

# 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)
8. 能力缺陷 (capacity fault)
9. 容量测试 (volume test)
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:                      )

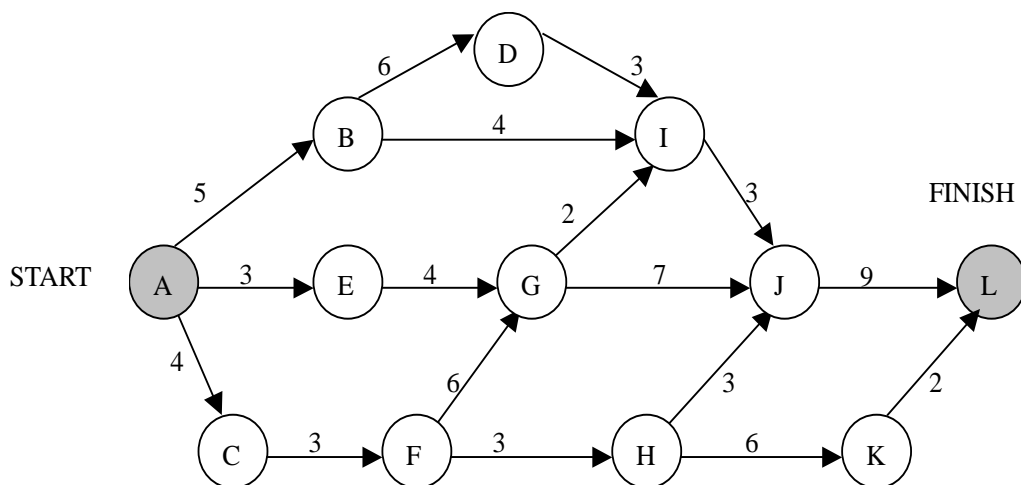
## 五. 简述题 (共 24 分)

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. 简述编程应考虑的三方面原则。
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。

### 一. 解释下列名词的含义

1. 过程：软件开发活动中产生某种期望结果的一系列有序任务，涉及活动、约束和资源
2. 迭代式开发：在需求中，系统初次提交时是一个完整的系统功能框架，后续版本是对原先版本功能的改变或增强。
3. 开发模式：软件开发全部过程、活动和任务的结构框架，能直观表达软件开发全过程，明确规定需要完成的重要活动，任务和开发策略，是软件开发方法的策略或哲学。
4. 项目进度：是对特定项目的软件开发周期的刻画。是通过对项目阶段、步骤、活动的分解而得到的。
5. 抛弃型原型：仅用于了解问题、探索可行性，并不打算用来作为将来实际提交系统的一部分，而是用完扔掉。
6. 耦合：软件模块之间的相互依赖性
7. 用例：通过建立用户、外部项、其他实体的对话模型，而对系统将要完成的功能进行描述或刻画。
8. 能力缺陷：当系统的活动达到设定的极限（例如设备数量）时，系统性能变得不可接受。
9. 容量测试：验证系统处理巨量数据的能力（检验数据结构是否足够大，以能够处理各种情况）。
10. 软件可靠性：软件系统在给定的时间间隔和给定条件下运行成功的概率。

### 二. 判断

1. x      2. √      3. x      4. x      5. x      6. x      7. √      8. √      9. x      10. x

### 三. 填空

1. 文档
2. 软件过程
3. 问题定义，《SRS》草稿，可行性研究，正式的《SRS》文档
4. 驱动模块，桩模块
5. 维护
6. 单继承，多重继承
7. 矩形，圆圈，单箭头，双横线
8. alpha 测试，beta 测试
9. 算法缺陷，计算和精度缺陷，过载缺陷/能力缺陷
10. 强度测试，容量测试，配置测试/兼容性测试
11. 作为一种部分开发的产品，用来让用户和开发者共同研究，提出意见，为最终产品定型

### 四. 选择

1. D      2. C      3. C      4. B      5. B      6. B      7. B  
8. [A] 2      [B] 4      [C] 1      [D] 4      [E] 4

### 五. 简述题

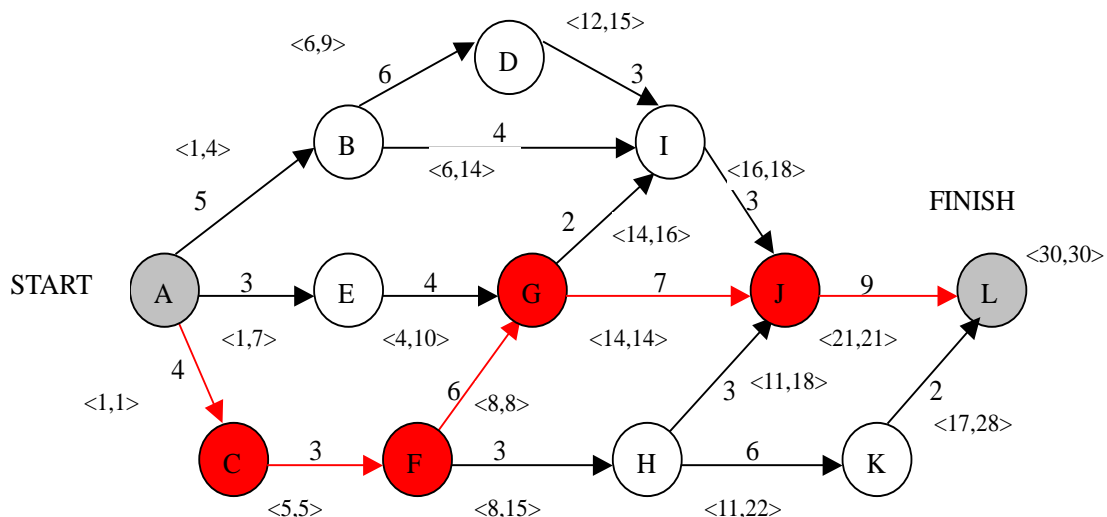
1. 错误是人为造成的错误，缺陷是程序功能中出现的问题，失败是软件运行时出现的故障。  
一个错误可能导致若干个缺陷，但缺陷不一定导致失败
2. 需求分析（项目计划，可行性研究报告，SRS”需求规格说明）、系统设计（软件结构图等）、程序设计（算法和数据描述文档）、编码（源程序及注释）、单元测试（测试报

告)、集成测试 (测试报告)、系统测试 (测试报告)、系统提交 (用户手册等), 维护 (维护报告)

3. 要点: 软件产品的质量, 软件过程的质量, 软件的商业价值
4. 必须满足的用户需求, 用户热切希望但不是必须的需求, 技术上有可能实现但必须去掉的需求
5. 影响软件工程开发实践发生变化的关键要素是:
  - ①. 商业产品推向市场的时间的重要性
  - ②. 计算经济学的改变 (计算行业经济的发展趋势: 硬件费用越来越低, 而开发、维护费用越来越高)
  - ③. 功能强大的桌面计算平台的出现
  - ④. 局域网和广域网的延伸
  - ⑤. 面向对象技术的出现及其应用
  - ⑥. 使用窗口、图标、菜单和指针的图形用户界面
  - ⑦. 软件开发瀑布模型的不可预知性 (时间、费用的不可预知性)
6. 偶然性内聚: 各部分功能上互不相关  
逻辑性内聚: 模块各部分功能相似  
时间性内聚: 模块各部分要求在同一时间完成  
过程性内聚: 各部分有一定次序  
通讯性内聚: 各部分访问共享数据  
顺序性内聚: 各部分之间有输入输出关系  
功能性内聚: 模块各部分合起来完成一个单一的功能。
7. 要点: 控制结构, 算法结构, 数据结构
8. 功能测试: 针对功能性需求的测试  
性能测试: 针对非功能性需求的测试  
确认测试 (验收测试): 由客户主导的对系统进行的测试, 看是否符合需求定义的要求  
安装测试: 在用户工作环境下的测试, 目的是解决开发环境和用户环境的不同所导致的问题

## 六. 计算和应用题 (共 22 分)

- 1、关键路径: ACFGJL (其他标注必须在图上注明)



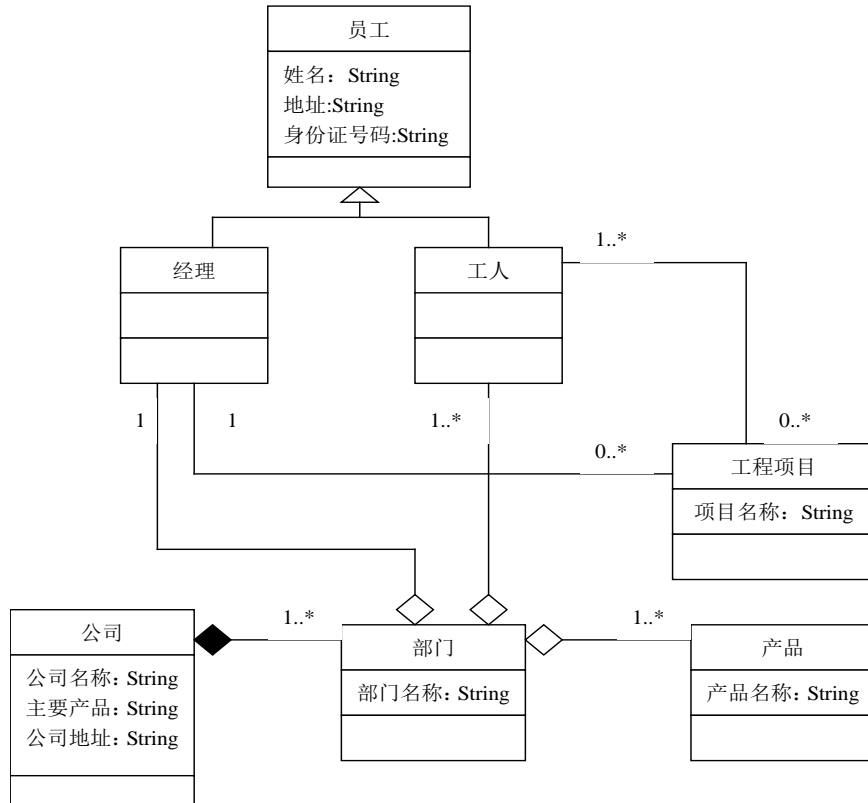
2、(1) DFD 图：1-2 个外部实体，四个数据存储，三个加工，另外加数据流及说明。

