量因素的认可是一致的。

## 三. 从供选择的答案中，选出正确的答案填入\_\_\_\_\_内。（每个小

题或题空为 1 分，共 25 分)

- 1、以下关于软件测试的叙述中，正确的是\_\_\_\_\_。
  - A. 软件测试不仅能表明软件中存在错误，也能说明软件中不存在错误。
  - B. 软件测试活动应该从编码开始。
  - C. 一个成功的测试能发现至今未发现的错误。
  - D. 在一个被测程序段中，若已发现的错误越多，则残存的错误越少。
- 2、不属于黑盒测试技术的是\_\_\_\_\_。
  - A. 错误猜测
  - B. 逻辑覆盖
  - C. 边界值分析
  - D. 等价类划分
- 3、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\_\_\_\_\_来表述“谁做”。
  - A. 角色
  - B. 活动
  - C. 制品
  - D. 工作流
- 4、\_\_\_\_\_ (1) 是把对象的属性和服务结合成一个独立的系统单元，并尽可能隐藏对象的内部细节；\_\_\_\_\_ (2) 是指子类可以自动拥有父类的全部属性和服务；\_\_\_\_\_ (3) 是对象发出的服务请求，一般包含提供服务的对象标识、服务标识、输入信息和应答信息等。
  - (1) A. 继承 B. 多态 C. 消息 D. 封装
  - (2) A. 继承 B. 多态 C. 消息 D. 封装
  - (3) A. 继承 B. 多态 C. 消息 D. 封装
- 5、风险预测从两个方面评估风险，即风险发生的可能性以及\_\_\_\_\_。
  - A. 风险产生的原因。
  - B. 风险监控技术。
  - C. 风险能否消除。
  - D. 风险发生所产生的后果。
- 6、UML 中关联的多重度是指\_\_\_\_\_。
  - A. 一个类中被另一个类调用的方法个数。
  - B. 一个类的某个方法被另一个类调用的次数。
  - C. 一个类的实例能够与另一个类的多少个实例相关联。
  - D. 两个类所具有的相同的方法和属性。
- 7、下图是一个软件项目的活动图，里程碑\_\_\_\_\_在关键路径上。



A.1                  B.2                  C.3                  D.4

8、UML 是一种面向对象的统一建模语言，它包含九种图。其中用例图中的外部 actor 指\_\_\_\_(1)\_\_\_\_，用例可以用\_\_\_\_(2)\_\_\_\_图来描述。协作图描述了协作的\_\_\_\_(3)\_\_\_\_之间的交互和联结。

- (1) A 人员      B 单位      C 人员或单位      D 人员或外部系统  
 (2) A 类      B 状态      C 活动      D 协作  
 (3) A 对象      B 类      C 用例      D 状态

9、按测试数据执行程序逻辑程度判断，判定覆盖\_\_\_\_条件覆盖。

- A. 强于                  B. 弱于                  C. 等价于                  D. 不等价于

10、包含 8 个成员的开发小组的沟通路径最多有\_\_\_\_条。

- A. 28      B. 32      C. 56      D. 64

11、模块 A 直接访问模块 B 的内部数据，则模块 A 和模块 B 的耦合类型为\_\_\_\_\_。

- A. 数据耦合      B. 标记耦合      C. 公共耦合      D. 内容耦合

12、下列关于项目估算的叙述不正确的是\_\_\_\_\_。

- A. 专家判断方法受到专家经验和主观性影响  
 B. 启发式方法（如 COCOMO 模型）的参数难以确定  
 C. COCOMOII 采用多次估算的策略，估算结果将越来越准确  
 D. 结合上述三种方法可以得到精确的估算结果

14、为了有效地捕获系统需求，应采用\_\_\_\_\_。

- A. 瀑布模型  
 B. V 模型  
 C. 原型模型  
 D. 螺旋模型

15、UML 设计视图包含了类、接口和协作，其中，设计视图的静态方面由\_\_\_\_(1)\_\_\_\_和\_\_\_\_(2)\_\_\_\_表现，动态方面由交互图、\_\_\_\_(3)\_\_\_\_表现。

- (1) A 类图      B 状态图      C 活动图      D 序列图  
 (2) A 交互图      B 对象图      C 通信图      D 定时图

(3) A 状态图和类图      B 活动图和类图      C 状态图和对象图      D 状态图和活动图

16、软件\_\_\_\_\_是指一个系统在给定的时间间隔内和给定条件下无失效运行的概率。

- A. 可靠性
- B. 可用性
- C. 可维护性
- D. 可伸缩性

17、高质量文档所应具备的特性中，不包括\_\_\_\_\_。

- A. 针对性，文档编制应考虑读者对象群
- B. 精确性，文档的行文应十分确切，不能出现多义性描述
- C. 完整性，任何文档都应该是完整的、独立的、应该自成体系
- D. 无重复性，统一软件系统的几个文档之间应该没有相同内容，若存在相同内容，则可以用“见\*\*文档\*\*节”方式引用

18、在面向对象软件开发过程中，采用设计模式\_\_\_\_\_

- A. 复用成功的设计
- B. 保证程序运行速度达到最优
- C. 减少设计中创建类的个数
- D. 允许在非面向对象程序设计语言中使用面向对象的概念

19、下列关于风险的叙述不正确的是：风险是指\_\_\_\_\_。

- A. 可能发生的事件
- B. 一定会发生的事件
- C. 会带来损失的事件
- D. 可能对其进行干预，以减少损失的事件

20、黑盒测试不能发现\_\_\_\_\_。

- A. 不正确或遗漏的功能。
- B. 初始化或终止性错误。
- C. 内部数据结构不合理。
- D. 性能不满足要求。

## 四. 简述题（共 30 分）

1. 请写出下列术语的英文全称或缩写：

软件需求规格说明书，设计模式，验收测试。

2. 影响软件工程开发实践的关键要素是什么？

3. 螺旋模型四个象限的基本任务和四重循环的大致结果是什么？

4. 考虑到需求的优先级，举例说明获取需求时的需求基本分类。

1. 传统"瀑布模型"的主要缺陷是什么？试说明造成缺陷的主要原因。

- 3、对于需求文档，如何进行测试？（哪些方面？ 什么原则？）
- 7. 举例说明模块间的各种内聚（Cohesion）关系。
- 6. 功能测试的基本指导原则是什么？
- 5. 试述 COCOMOII 模型的工作原理。
- 8. 试述设计用户界面应考虑的问题