(2) Use Case。

至少有 3 个用例：计算标准分，计算录取分数线，考生信息查询。1 个活动者：用户  
有用例“分类服务”并与以上三个用例建立扩展关系的，可适当加分。

3. UML 类图

要求使用正确的符号表达类、类关系，注明类关联基数。



一、解释下列名词的含义

1. 原型：一种部分开发的产品，用来让用户和开发者共同研究，提出意见，为最终产品定型
2. 增量式开发：系统需求按照功能分成若干子系统，开始建造的版本是规模小的、部分功能的系统，后续版本添加包含新功能的子系统，最后版本是包含全部功能的完整系统
3. 演化型原型：该原型的开发是用于了解问题，并作为将来准备提交的系统的一部分。
4. 面向对象：是一种软件开发方法，它将问题和方法组织成一系列独立的不连续的对象，数据结构和动作都被包括在内。
5. 过载缺陷 (overload fault)：软件内部设定的数据结构被填充后溢出，系统不再能完成各种功能。
6. 强度测试 (stress tests)：系统在短时间内加载极限负荷，以验证系统性能。(例如：系统在最大设备数量的情况下进行性能测试)
7. 软件可维护性：是指在给定的使用条件 (预定的时间间隔、维护程序、维护资源之下进行维护) 下，维护活动能被执行的概率。
8. 由底向上测试：集成测试时从模块结构图的最低层开始，由下而上按调用关系逐步添加新模块，组成子系统并分别测试，直到全部模块组装完毕为止。
9. 极限编程(XP)：极限编程 (XP) 是一种轻量级的软件开发方法论，属于敏捷开发方法。XP 的主要特征是要适应环境变化和需求变化，充分发挥开发人员的主动精神。(XP 承诺降低软件项目风险，改善业务变化的反应能力，提高开发期间的生产力，为软件开发过程增加乐趣等等。)
10. 项目进度 (Schedule)：项目进度是对特定项目的软件开发周期的刻画。包括对项目阶段、步骤、活动的分解，对各个任务的交互关系的描述，以及对各活动完成时间的初步估算等。

一. 判断

1. √    2. x    3. x    4. x    5. x    6. x    7. x    8. x    9. x    10. x

二. 填空

1. 软件危机
2. 驱动模块，桩模块
3. 产品质量，过程质量，商业质量
4. 原始需求获取，问题分析，需求规格说明草稿，需求核准，正式的需求规格说明(SRS)
5. 软件维护
6. alpha 测试， beta 测试
7. 强度测试，容量测试，配置测试，(兼容性测试，回归测试，安全性测试等)
8. 操作概念，软件需求，软件设计，系统实现与执行。

三. 选择

1. A.    2. B    3. B    4. A    5. B    6. D    7. B    8. B  
9. [A] 2    [B] 4    [C] 1    [D] 4    [E] 4

四. 简述题

1. 软件开发将软件的开发过程分为若干阶段，包括需求分析、系统设计、程序设计、编码、测试等等，而程序设计仅是软件开发的一个组成部分，并且软件开发所指的软件不同于一般程序，而是指大型程序及文档。



2. 影响软件工程开发实践发生变化的关键要素是：

- ①. 商业软件产品的时间--市场的重要性
- ②. 计算经济学的改变（计算行业经济的发展趋势：硬件费用越来越低，而开发、维护费用越来越高）
- ③. 功能强大的桌面计算平台的出现
- ④. 局域网和广域网的延伸
- ⑤. 面向对象技术的出现及其应用
- ⑥. 使用窗口、图标、菜单和指针的图形用户界面
- ⑦. 软件开发瀑布模型的不可预知性（时间、费用的不可预知性）

3. COCOMO II 的估算过程反映了任何一个软件开发项目的三个主要阶段。在阶段 1(计划阶段)中，COCOMO II 用所谓的应用点来估计规模。对于阶段 2(早期设计)，COCOMO II 采用了功能点作为对规模的度量。在阶段 3(次结构阶段)，规模可以用功能点或代码行的形式来表述。