五. 综合应用题（共 25 分）

- 1、（12 分）某客户信息管理系统中保存着两类客户的信息。
- （1）个人客户。对于这类客户，系统保存了其客户标识（由系统生成）和基本信息（包括姓名、住宅电话和 E-mail）。
  - （2）集团客户。集团客户可以创建和管理自己的若干名联系人。对于这类客户，系统除了保存其客户标识（由系统生成）之外，也保存了其联系人的信息。联系人的信息包括姓名、住宅电话、E-mail、办公电话以及职位。

该系统除了可以保存客户信息之外，还具有以下功能：

- （1）向系统中添加客户（addCustomer）；
- （2）根据给定的客户标识，在系统中查找该客户（getCustomer）；
- （3）根据给定的客户标识，从系统中删除该客户（removeCustomer）；
- （4）创建新的联系人（addContact）；
- （5）在系统中查找指定的联系人（getContact）；
- （6）从系统中删除指定的联系人（removeContact）。

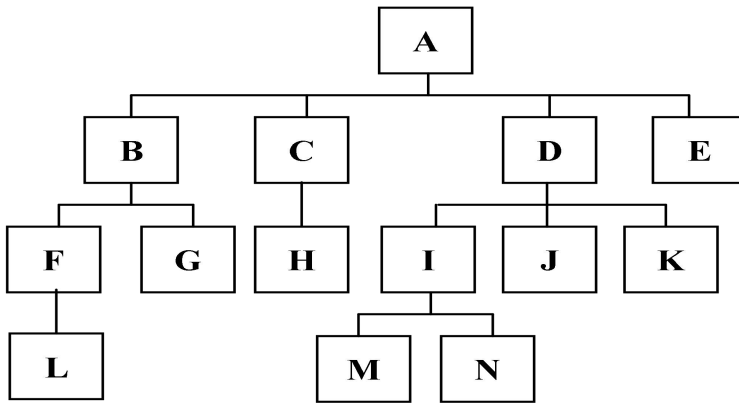
该系统采用面向对象方法进行开发。在面向对象分析阶段，根据上述描述，得到如下表所示的类。

类 名	说 明
CustomerInformationSystem	客户信息管理系统
IndividualCustomer	个人客户
InstitutionalCustomer	集团客户
Contact	联系人

描述该客户信息管理系统的 UML 类图如下图所示。







3、（7 分）下面是一段求最大值的程序，其中 datalist 是数据表，n 是 datalist 的长度。

```
int GetMax(int n, int datalist[ ])
{
    int k=0;
    for ( int j=1; j<n; j++ )
        if ( datalist[j] > datalist[k] ) k=j;
    return k;
}
```

- （1） 画出该程序的控制流图。
- （2） 设计路径覆盖的测试用例。

# 2009 年软件工程试卷(A 版定稿 02)

注意：请使用中文回答问题

## 一. 解释下列名词的含义（每个小题 1 分，共 15 分）

1. 抽象 (Abstraction)
2. 原型 (Prototype)
3. 分阶段开发模型 (Phased Development)
4. 功能性需求 (Functional Requirement)
5. 公共耦合 (Common Coupling)
6. 用例 (Use Case)
7. 组合 (Composition)
8. 派对编程 (Pair Programming)
9. 路径测试 (Path Testing)
10. 驱动模块 (Component Driver)
11. 基线 (Base Line)
12. 软件可靠性 (Software Reliability)
13. 项目进度 (Project Schedule)
14. 能力缺陷 (Capacity fault)
15. 抛弃式原型 (Throw-away prototype)

## 二. 判断（填写 × 或 √）（每个小题 1 分，共 15 分）

1. ( ) 软件生存周期是从软件开始开发到开发结束的整个时期。
2. ( ) 当前管理信息系统建设的主要问题是技术问题。
3. ( ) 尽可能推迟程序的逻辑实现是按照瀑布模型开发软件的一条重要的指导思想。
4. ( ) 开发人员和客户对软件质量因素的认可是完全一致的。
5. ( ) 信息隐蔽原则禁止在模块外使用在模块接口说明的关于该模块的信息。
6. ( ) 在划分模块时, 模块的内聚性越高越好, 而模块的耦合度越低越好。
7. ( ) 如果测试数据满足条件覆盖, 则必然满足判定覆盖。
8. ( ) 面向对象方法更适合于软件重用的根本原因在于它是软部件唯一的合成技术。
9. ( ) 在系统测试中发现的错误越多, 说明仍存在错误的可能性越小。
10. ( ) 有时为了测试的方便, 而可以局部地修改软件系统。
11. ( ) 设计成功的系统不需维护, 维护意味着系统存在问题。
12. ( ) 由于功能测试是一次测试系统的一个功能, 有时若需要, 功能测试可以早于整个系统的集成来进行。
13. ( ) 结构化程序设计方法能改善程序结构, 提高程序的运行效率。

14. ( ) 系统设计是程序设计的先导和前提条件。
15. ( ) 类的行为可以基于用例图进行测试。

三. 从供选择的答案中, 选出正确的答案填入 ( ) 内。(每个小题或题为 1 分, 共 20 分)

1. 下列选项不属于瀑布模型的优点的是 \_\_\_\_\_。

- A. 可迫使开发人员采用规范的方法  
B. 严格的规定了每个阶段必须提交的文档  
C. 要求每个阶段交出的所有产品都必须经过质量保证小组的仔细验证  
D. 支持后期的变动