4. (1) 必须满足的用户需求；

(2) 用户热切希望但不是必须的需求；

(3) 技术上有可能实现但必须去除的需求

5. (1) 设计界面要注意解决的要素：寓意/比喻，思维模型，领航规则，外观，感觉

(2) 文化差异问题

(3) 用户爱好问题

6. 偶然性内聚：各部分功能上互不相关

逻辑性内聚：模块各部分功能相似

时间性内聚：模块各部分要求在同一时间完成

过程性内聚：各部分有一定次序

通讯性内聚：各部分访问共享数据

顺序性内聚：各部分之间有输入输出关系

功能性内聚：模块各部分合起来完成一个单一的功能。

（至于举例，可以酌情给出分数）

7. UML 类图之间关系的类型通常有 5 种：归纳、关联、依赖、聚集和组合。

（若上述类型不够 5 分，但又有举例的话可考虑适当给满分）

8. (1) 传统测试：当系统改变时，需要新老测试用例（回归测试）

(2) OO 测试：必须对重载的子类进行测试，可能会使用不同的测试用例

(3) OO 测试：单元测试比较简单，但集成测试比较广泛。

9. 功能测试：针对功能性需求的测试

性能测试：针对非功能性需求的测试

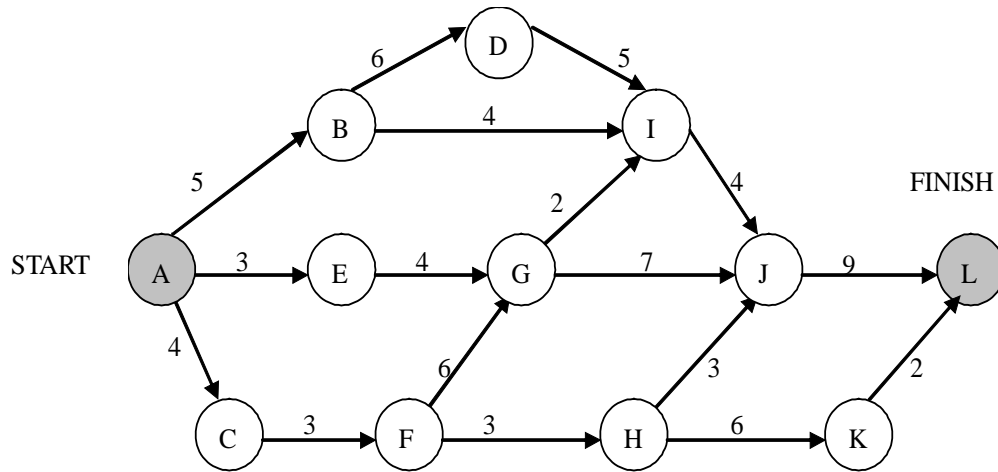
确认测试（验收测试）：由客户主导的对系统进行的测试，看是否符合需求定义的要求

安装测试：在用户工作环境下的测试，目的是解决开发环境和用户环境的不同所导致的问题

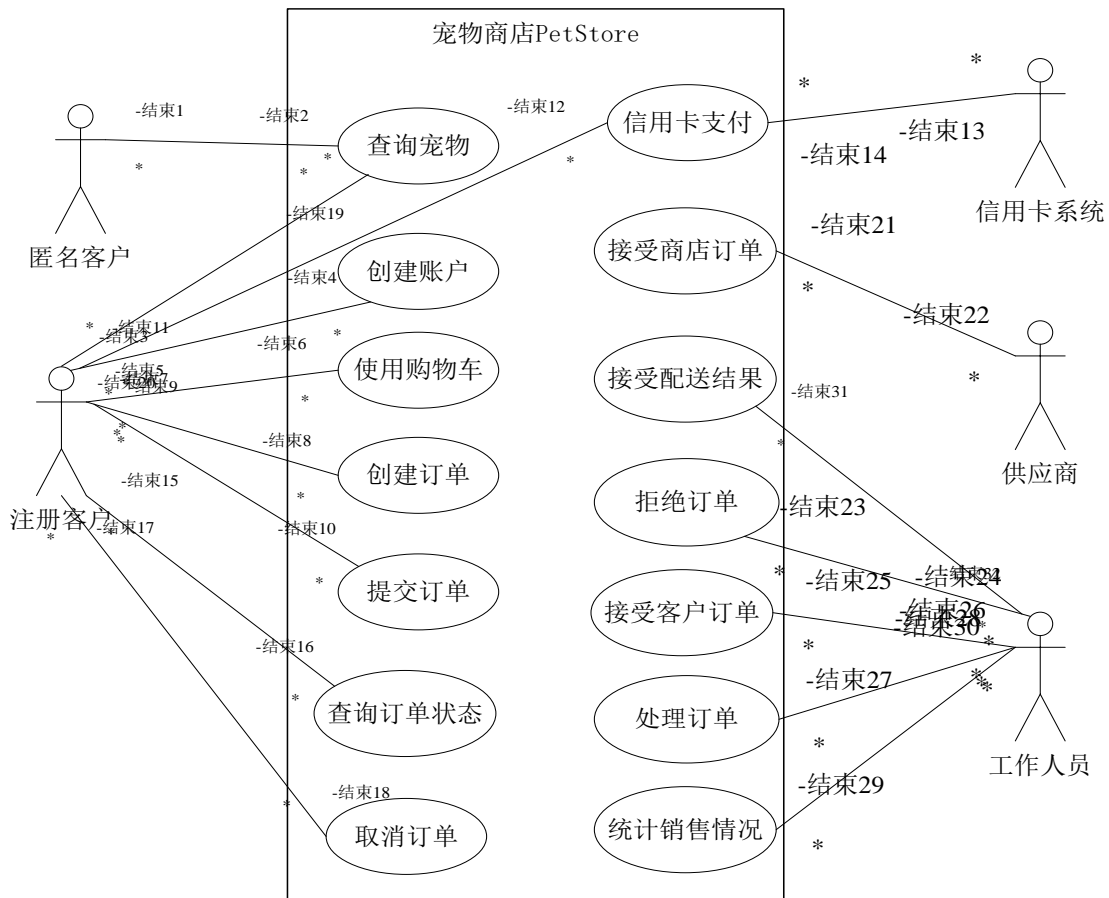
（若前面多写了单元测试，集成测试，则应适当减分）

## 六. 计算和应用题（共 20 分）

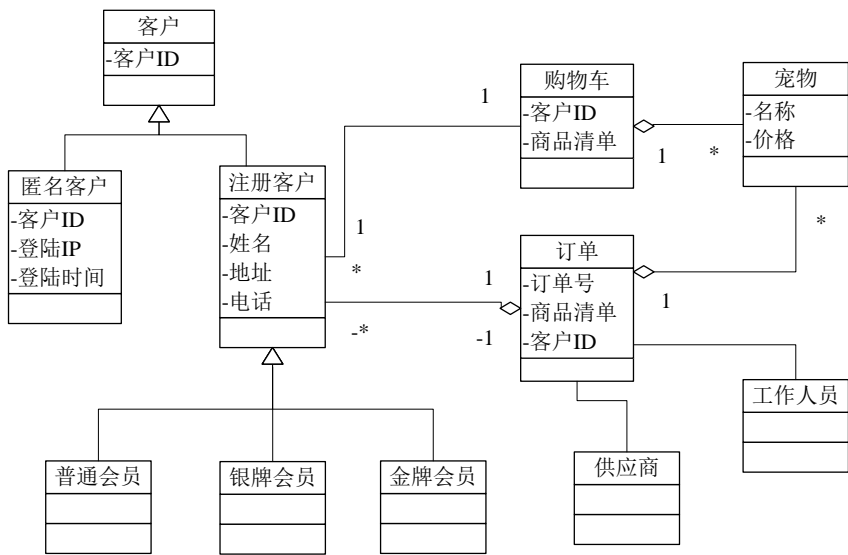
1、关键路径： ABDIJL 和 ACFGJL。（其他标注必须在图上注明）



2. (1)。



(2)



3.

参考答案：

路径	测试用例
Y	I=20
NNNY	I=10,X=3,Y=3
NNYY	I=10,X=2,Y=-1
NYYN	I=10,X=1,Y=1
NYYY	(不存在测试用例)

# 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: )

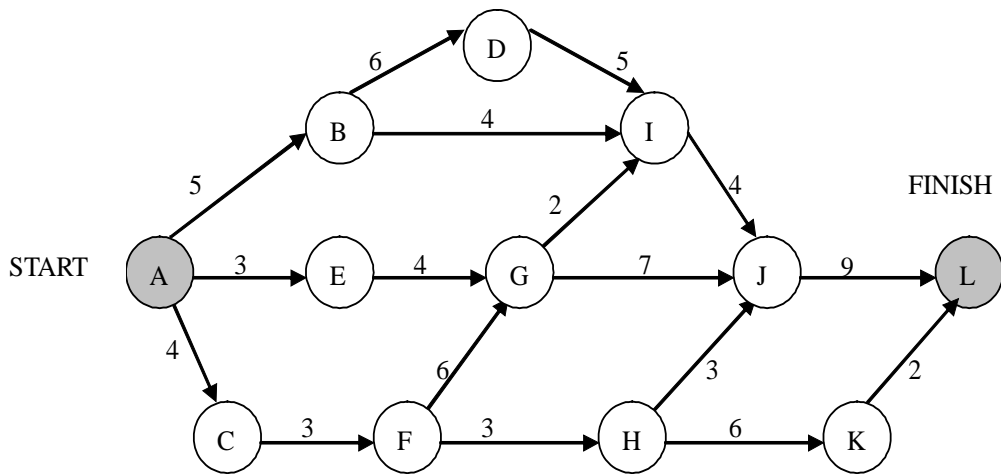
## 五. 简述题 (共 27 分)

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；创建账户并登陆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。

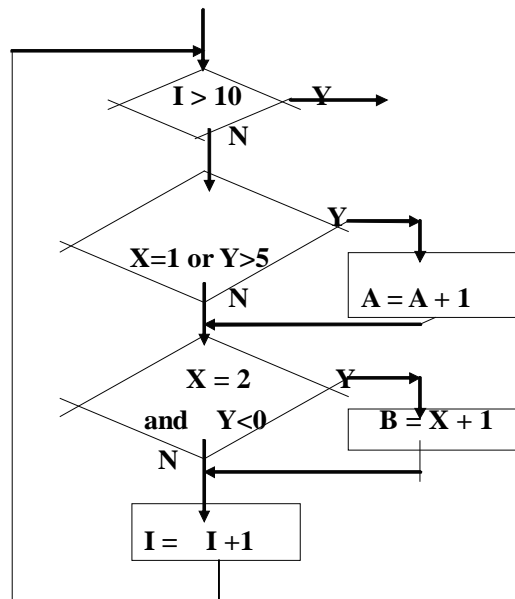
宠物商店的工作人员能够接受或拒绝客户提交的订单、处理订单、发订单给供应商、接受供应商返回的配送结果；还能够统计销售情况。

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。





# 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. 结对编程（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。(每个小题或题空为 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。 ( )

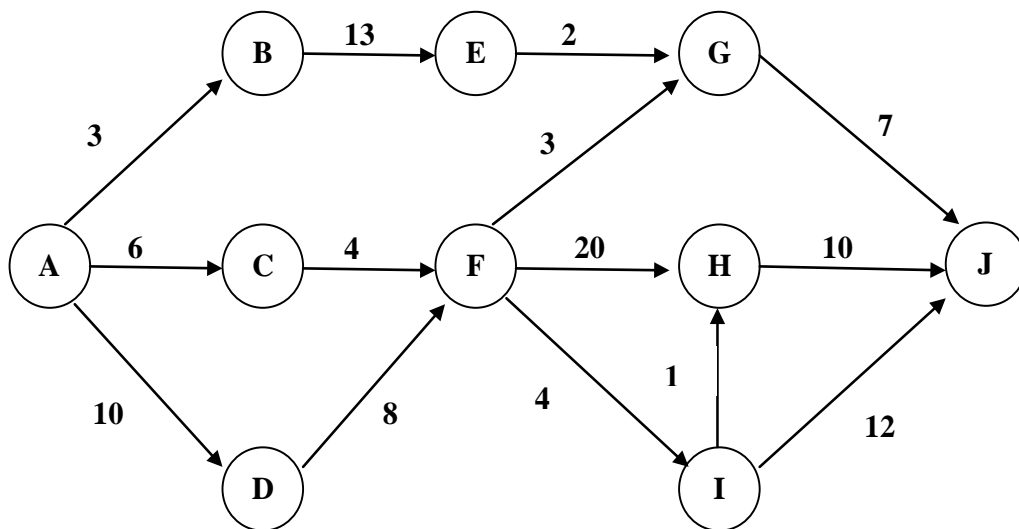
(3) 需求分析人员不知道读者在还书的时候不需要出示借书卡。 ( )

(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). 以下哪一个是从里程碑 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】

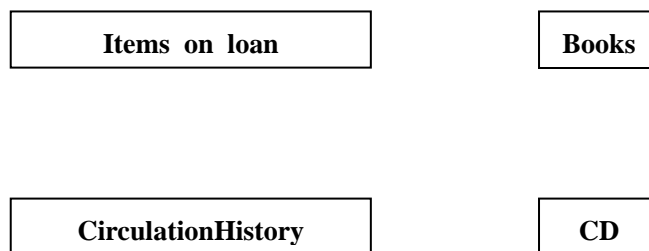
某公司的主要业务是出租图书和唱碟。由于业务需求，该公司委托软件开发公司 A 开发一套信息管理系统，该系统将记录所有的图书信息、唱碟信息、用户信息、用户租借信息等。A 公司决定采用面向对象的分析和设计方法开发此系统。图[1]所示为某类图书或唱碟被借阅时应记录的信息，图[2]描述了系统定义的两个类 Book 和 CD，分别表示图书和唱碟的信息。

### 【问题 1】

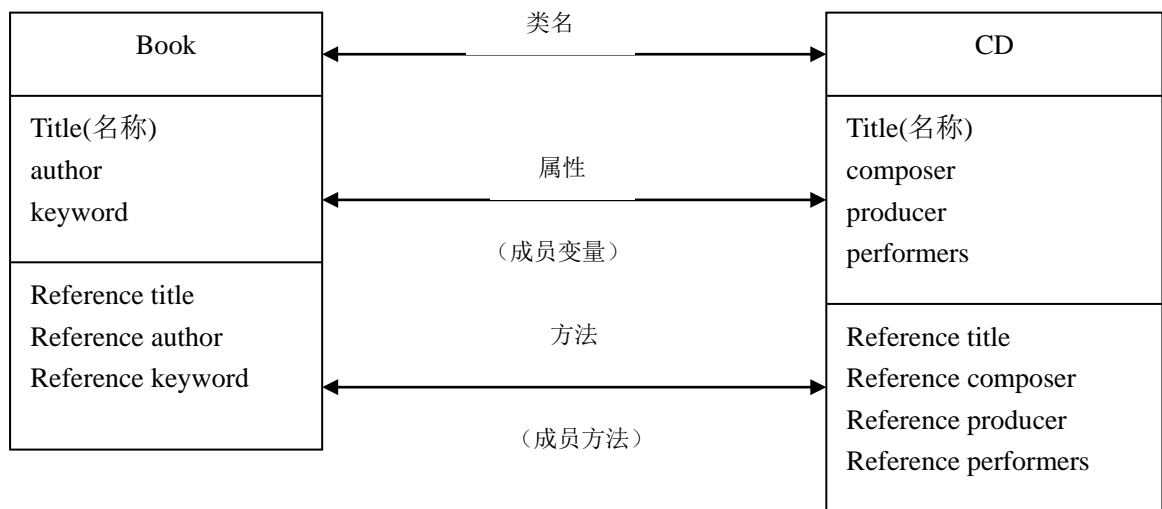
经过进一步分析，设计人员决定定义一个类 Items\_on\_loan，以表示类 Book 和 CD 的共有属性和方法。请采用图[2]中属性和方法的名称给出类 Items\_on\_loan 应该具有的属性和方法。（注意：不同名称的属性和方法表示不同的含义，如 CD 中的 composer 与 Book 中的 author 无任何关系。）