2. 在下列说法中, \_\_\_\_\_ 是造成软件危机的主要原因。

- ① 用户使用不当                      ② 软件本身特点                      ③ 硬件不可靠  
④ 对软件的错误认识                  ⑤ 缺乏好的开发方法和手段              ⑥ 开发效率低

供选择的答案:    A: ①③⑥                      B: ①②④                      C: ③⑤⑥                      D: ②⑤⑥

3. 需求分析所要做的工作是深入描述软件的功能和性能, 确定软件设计的限制和软件同其他系统元素的接口细节, 定义软件的其他有效性需求, 细化软件要处理的数据域。需求可以分为功能需求、非功能需求、设计约束三大类, 其中 \_\_\_\_\_ 属于设计约束。

供选择的答案:

- A: 系统必须完成自动统计功能。              B: 系统要采用国有自主知识产权的中间件系统。  
C: 每笔查询都必须在 10 秒内响应。              D: 系统必须能够自动记录用户的操作过程。

4. 某软件在应用初期运行在 Windows XP 环境中, 现因某种原因, 该软件需要在 UNIX 环境中运行, 而且必须完成相同的功能。为适应这个要求, 软件本身需要进行修改, 而所需修改的工作量取决于该软件的 \_\_\_\_\_。

供选择的答案:

- A: 可扩充性。                      B: 可靠性。                      C: 复用性。                      D: 可移植性。

5. 软件生命周期一般都被划分为若干个独立的阶段, 其中占用精力和费用最多的阶段往往是 \_\_\_\_\_。

- A. 运行和维护阶段              B. 设计阶段              C. 代码实现阶段              D. 测试阶段

6. 两个小组独立地测试同一个程序, 第一组发现 25 个错误, 第二组发现 30 个错误, 在两个小组发现的错误中有 15 个是共同的, 那么可以估计程序中的错误总数是 \_\_\_\_\_ 个。

供选择的答案:

- A. 25                      B. 30                      C. 50                      D. 60

7. 系统测试是将软件系统与硬件、外设和网络等其他因素结合, 对整个软件系统进行测试。 \_\_\_\_\_ 不是系统测试的内容。

供选择的答案:

A: 路径测试。 B: 可靠性测试。 C: 安装测试。 D: 安全测试。

8. 内聚是一种指标,表示一个模块\_\_\_\_\_。

供选择的答案:

A: 代码优化的程度。 B: 代码功能的集中程度。  
C: 完成任务的及时程度。 D: 为了与其他模块连接所要完成的工作量。

9. 下列叙述中,与提高软件可移植性相关的是\_\_\_\_\_。

供选择的答案:

A: 选择时间效率高的算法。  
B: 尽可能减少注释。  
C: 选择空间效率高的算法。  
D: 尽量用高级语言编写系统中对效率要求不高的部分。

10. 在 UML 提供的图中,可以采用\_\_\_(1)\_\_\_对逻辑数据库模式建模; \_\_\_(2)\_\_\_用于接口、类和协作的行为建模,并强调对象行为的时间顺序; \_\_\_(3)\_\_\_用于系统的功能建模,并强调对象的控制流。

供选择的答案:

① A. 用例图 B. 构件图 C. 活动图 D. 类图  
② A. 协作图 B. 状态图 C. 序列图 D. 对象图  
③ A. 状态图 B. 用例图 C. 活动图 D. 类图

填入答案 ( (1): \_\_\_\_\_ (2): \_\_\_\_\_ (3): \_\_\_\_\_ )

11. 假定一个图书管理系统即将被开发,判断开发过程中出现的以下问题是属于哪种类型的bug,请做出选择:

A. 错误 (errors) B. 缺陷 (faults) C. 失败 (failures)

(1) 在计算超期罚款的代码中, final\_total 变量未被初始化。 ( )

(2) 当管理员试图向库存中增加一本新书时,系统崩溃了。 ( )

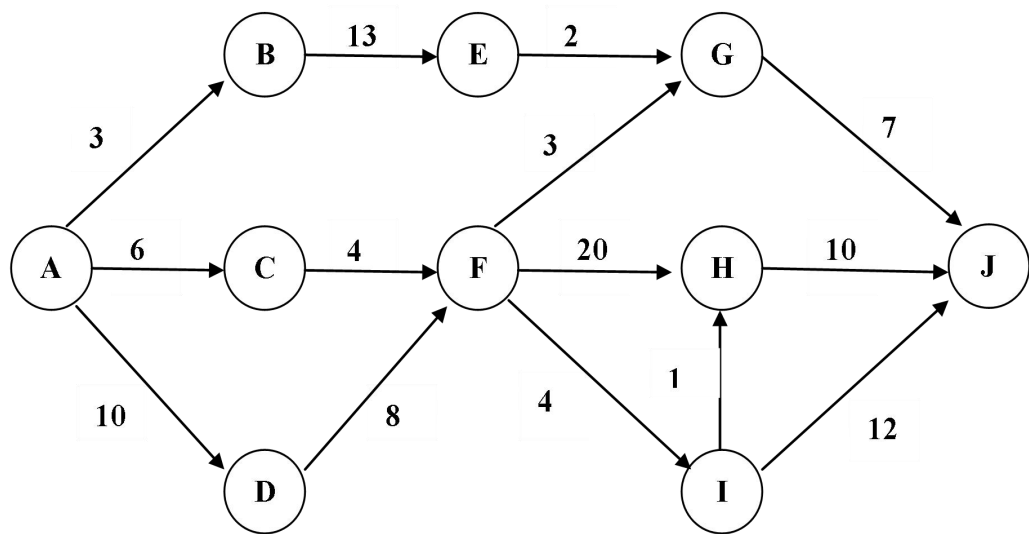
(3) 需求分析人员不知道读者在还书的时候不需要出示借书卡。 ( )

(4) 在需求文档中,超期罚款为每天 0.25 元,但最高不超过 15 元。在计算超期罚款的代码中并没有检查这个最大值。 ( )

(5) 每天晚上 11 点,系统需要备份当天的交易,但是这个星期二晚上没有备份。( )

填入答案 ((1): \_\_\_\_\_ (2): \_\_\_\_\_ (3): \_\_\_\_\_ (4) \_\_\_\_\_ (5) \_\_\_\_\_)

12. 以下是某个项目的活动图:



(1). 以下哪一个是里程碑 A 到 J 的一条关键路径?