### 【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> <td style="width: 33%;">借出时间</td> <td style="width: 33%;">归还时间</td> <td style="width: 33%;">用户</td> </tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td></tr> <tr><td> </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。

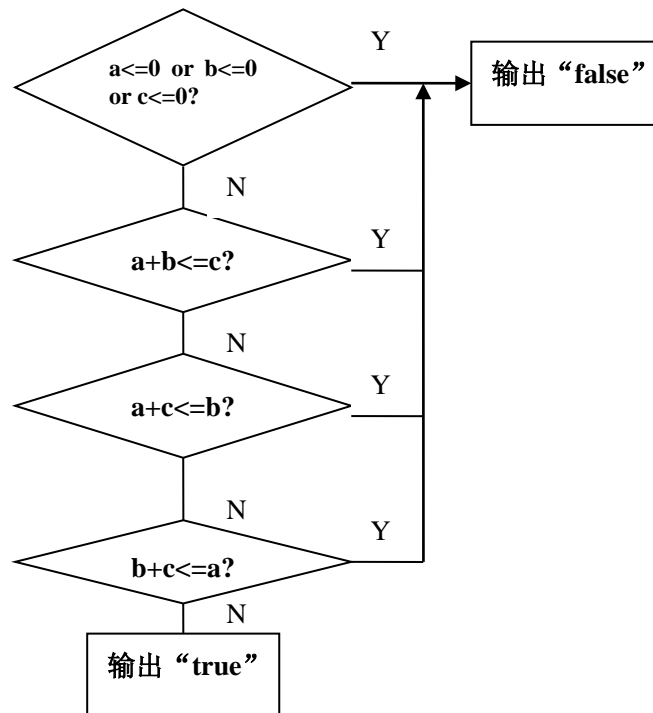
茶庄(TeaStall)包含了以上所有,但是它有容量限制,即可容纳的最多茶客数,每天开店时,派出一位沏茶员和一位侍者,并接待最多数量的茶客。每天闭店后,会统计整个茶庄当

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：

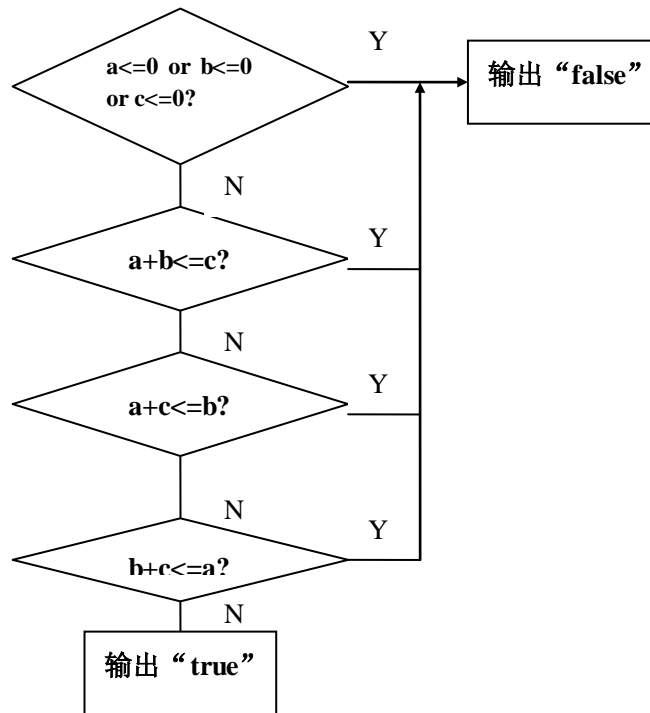


假设程序中变量均为整型变量，请设计测试用例，满足“路径覆盖 (Path Testing)”的要求。

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 年软件工程

### 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）



## 一、理论应用题

写出下列软工的理论和其针对的领域应用。

1. Chief programmer team, egoless approach 应用于什么领域，各有什么特点？结合自己的项目实践，谈谈你们是如何做的？（10 分）

答：

Chief Programmer Team即主程序员负责制，由一个主程序员负责系统设计和开发，其他的成员向其汇报，主程序员对每一个决定有绝对决策权。特点是：可以使交流最小化、迅速做出决定，缺点是创造性低、对主程序员要求高，个人主观性强。

Egoless Approach即忘我方法，每个成员平等的承担责任，而且过程与个人是分开的；批评是针对产品和结果的，不针对个人的。

二者应用于计划和管理项目中的项目团队组织领域。

我们的项目实践 SDUOJ 小组，共 6 人，6 个人全部属于程序算法编码基础强，但是大多工程经验较弱的成员类型。我本人作为后端开发，难免要协调前端和评测端，所以也负责系统设计和开发，但 SDUOJ 团队谈不上是 Chief Programmer Team，我也谈不上是主程序员。我们 6 人是使用 Egoless Approach 的团队，每个人负责一部分，比如前端、后端、评测端，每人平等地承担责任，自主学习，每周定时组会汇报学习进度和编码进度，遇到决策困难和设计困难会在组会上讨论，调研，最终投票做出决定，由具体人员负责开发。属于交流充分，减少开发周期反复联调的项目实践方法。

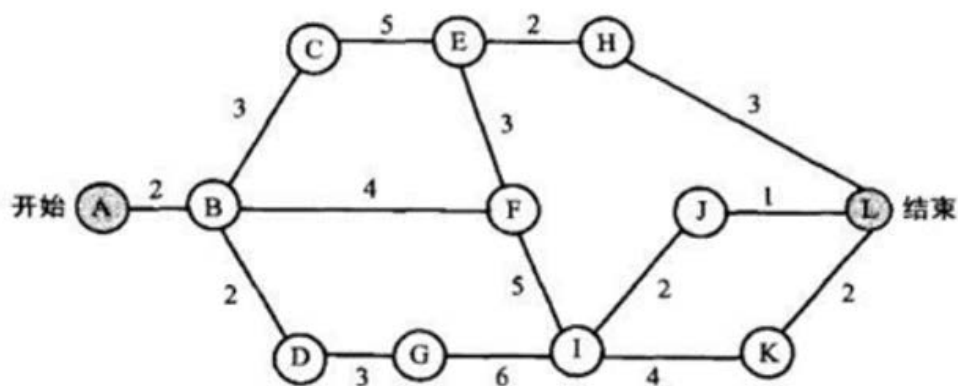
2. Incremental development, iterative development 应用于什么领域，各有什么特点？结合自己的项目，举例说明你们的实践？（10 分）

答：

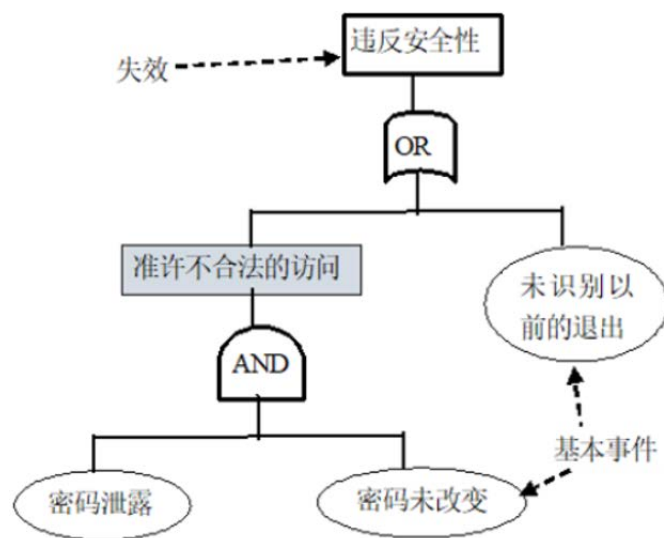
Incremental development, iterative development

## 二、分析题

1. 下图是一个软件开发项目的活动图，请分析标出每一个活动的最早开始时间、最晚开始时间和时差。然后找出关键路径和其总长度。（10 分）



2. 下图是违反安全性 Fault-tree 故障树，请由它设计生成 Cut-set-tree 割集树。(5 分)



### 三、综合设计题

1. 结合自己的项目开发时间，从体系结构、OO 设计原则等（至少选两个案例）方面论述你在提高软件指令（易重用、维护等）方面所做的重构工作。(15 分)

## 一、名词解释（共15分）

1、极限编程 XP， Extreme Programming

2. 结对编程, Pair programming

3. 软件生命周期, Software Life Cycle

4. 设计模式, design pattern

5. 项目调度, project scheduling;

## 二、判断题（共5分）

1. 详细设计也称模式设计 ( )

2. 在数据代码设计时, 应尽量让一条代码代表多个信息 ( )

3. 在输出界面设计时, 要尽可能使用代码或缩写, 以求简洁 ( )

4. 在数据代码设计时, 尽可能设计字母和数字混合代码 ( )

5. 详细设计评审应尽可能和概要设计评审一同进行 ( )

## 三、简述题（共40分）

1. 什么是软件过程? 它与软件工程方法学有何关系?