- a. ACFHJ      b. ACFIHJ      c. ABEGHJ      d. ADFHJ

(2). 从里程碑 C 开始的活动的空闲时间是

- a. 7      b. 8      c. 15      d. 20

(3). 开始于里程碑 E 的活动的最晚开始时间是

- a. 10      b. 18      c. 25      d. 40

填入答案 ((1): \_\_\_\_\_ (2): \_\_\_\_\_ (3): \_\_\_\_\_ )

#### 四. 简述题 (共 30 分)

1. 请列举出用以衡量软件质量的三方面因素及其基本含义。
2. 影响软件工程开发实践的关键要素是什么?
3. 使用自然语言书写的软件需求有何问题, 试讨论之。
4. 请介绍极限编程 (XP) 的含义、特征、目的等。
5. 试述 COCOMOII 模型的工作原理。
6. 功能测试的基本指导原则是什么?
7. 试简单论述 OO 测试的困难之处。
8. 请按照自己的理解, 阐述在软件工程项目开发中, 用例驱动的含义。
9. 有一种说法, “在软件开发团队中, 书写程序有时是一件令人气馁的任务”, 试解释之。
10. 试述设计用户界面应考虑的问题

五. 综合应用题（共 20 分）

1.（6 分）阅读下列说明及图示，回答问题[1]和[2]，将解答填入本题后面的对应栏内。

[说明]

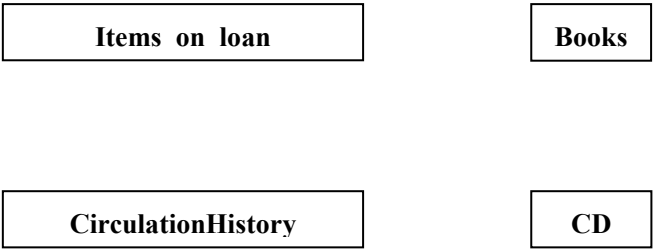
某公司的主要业务是出租图书和唱碟。由于业务需求，该公司委托软件开发公司 A 开发一套信息管理系统，该系统将记录所有的图书信息、唱碟信息、用户信息、用户租借信息等。A 公司决定采用面向对象的分析和设计方法开发此系统。图[1]所示为某类图书或唱碟被借阅时应记录的信息，图[2]描述了系统定义的两个类 Book 和 CD，分别表示图书和唱碟的信息。

[问题 1]

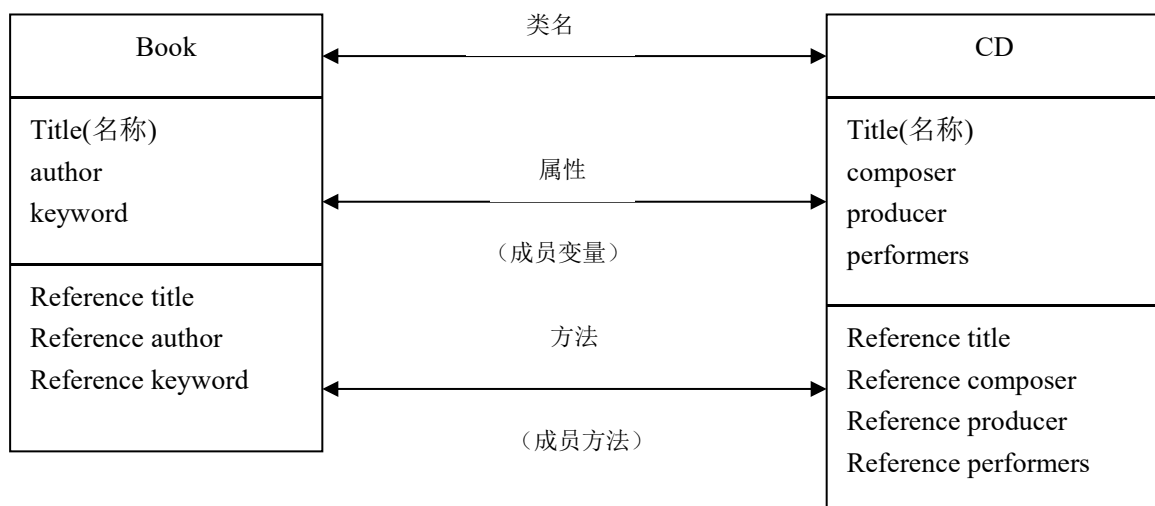
经过进一步分析，设计人员决定定义一个类 Items\_on\_loan，以表示类 Book 和 CD 的共有属性和方法。请采用图[2]中属性和方法的名称给出类 Items\_on\_loan 应该具有的属性和方法。（注意：不同名称的属性和方法表示不同的含义，如 CD 中的 composer 与 Book 中的 author 无任何关系。）

[问题 2]

为了记录每种图书或唱碟的历史记录，引入类 CirculationHistory，类中存储的信息是图[1]中所表示的内容。请采用 UML 表示法将下列 4 个类间的关系表示出来。



图书/唱碟 名称: _____																	
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <th style="padding: 5px;">借出时间</th> <th style="padding: 5px;">归还时间</th> <th style="padding: 5px;">用户</th> </tr> <tr><td style="height: 20px;"></td><td></td><td></td></tr> <tr><td style="height: 20px;"></td><td></td><td></td></tr> <tr><td style="height: 20px;"></td><td></td><td></td></tr> <tr><td style="height: 20px;"></td><td></td><td></td></tr> </table>			借出时间	归还时间	用户												
借出时间	归还时间	用户															



## 2. (8 分) 认真阅读以下关于一个茶庄系统的需求定义：

在一个茶庄(TeaStall)所使用的系统里,有各种风味的名茶(Tea),按照茶的风味(Flavor)不同,可分为“西湖龙井”,“洞庭碧螺春”,“黄山毛峰”,“安溪铁观音”,“信阳毛尖”,“庐山云雾”等,而且不同风味的茶其冲泡方法也不同。

茶庄的菜单(TeaMenu)中包含了茶庄中可供顾客点的各种风味的茶。茶庄的沏茶员(InfuseMan)可以根据侍者(Waitor)所下的单,冲泡各种风味的茶,如果订单中风味的茶已沏好,就直接为侍者提供茶;如果还没沏好则先冲泡再给侍者提供。沏茶员负责维护茶壶柜(teapotCabinet)和冲泡好的风味数。闭店时,沏茶员可以统计一天冲泡的风味数。

每个茶客(TeaClient)一般都有自己最喜欢的风味(favoriteFlavor),茶客每天根据菜单点适合自己心情的茶。



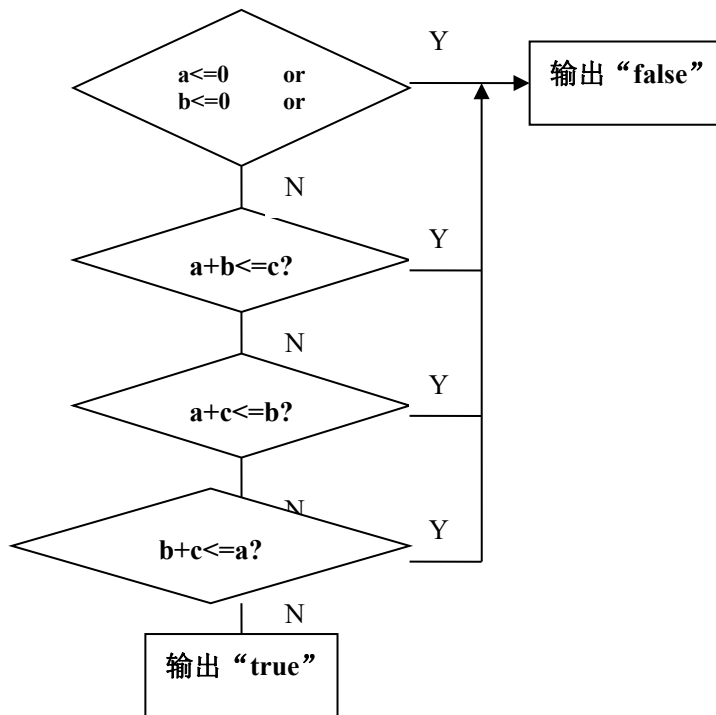
侍者 (Waitor) 担负着茶客和沏茶员之间的桥梁。他询问茶客最喜欢的风味, 并根据顾客的要求下单给沏茶员, 从沏茶员那里接过茶杯放于茶盘上。侍者将茶盘上的茶杯端给茶客。侍者会在一天中, 统计自己招待过的订单数, 并在闭店时, 统计服务的订单数。

茶庄 (TeaStall) 包含了以上所有, 但是它有容量限制, 即可容纳的最多茶客数, 每天开店时, 派出一位沏茶员和一位侍者, 并接待最多数量的茶客。每天闭店后, 会统计整个茶庄当天的订单数和沏茶风味数。

- (1) 画出用例图;
- (2) 画出类及类关系图。

3. (6 分) 组件 `Boolean isTriangle(int a, int b, int c)` 用来判断边长分别为 `a, b, c` 的图形是否为三角形。

- (1) 使用黑盒测试法划分等价类, 然后选择测试用例。
- (2) 以下为此组件的逻辑结构:

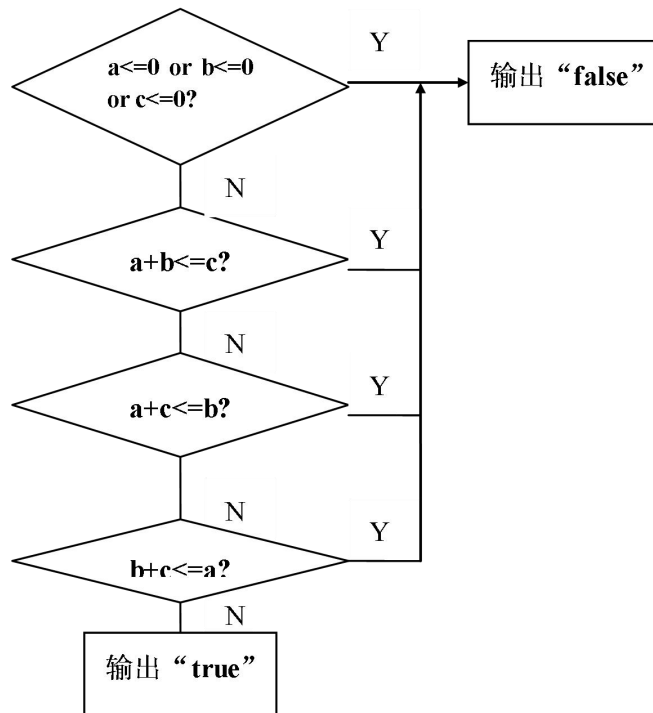


假设程序中变量均为整型变量, 请设计测试用例, 满足“路径覆盖 (Path Testing)”的要求。

3. (6分) 组件 **Boolean isTriangle(int a, int b, int c)**用来判断边长分别为 **a,b,c** 的图形是否为三角形。