2. 简述软件错误(Error)、故障(Fault)和失效(Failure)之间的区别。

3. 试比较瀑布模型、快速原型模型、增量模型和螺旋模型的优缺点, 说明它们各自的适用范围。

4. 通常系统的需求可以分为哪几个层次?

5. 如何衡量软件质量?

6. 传统测试与 OO 测试有何不同?

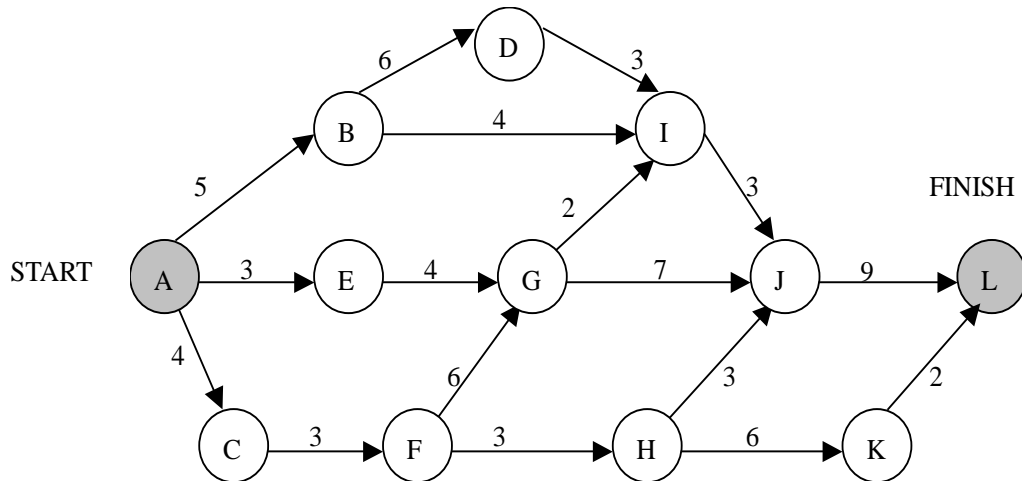
7. 工作量估计有哪几种方法?

8. 外部文档需要完成哪些工作, 并简要解释这些工作?

#### 四、应用题（共20分）

1. 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。



2. 问题求解的步骤有哪些（画出图并加以解释）？

#### 五、综合题（共20分）

1. 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。

2. 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；创建账户并登陆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。

# 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）

基于某种归纳水平的问题描述，他是我们将注意力集中在问题的的关键方面而非细节。

2. 原型（Prototype）

一种部分开发的产品，用来让用户和开发者共同研究，提出意见，为最终产品定型

3. 分阶段开发模型（Phased Development）

系统被设计成部分提交，每次用户只能得到部分功能，而其他部分处于开发过程中。

4. 功能性需求（Functional Requirement）

描述系统内部功能或系统与外部环境的交互作用。涉及系统输入应对，实体状态变化，输出结果，设计约束与过程约束等。

5. 公共耦合（Common Coupling）

不用模块访问公共数据

6. 用例（Use Case）

7. 组合（Composition）

当一个类是另一个类的部分时，二者关系为组合

8. 派对编程（Pair Programming）

属于主要的敏捷开发方法，方式是两个程序员共同开发程序，且角色分工明确，一个负责编写，另一个负责审核与测试，两人定期交换角色

9. 路径测试（Path Testing）

借助于程序图设计测试用例的一种白盒方法，测试用例基于流程图来设计。

10. 驱动模块（Component Driver）

11. 基线（Base Line）

12. 软件可靠性（Software Reliability）

软件系统在给定的时间间隔和给定的条件下运行成功的概率。

13. 项目进度（Project Schedule）

项目进度是对特定项目的软件开发周期的刻画。

14. 能力缺陷（Capacity fault）

系统的处理超过需求描述中的最大负载就会引发压力缺陷

15. 抛弃式原型（Throw-away prototype）

仅用于了解问题、探索可行性，并不打算用来作为将来实际提交系统的一部分，而是用完扔掉

二. 判断（填写 × 或 √）（每个小题 1 分，共 15 分）

1. ( F ) 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. ( F ) 当前管理信息系统建设的主要问题是技术问题。

3. ( T ) 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。



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\_\_\_\_C\_\_\_\_个。25\*30/15  
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。  
\_\_\_\_A\_\_\_\_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\_\_\_\_B\_\_\_\_。

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\_\_\_\_D\_\_\_\_。

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。            ( A )

(2) 当管理员试图向库存中增加一本新书时, 系统崩溃了。            ( C )

(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。            ( A )

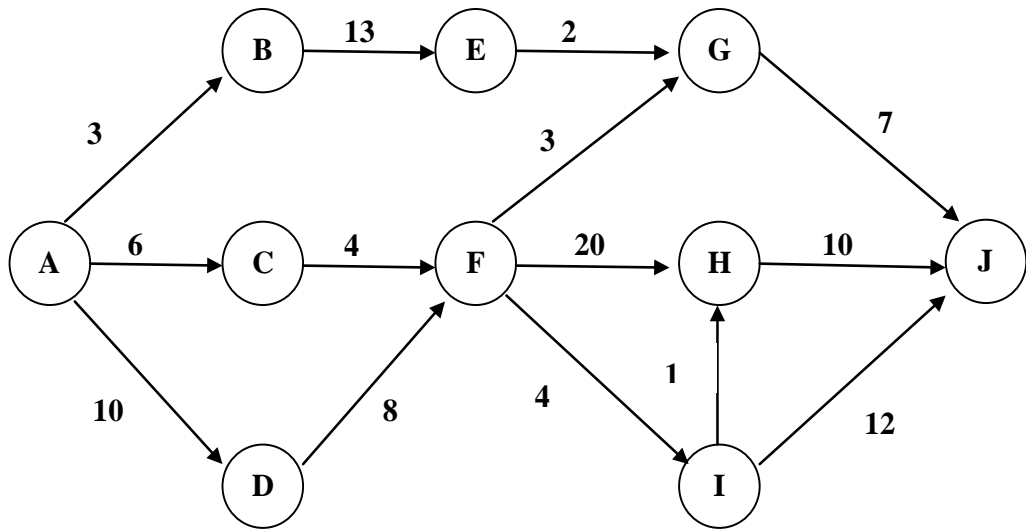
(4) 在需求文档中, 超期罚款为每天 0.25 元, 但最高不超过 15 元。在计算超期罚款的代码中并没有检查这个最大值。            ( A )

(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。 ( B )

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

12. 以下是某个项目的活动图:





(1). 以下哪一个是从里程碑 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】

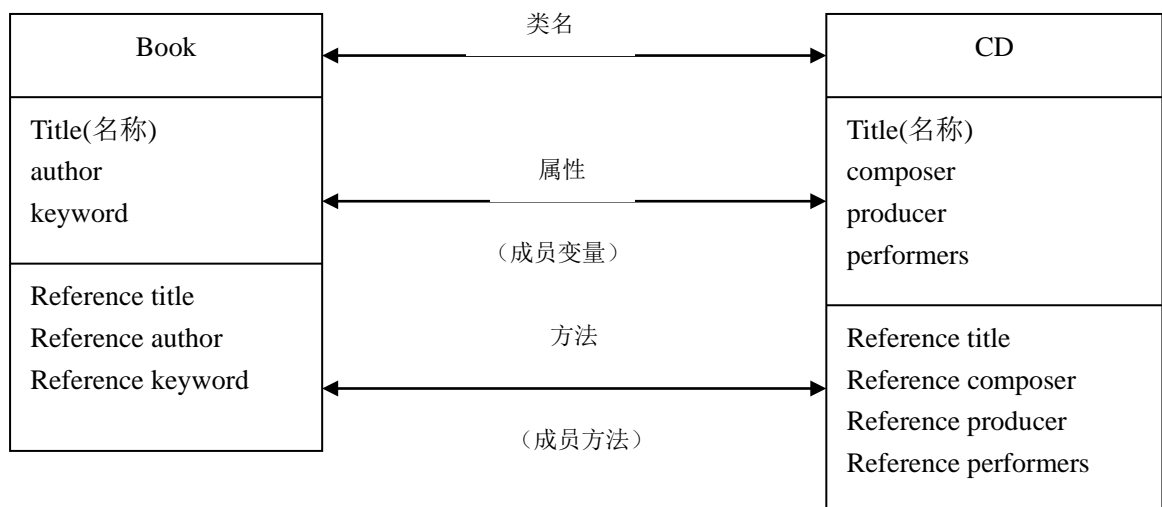
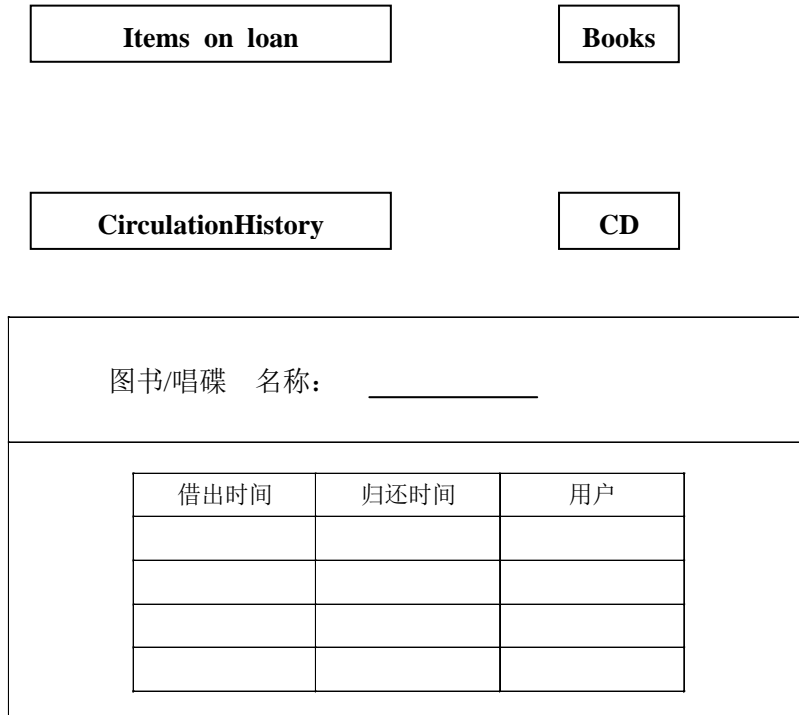
某公司的主要业务是出租图书和唱碟。由于业务需求, 该公司委托软件开发公司 A 开发一套信息管理系统, 该系统将记录所有的图书信息、唱碟信息、用户信息、用户租借信息等。A 公司决定采用面向对象的分析和设计方法开发此系统。图[1]所示为某类图书或唱碟被借阅时应记录的信息, 图[2]描述了系统定义的两个类 Book 和 CD, 分别表示图书和唱碟的信息。