(1) 使用黑盒测试法划分等价类，然后选择测试用例。

(2) 以下为此组件的逻辑结构：



假设程序中变量均为整型，请设计测试用例，满足“路径覆盖 (Path Testing)”的要求。

# 2011 回忆版

## 一、名词解释

软件工程、重用、文档缺陷、分阶段模型、派对编程、 $\alpha$  测试、组合、RUP、还有两个

## 二、判断

应该只有三个是往年的原题，其他的都没见过

## 三、选择

应该有 14 个题（不过与往年的题一模一样的也就三四个吧）

有考察对错误、缺陷、失败的理解，求关键路径，考风险，某个测试，总之范围比较广，需要平时的积累。

## 四、简答题

1、写出分阶段模型、需求规格说明书、设计模式、验收测试、软件配置管理的英文单词

2、如何衡量软件质量

3、主程序员负责制的优缺点

4、写出需求工程中获取需求的步骤吧（我的表述不太准确，大概就是这个意思）

我记得主要答案是原始需求的获取、问题分析、规格草稿说明、需求核准、输出正式的需求规格说明书吧，大概是这样。你还需要对每一步有个简短的解释。

5、写出所有的内聚并解释其含义

6、功能测试的原则

## 五、综合题

1、类似填用例图，给的是一个旅游景点的管理系统的描述（不难）

2、给一小段程序（其实你可以根据这段程序画出一个流程图，程序非常简单），然后让你设计判定覆盖和条件覆盖的测试用例

3、给你一段描述（很短），让你画类图（大概是一辆红色金属漆的轿车，它是四门车，前后门，前后挡风玻璃、轮胎、后视镜等，总之就是告诉你轿车的各个构件，让你画类图）

# 2012 年软件工程试卷(A 版 01 稿)

注意：请使用中文回答问题

## 一. 解释下列名词的含义（每个小题 1 分，共 10 分）

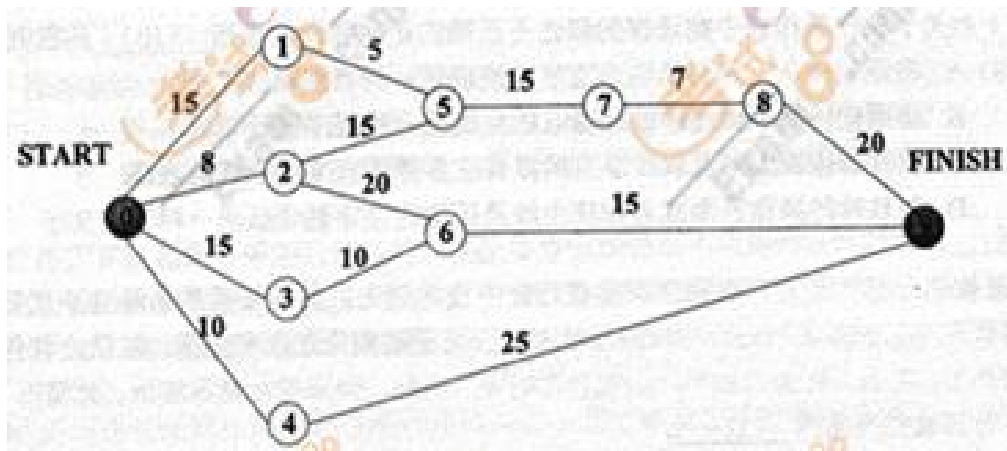
1. 统一过程 (UP-Unify Process)
2. 项目进度 (Project Schedule)
3. 重用 (Reuse)
4. 项目成本 (Project Cost)
5. 需求 (requirement)
6. 耦合 (Coupling)
7. 关联 (Association)
8. 极限编程 (XP-Extreme Programming)
9. 驱动模块 (Component Driver)
10.  $\beta$  测试 ( $\beta$  / Beta Test)

## 二. 判断（填写 $\times$ 或 $\checkmark$ ）（每个小题 1 分，共 10 分）

1. ( ) 原型化可以改善对需求和设计的理解，因此选择原型化过程可以降低项目风险。
2. ( ) 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。
3. ( ) 软件过载缺陷就是当运行程序时，软件内部定长的数据结构被溢出，系统任务无法完成。
4. ( ) 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。
5. ( ) RUP (Rational Unified Process: 统一软件过程) 本质上是轻量级的软件过程规范。
6. ( ) 如果测试数据满足了条件覆盖，则必然满足判定覆盖。
7. ( ) 每一个故障(缺陷)都会对应于一个失效(失败)。
8. ( ) 设计成功的系统不需维护, 维护意味着系统存在问题。
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. ( ) 开发人员和客户对软件质量因素的认可是一致的。

## 三. 从供选择的答案中，选出正确的答案填入\_\_\_\_\_内。（每个小题或题空为 1 分，共 25 分）

- 1、以下关于软件测试的叙述中，正确的是\_\_\_\_\_。
  - A. 软件测试不仅能表明软件中存在错误，也能说明软件中不存在错误。
  - B. 软件测试活动应该从编码开始。
  - C. 一个成功的测试能发现至今未发现的错误。
  - D. 在一个被测程序段中，若已发现的错误越多，则残存的错误越少。
- 2、不属于黑盒测试技术的是\_\_\_\_\_。
  - A. 错误猜测
  - B. 逻辑覆盖
  - C. 边界值分析
  - D. 等价类划分
- 3、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\_\_\_\_\_来表述“谁做”。
  - A. 角色
  - B. 活动
  - C. 制品
  - D. workflow
- 4、\_\_\_\_\_(1)\_\_\_\_\_是把对象的属性和服务结合成一个独立的系统单元，并尽可能隐藏对象的内部细节；\_\_\_\_\_(2)\_\_\_\_\_是指子类可以自动拥有父类的全部属性和服务；\_\_\_\_\_(3)\_\_\_\_\_是对象发出的服务请求，一般包含提供服务的对象标识、服务标识、输入信息和应答信息等。
  - (1)
    - A. 继承
    - B. 多态
    - C. 消息
    - D. 封装
  - (2)
    - A. 继承
    - B. 多态
    - C. 消息
    - D. 封装
  - (3)
    - A. 继承
    - B. 多态
    - C. 消息
    - D. 封装
- 5、风险预测从两个方面评估风险，即风险发生的可能性以及\_\_\_\_\_。
  - A. 风险产生的原因。
  - B. 风险监控技术。
  - C. 风险能否消除。
  - D. 风险发生所产生的后果。
- 6、UML 中关联的多重度是指\_\_\_\_\_。
  - A. 一个类中被另一个类调用的方法个数。
  - B. 一个类的某个方法被另一个类调用的次数。
  - C. 一个类的实例能够与另一个类的多少个实例相关联。
  - D. 两个类所具有的相同的方法和属性。
- 7、下图是一个软件项目的活动图，里程碑\_\_\_\_\_在关键路径上。



A.1

B.2

C.3

D.4

8、UML 是一种面向对象的统一建模语言，它包含九种图。其中用例图中的外部 actor 指\_\_\_\_(1)\_\_\_\_，用例可以用\_\_\_\_(2)\_\_\_\_图来描述。协作图描述了协作的\_\_\_\_(3)\_\_\_\_之间的交互和联结。

(1) A 人员      B 单位      C 人员或单位      D 人员或外部系统

(2) A 类      B 状态      C 活动      D 协作

(3) A 对象      B 类      C 用例      D 状态

9、按测试数据执行程序逻辑程度判断，判定覆盖\_\_\_\_条件覆盖。

A. 强于      B. 弱于      C. 等价于      D. 不等价于

10、包含 8 个成员的开发小组的沟通路径最多有\_\_\_\_条。

A. 28      B. 32      C. 56      D. 64

11、模块 A 直接访问模块 B 的内部数据，则模块 A 和模块 B 的耦合类型为\_\_\_\_。

A. 数据耦合      B. 标记耦合      C. 公共耦合      D. 内容耦合

12、下列关于项目估算的叙述不正确的是\_\_\_\_。

- A. 专家判断方法受到专家经验和主观性影响
- B. 启发式方法（如 COCOMO 模型）的参数难以确定
- C. COCOMOII 采用多次估算的策略，估算结果将越来越准确
- D. 结合上述三种方法可以得到精确的估算结果

14、为了有效地捕获系统需求，应采用\_\_\_\_。

- A. 瀑布模型
- B. V 模型
- C. 原型模型
- D. 螺旋模型

15、UML 设计视图包含了类、接口和协作，其中，设计视图的静态方面由\_\_\_\_(1)\_\_\_\_和\_\_\_\_(2)\_\_\_\_表现，动态方面由交互图、\_\_\_\_(3)\_\_\_\_表现。

(1) A 类图      B 状态图      C 活动图      D 序列图

(2) A 交互图      B 对象图      C 通信图      D 定时图

(3) A 状态图和类图      B 活动图和类图      C 状态图和对象图      D 状态图和活动图

16、软件\_\_\_\_是指一个系统在给定的时间间隔内和给定条件下无失效运行的概率。

- A. 可靠性
- B. 可用性
- C. 可维护性
- D. 可伸缩性

17、高质量文档所应具备的特性中，不包括\_\_\_\_。

- A. 针对性，文档编制应考虑读者对象群
- B. 精确性，文档的行文应十分确切，不能出现多义性描述
- C. 完整性，任何文档都应该是完整的、独立的、应该自成体系
- D. 无重复性，统一软件系统的几个文档之间应该没有相同内容，若存在相同内容，则可

以用“见\*\*文档\*\*节”方式引用

- 18、在面向对象软件开发过程中，采用设计模式\_\_\_\_\_
- A. 复用成功的设计
  - B. 保证程序运行速度达到最优
  - C. 减少设计中创建类的个数
  - D. 允许在非面向对象程序设计语言中使用面向对象的概念
- 19、下列关于风险的叙述不正确的是：风险是指\_\_\_\_\_。
- A. 可能发生的事件
  - B. 一定会发生的事件
  - C. 会带来损失的事件
  - D. 可能对其进行干预，以减少损失的事件
- 20、黑盒测试不能发现\_\_\_\_\_。
- A. 不正确或遗漏的功能。
  - B. 初始化或终止性错误。
  - C. 内部数据结构不合理。
  - D. 性能不满足要求。

#### 四. 简述题（共 30 分）

1. 请写出下列术语的英文全称或缩写：  
软件需求规格说明书，设计模式，验收测试。
2. 影响软件工程开发实践的关键要素是什么？
3. 螺旋模型四个象限的基本任务和四重循环的大致结果是什么？
4. 考虑到需求的优先级，举例说明获取需求时的需求基本分类。
1. 传统"瀑布模型"的主要缺陷是什么？试说明造成缺陷的主要原因。
- 3、对于需求文档，如何进行测试？（哪些方面？ 什么原则？）
7. 举例说明模块间的各种内聚（Cohesion）关系。
6. 功能测试的基本指导原则是什么？
5. 试述 COCOMOII 模型的工作原理。
8. 试述设计用户界面应考虑的问题

#### 五. 综合应用题（共 25 分）

- 1、（12 分）某客户信息管理系统中保存着两类客户的信息。
- （1）个人客户。对于这类客户，系统保存了其客户标识（由系统生成）和基本信息（包括姓

名、住宅电话和 E-mail)。

(2) 集团客户。集团客户可以创建和管理自己的若干名联系人。对于这类客户，系统除了保存其客户标识(由系统生成)之外，也保存了其联系人的信息。联系人的信息包括姓名、住宅电话、E-mail、办公电话以及职位。