### [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。



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。

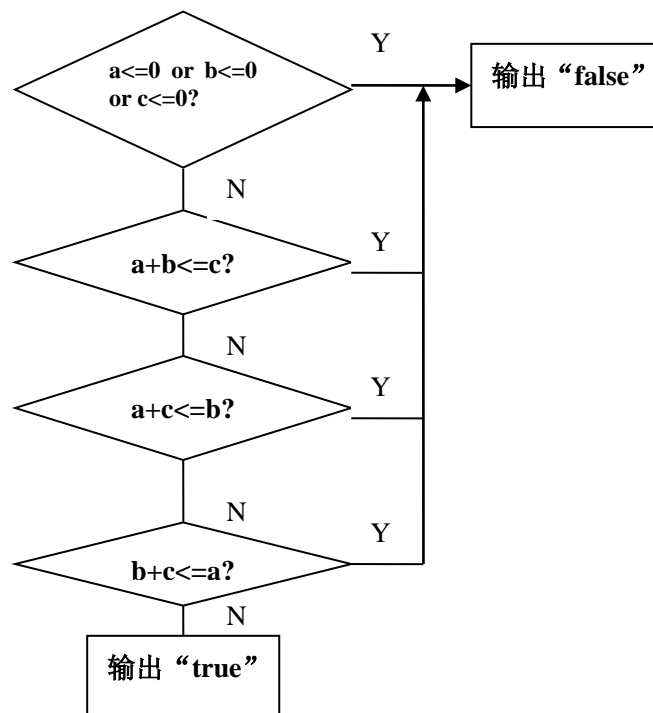
（1）画出用例图；

（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：

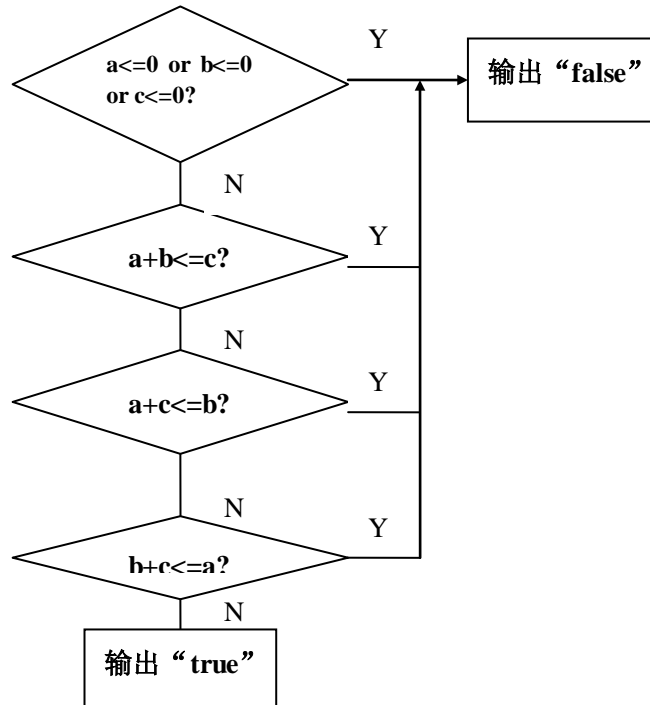


假设程序中变量均为整型变量，请设计测试用例，满足“路径覆盖（Path Testing）”的要求。

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。

# 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）

基于某种归纳水平的问题描述，他是我们将注意力集中在问题的的关键方面而非细节。

2. 原型（Prototype）

一种部分开发的产品，用来让用户和开发者共同研究，提出意见，为最终产品定型

3. 分阶段开发模型（Phased Development）

系统被设计成部分提交，每次用户只能得到部分功能，而其他部分处于开发过程中。

4. 功能性需求（Functional Requirement）

描述系统内部功能或系统与外部环境的交互作用。涉及系统输入应对，实体状态变化，输出结果，设计约束与过程约束等。

5. 公共耦合（Common Coupling）

不用模块访问公共数据

6. 用例（Use Case）

7. 组合（Composition）

当一个类是另一个类的部分时，二者关系为组合

8. 派对编程（Pair Programming）

属于主要的敏捷开发方法，方式是两个程序员共同开发程序，且角色分工明确，一个负责编写，另一个负责审核与测试，两人定期交换角色

9. 路径测试（Path Testing）

借助于程序图设计测试用例的一种白盒方法，测试用例基于流程图来设计。

10. 驱动模块（Component Driver）

11. 基线（Base Line）

12. 软件可靠性（Software Reliability）

软件系统在给定的时间间隔和给定的条件下运行成功的概率。

13. 项目进度（Project Schedule）

项目进度是对特定项目的软件开发周期的刻画。

14. 能力缺陷（Capacity fault）

系统的处理超过需求描述中的最大负载就会引发压力缺陷

15. 抛弃式原型（Throw-away prototype）

仅用于了解问题、探索可行性，并不打算用来作为将来实际提交系统的一部分，而是用完扔掉

二. 判断（填写 × 或 √）（每个小题 1 分，共 15 分）

1. ( F ) 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. ( F ) 当前管理信息系统建设的主要问题是技术问题。

3. ( T ) 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。



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\_\_\_\_C\_\_\_\_个。25\*30/15  
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。  
\_\_\_\_A\_\_\_\_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\_\_\_\_B\_\_\_\_。

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\_\_\_\_D\_\_\_\_。

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。 ( A )

(2) 当管理员试图向库存中增加一本新书时, 系统崩溃了。 ( C )

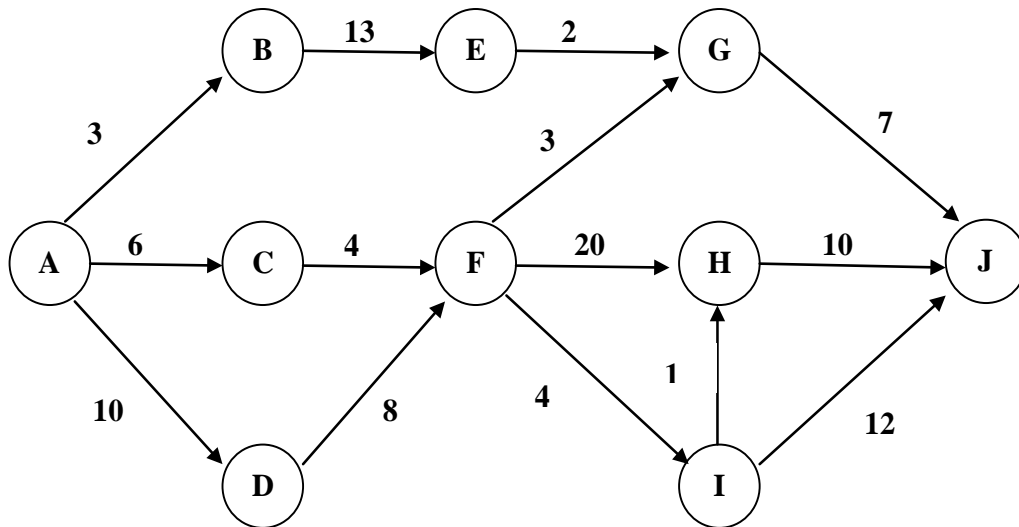
(3) 需求分析人员不知道读者在还书的时候不需要出示借书卡。 ( A )

(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。 ( A )

(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。 ( B )

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

12. 以下是某个项目的活动图:



(1). 以下哪一个是从里程碑 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。

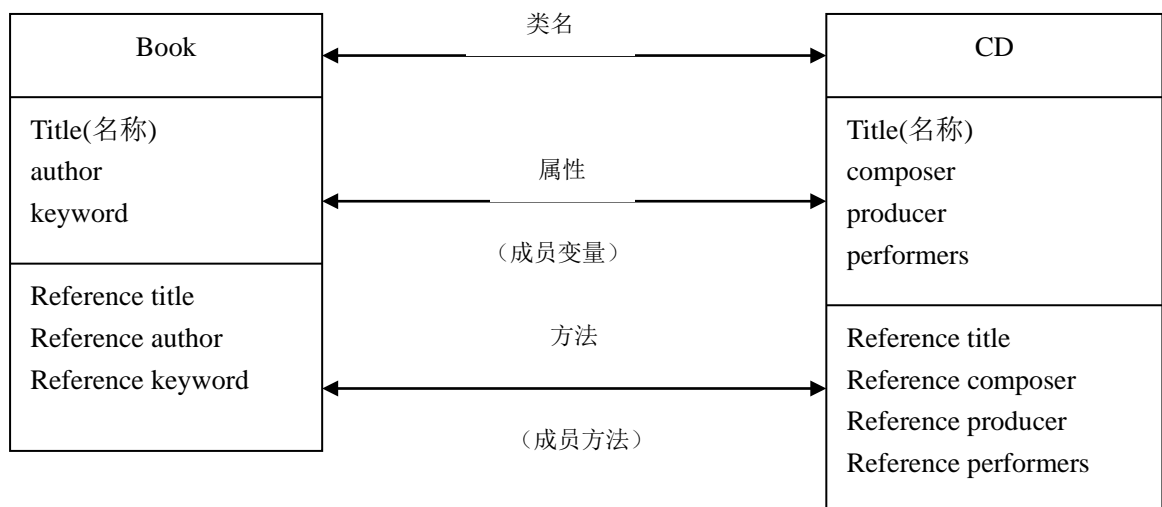
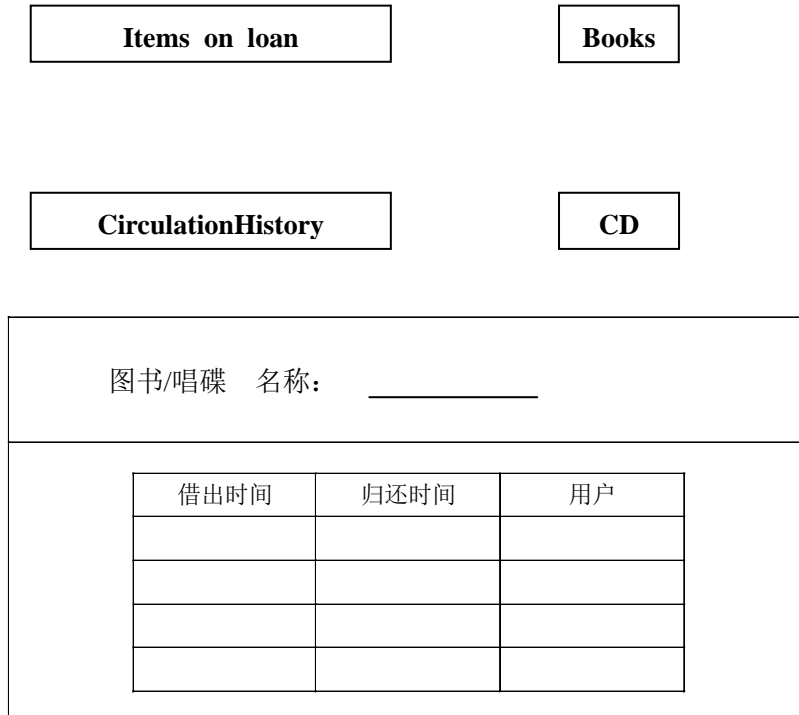


### [问题 1]

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。



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。

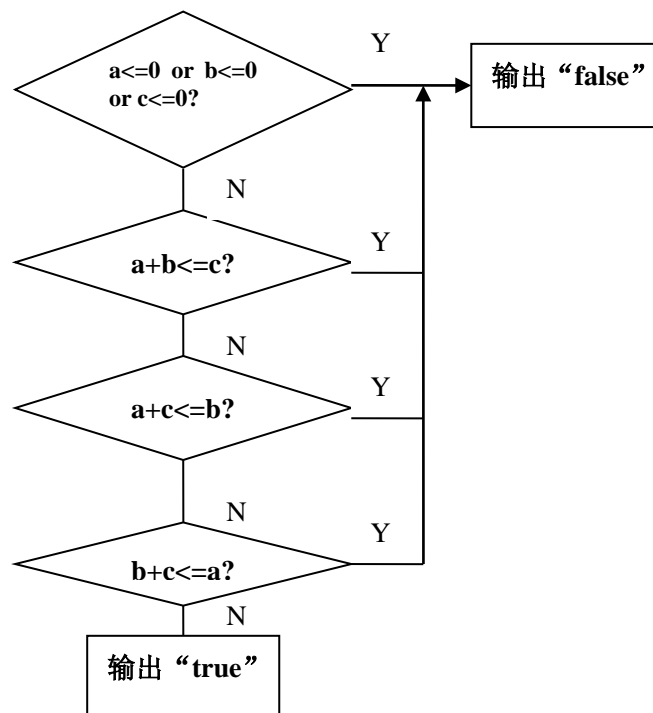
（1）画出用例图；

（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：

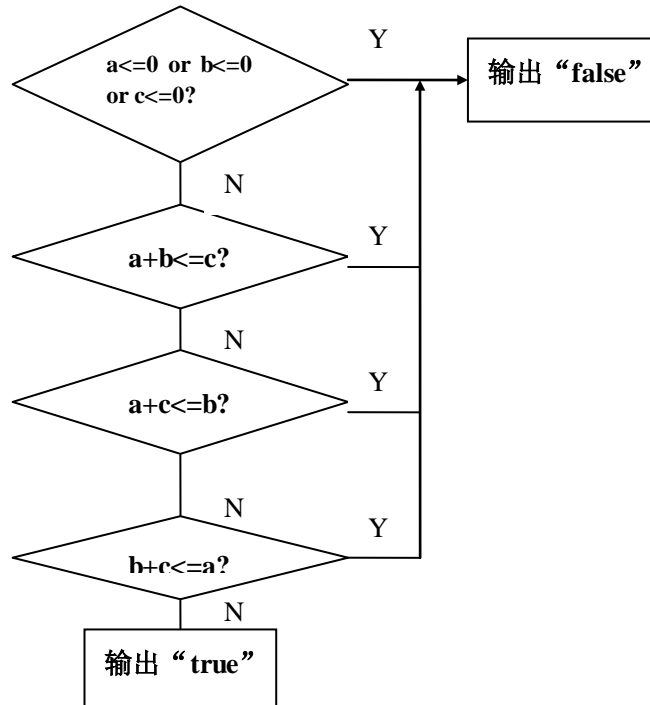


假设程序中变量均为整型变量，请设计测试用例，满足“路径覆盖（Path Testing）”的要求。

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。

姓名：

学号：

班级：

《软件工程》试卷（A 卷）

题号	一	二	三	四	五	总分
得分						

一. 单项选择题（每小题 1 分，共 22 分）

- 对象实现了数据和操作的结合，使数据和操作（ C ）于对象的统一体中。  
A. 结合                      B. 隐藏                      C. 封装                      D. 抽象
- 软件生命周期中所花费用最多的阶段是（ D ）。  
A. 详细设计                      B. 软件编码                      C. 软件测试                      D. 软件维护
- 需求分析中开发人员要从用户那里了解（ A ）。  
A. 软件做什么                      B. 用户使用界面                      C. 输入的信息                      D. 软件的规模
- 测试的关键问题是（ D ）。  
A. 如何组织对软件的评审                      B. 如何验证程序的正确性  
C. 如何采用综合策略                      D. 如何选择测试用例
- （ D ）是将系统化的、规范的、可量化的方法应用于软件的开发、运行和维护的过程，它包括方法、工具和过程三个要素。  
A. 软件过程                      B. 软件测试  
C. 软件生命周期                      D. 软件工程方法学
- （ A ）意味着一个操作在不同的类中可以有不同的实现方式。  
A. 多态性                      B. 多继承                      C. 类的可复用                      D. 信息隐藏
- 单元测试的测试用例主要根据（ D ）的结果来设计。  
A. 需求分析                      B. 源程序                      C. 概要设计                      D. 详细设计
- 随着软硬件环境变化而修改软件的过程是（ B ）。  
A. 校正性维护                      B. 适应性维护  
C. 完善性维护                      D. 预防性维护
- 在白盒测试技术测试用例的设计中，（ A ）是最弱的覆盖标准。  
A. 语句覆盖                      B. 路径覆盖  
C. 条件组合覆盖                      D. 判定覆盖
- 结构化程序设计的一种基本方法是（ C ）。  
A. 筛选法                      B. 迭代法  
C. 逐步求精法                      D. 递归法
- 不适合作为科学工程计算的程序设计语言是（ D ）。

- A. Pascal                      B. C  
C. Fortran                      D. Prolog

- 只有单重继承的类层次结构是（ C ）层次结构。  
A. 网状型                      B. 星型  
C. 树型                      D. 环型
- 面向对象的开发方法中，（ C ）将是面向对象技术领域内占主导地位的标准建模语言。  
A. Booch 方法                      B. Coad 方法                      C. UML 语言                      D. OMT 方法
- 瀑布模型本质上是一种（ A ）。  
A. 线性顺序模型                      B. 顺序迭代模型  
C. 线性迭代模型                      D. 及早见产品模型
- 软件维护产生的副作用，是指（ C ）。  
A. 开发时的错误                      B. 隐含的错误  
C. 因修改软件而造成的错误                      D. 运行时误操作
- 软件详细设计的主要任务是确定每个模块的（ A ）。  
A. 算法和使用的数据结构                      B. 外部接口  
C. 功能                      D. 编程
- 软件结构图的形态特征能反映程序重用率的是（ C ）。  
A. 深度                      B. 宽度                      C. 扇入                      D. 扇出
- 为了提高模块的独立性，模块内部最好是（ C ）。  
A. 逻辑内聚                      B. 时间内聚  
C. 功能内聚                      D. 通信内聚
- 软件结构图中，模块框之间若有直线连接，表示它们之间存在（ A ）。  
A. 调用关系                      B. 组成关系  
C. 链接关系                      D. 顺序执行关系
- “软件危机”是指（ C ）。  
A. 计算机病毒的出现                      B. 利用计算机进行经济犯罪活动  
C. 软件开发和维护中出现的一系列问题                      D. 人们过分迷恋计算机系统
- 具有风险分析的软件生命周期模型是（ C ）。  
A. 瀑布模型                      B. 喷泉模型  
C. 螺旋模型                      D. 增量模型
- 数据流图（DFD）中的每个数据处理至少需要（ D ）。  
A. 一个输入数据流                      B. 一个输出数据流  
C. 一个输入或者输出数据流                      D. 一个输入数据流和一个输出流