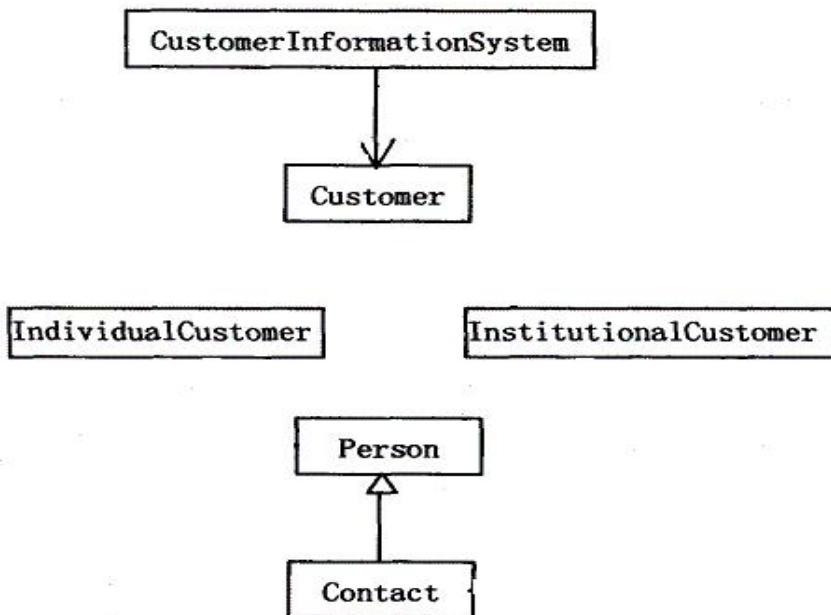
该系统除了可以保存客户信息之外，还具有以下功能：

- (1) 向系统中添加客户 (addCustomer)；
- (2) 根据给定的客户标识，在系统中查找该客户 (getCustomer)；
- (3) 根据给定的客户标识，从系统中删除该客户 (removeCustomer)；
- (4) 创建新的联系人 (addContact)；
- (5) 在系统中查找指定的联系人 (getContact)；
- (6) 从系统中删除指定的联系人 (removeContact)。

该系统采用面向对象方法进行开发。在面向对象分析阶段，根据上述描述，得到如下表所示的类。

类 名	说 明
CustomerInformationSystem	客户信息管理系统
IndividualCustomer	个人客户
InstitutionalCustomer	集团客户
Contact	联系人

描述该客户信息管理系统的 UML 类图如下图所示。



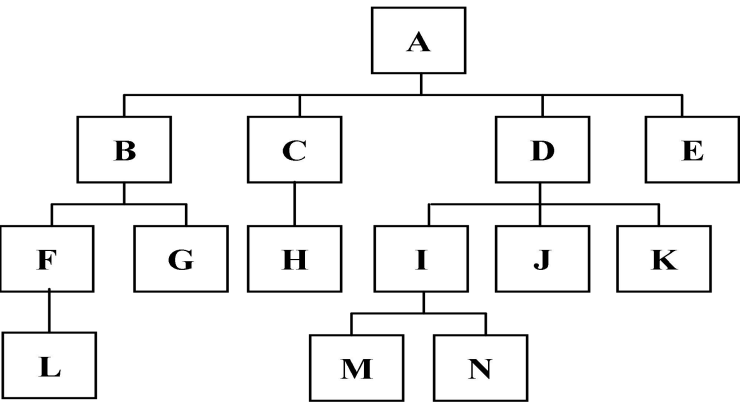


【题目要求】:

- 1、请使用需求定义中的术语，给出上图中类 Customer 和类 Person 的属性。
- 2、考虑上图，请画出类之间的联系。
- 3、根据说明中的叙述，抽象出如下表所示的方法，请指出类 CustomerInformationSystem 和 InstitutionalCustomer 应分别具有其中的哪些方法。

功 能 描 述	方 法 名
向系统中添加客户	addCustomer
根据给定的客户标识，在系统中查找该客户	getCustomer
根据给定的客户标识，从系统中删除该客户	removeCustomer
创建新的联系人	addContact
在系统中查找指定的联系人	getContact
从系统中删除指定的联系人	removeContact

2、(6 分)This figure illustrates the component hierarchy in a software system. Describe the sequence of tests for integrating the components using a bottom-up approach, a modified top-down approach.



- 3、（7 分）下面是一段求最大值的程序，其中 datalist 是数据表，n 是 datalist 的长度。

```
int GetMax(int n, int datalist[ ])
{
    int k=0;

    for ( int j=1; j<n; j++ )
```

```

        if ( datalist[j] > datalist[k] ) k=j;

    return k;

}

```

- (1) 画出该程序的**控制流图**。
- (2) 设计路径覆盖的测试用例。

## 历年概念题

2005	2007	2009
1. 过程 (Process)	1. 原型 (Prototype)	1. 抽象 (Abstraction)
2. 迭代式开发 (Iterative development)	2. 增量式开发 (Incremental Development)	2. 原型 (Prototype)
3. 开发模式 (Development)	3. 演化型原型 (Evolutionary	3. 分阶段开发模型 (Phased Development)

Paradigm) 4. 项目进度 (Project schedule) 5. 抛弃式原型 (Throw-away prototype) 6. 耦合 (Coupling) 7. 用例 (Use case) 8. 能力缺陷 (capacity fault) 9. 容量测试 (volume test) 10. 软件可靠性 (Reliability)	Prototype) 4. 面向对象 (Object Orientation) 5. 过载缺陷 (overload fault) 6. 强度测试 (stress tests) 7. 软件可维护性 (maintainability) 8. 由底向上测试 (Bottom-Up Test) 9. 极限编程 (XP) (Extreme Programming) 10. 项目进度 (Schedule)	4. 功能性需求 (Functional Requirement) 5. 公共耦合 (Common Coupling) 6. 用例 (Use Case) 7. 组合 (Composition) 8. 派对编程 (Pair Programming) 9. 路径测试 (Path Testing) 10. 驱动模块 (Component Driver) 11. 基线 (Base Line) 12. 软件可靠性 (Software Reliability) 13. 项目进度 (Project Schedule) 14. 能力缺陷 (Capacity fault) 15. 抛弃式原型 (Throw-away prototype)
2011	2012	2016
软件工程 重用 文档缺陷 分阶段模型 派对编程 α测试 组合 RUP 汉英解释: 验收测试 软件配置管理 设计模式	1. 统一过程 (UP-Unify Process) 2. 项目进度 (Project Schedule) 3. 重用 (Reuse) 4. 项目成本 (Project Cost) 5. 需求 (requirement) 6. 耦合 (Coupling) 7. 关联 (Association) 8. 极限编程 (XP-Extreme Programming) 9. 驱动模块 (Component Driver) 10. β 测试 (β / Beta Test)	<b>2. 过程 (Process)</b> <b>3. 功能性需求 (Functional Requirement)</b> <b>4. 黑盒测试 (Black-box Test)</b>