二. 判断题（每小题 2 分，共 24 分）

1. 文档是软件产品的一部分，没有文档的软件就不称其为软件。（ Y ）
2. 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。（ N ）
3. 数据流图建立系统的功能模型，它由数据流、数据处理和数据存贮组成。（ N ）
4. 在结构化分析建模中要建立的是对象模型、功能模型和行为模型，它们以数据字典为核心。（ N ）
5. 好的测试是用少量的测试用例运行程序，发现被测程序尽可能多的错误。（ Y ）
6. 边界值分析方法是取输入/输出等价类的边界值来构成测试用例的测试方法。（ Y ）
7. 在软件开发的过程中，若能推迟暴露其中的错误，则为修复和改正错误所花费的代价就会降低。（ N ）
8. 在需求分析中，分析员要从用户那里解决的最重要的问题是明确软件做什么。（ Y ）
9. 软件需求规格说明书在软件开发中具有重要的作用，是软件可行性分析的依据。（ N ）
10. 面向对象方法学的四要素是对象、类、继承和消息通信。（ Y ）
11. 详细设计的结果基本最终决定了软件的质量。（ Y ）
12. 在详细设计中常采用的工具有层次图、PAD(问题分析图)、程序流程图、盒图等。（ N ）

三. 填空题（每空 2 分，共 24 分）

1. 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。
2. 软件结构是以 模块 为基础而组成的一种控制层次结构。
3. 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 黑盒 进行测试。
4. 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 数值耦合。
5. 软件概要设计关键的一步是 确定软件体系结构，它直接影响下一阶段详细设计和编码的工作。
6. 数据流图中的箭头表示 数据流，椭圆或圆形表示 数据处理，矩形表示 数据源点或终点。
7. 软件开发是一个自顶向下逐步细化和求精过程，而软件测试是一个 自底向上或相反 集成的过程。

8. 按数据流的类型，面向数据流的设计方法有两种设计策略，它们是变换分析设计和 事务分析设计。

9. 数据流图仅反映系统必须完成的逻辑功能，所以它是一种 功能 模型。
10. 软件结构中的 深度 表示的软件的控制层数，它往往能够粗略地标志一个系统的大小和复杂程度。

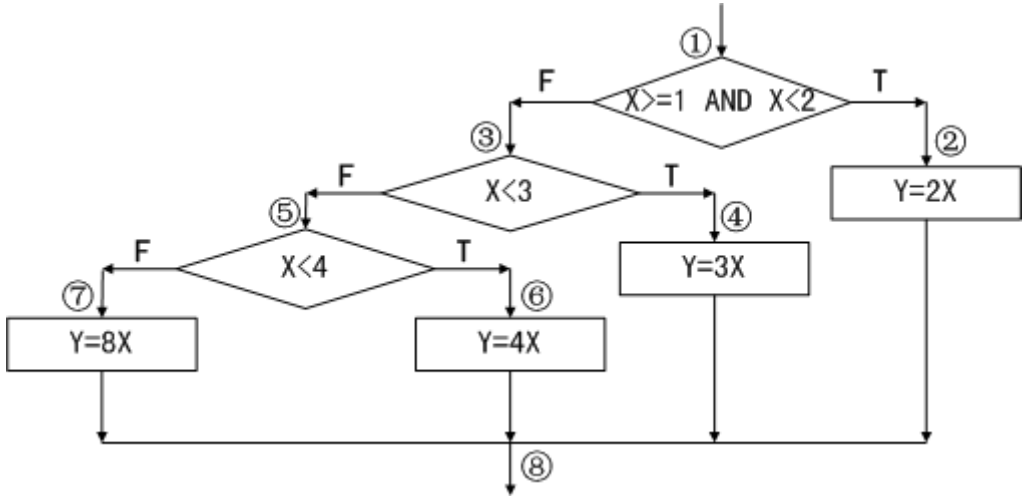
四. 简答题（每小题 5 分，共 10 分）

1. 衡量模块独立性的两个定性标准是什么（2 分）？这两个标准的定义分别是什么（2 分）？在我们的软件设计中，关于模块独立性我们追求的目标是什么（1 分）？
2. 简述结构化程序设计方法的基本特点。（5 分）

五. 应用题（第一小题 8 分，第二小题 12 分，共 20 分）

1. 某旅馆的电话服务如下：（8 分）
- 可以拨分机号码和外线号码。分机号是从 7201 至 7299。外线号码先拨 9，然后是市话号码或长话号码。长话号码是以区号和市话号码组成。区号是从 100 到 300 中任意的数字串。市话号码是以局号和分局号组成，局号可以是 455、466、888、552 中任意一个号码。分局号是任意长度为 4 的数字串。
- 要求：写出在数据字典中，电话号码的数据条目的定义（即组成）。

2. 下图是一个被测程序的流程图，请按照要求完成下列问题。（共 12 分）



（1）根据该程序流程图画出该程序的流图；（3 分）

- (2) 计算该流图的环形复杂度；(3 分)
- (3) 确定独立路径的集合；(3 分)
- (4) 设计测试用例以满足逻辑覆盖的判定覆盖标准。(3 分)

附录：答案及评分标准

一. 单项选择题（每小题 1 分，共 22 分）

- 1. (C) 2. (D) 3. (A) 4. (D) 5. (D)
- 6. (A) 7. (D) 8. (B) 9. (A) 10. (C)
- 11. (D) 12. (C) 13. (C) 14. (A) 15. (C)
- 16. (A) 17. (C) 18. (C) 19. (A) 20. (C)
- 21. (C) 22. (D)

二. 判断题（每小题 2 分，共 24 分）

- 1. 对 2. 错 3. 错 4. 错 5. 对 6. 对
- 7. 错 8. 对 9. 错 10. 对 11. 对 12. 错

三. 填空题（每小题 2 分，共 24 分）

- 1. 输出结果
- 2. 模块
- 3. 黑盒测试方法
- 4. 数值耦合
- 5. 确定软件的体系结构
- 6. 数据流，数据处理，数据的源点/终点
- 7. 自底向上或相反
- 8. 事务分析设计
- 9. 功能
- 10. 深度

四. 简答题（每小题 5 分，共 10 分）

- 1. 衡量模块独立性的两个定性标准是什么（2 分）？这两个标准的定义分别是什么（2 分）？

在我们的软件设计中，关于模块独立性我们追求的目标是什么（1 分）？

答：衡量模块独立性的两个定性标准是内聚和耦合（2 分）。

耦合是指对一个软件结构内不同模块彼此之间互相依赖（连接）的紧密程度；而内聚则标志一个模块内部各个元素彼此结合的紧密程度。（2 分）。

在我们的软件设计中，关于模块独立性我们追求的目标是紧密内聚松散耦合（1 分）。

- 2. 简述结构化程序设计方法的基本特点。（5 分）

答：

- 1). 是尽可能少用 goto 语句的程序设计方法；
- 2). 每个代码块都单入单出；
- 3). 使用控制结构：顺序、选择和循环+Do\_UNTIL 和 Do\_CASE；
- 4). 自顶向下逐步求精。

五. 应用题（每小题 10 分，共 20 分）

- 1. 某旅馆的电话服务如下：（8 分）

可以拨分机号码和外线号码。分机号是从 7201 至 7299。外线号码先拨 9，然后是市话号码或长话号码。长话号码是以区号和市话号码组成。区号是从 100 到 300 中任意的数字串。市话号码是以局号和分局号组成，局号可以是 455、466、888、552 中任意一个号码。分局号是任意长度为 4 的数字串。

要求：写出在数据字典中，电话号码的数据条目的定义（即组成）。

答：

电话号码=[分机号|外线号码]

分机号=7201..7299

外线号码=9+[市话号码|长话号码]

长话号码=区号+市话号码

区号=100..300

市话号码=局号+分局号

局号=[455|466|888|552]

分局号=4{0..9}4

- 2. 略

# 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)

统一过程是一个面向对象且基于网络的程序开发方法论。

2. 项目进度 (Project Schedule)

**项目进度**是对特定项目的软件开发周期的刻画。

3. 重用 (Reuse)

重复采用以前开发的软件系统中具有共性的部件, 用到新的开发项目中去。

4. 项目成本 (Project Cost)

**项目成本**就是指项目消耗和占用资源的数量总和

5. 需求 (requirement)

**需求**, 就是对期望的行为的表达。

6. 耦合 (Coupling)

**耦合**: 两个软件部件之间的相互关联程度

7. 关联 (Association)

两个类之间的连线称为**关联**, 表示类的实体之间的关系。

8. 极限编程 (XP-Extreme Programming)

**极限编程 (XP)** 是一种轻量级的软件开发方法论, 属于敏捷开发方法。XP 从实践中来, 是对实践的总结, 也是经过实践检验的, 其主要特征是要适应环境变化和 demand 变化, 充分发挥开发人员的主动精神

9. 驱动模块 (Component Driver)

驱动模块就是集成测试里面自顶向下测试时的主模块

10.  $\beta$  测试 ( $\beta$  / Beta Test)

客户的测试, 属于试用性质的非正式的验收测试

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

1. ( $\checkmark$ ) 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. ( $\checkmark$ ) 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. ( $\checkmark$ ) 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. ( $\times$ ) 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. ( $\times$ ) 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. ( $\times$ ) 如果测试数据满足了条件覆盖, 则必然满足判定覆盖。
7. ( $\times$ ) 每一个故障 (缺陷) 都会对应于一个失效 (失败)。
8. ( $\times$ ) 设计成功的系统不需维护, 维护意味着系统存在问题。
9. ( $\times$ ) 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. ( $\times$ ) 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\_\_B\_\_。

- 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\_\_B\_\_。

- 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\_\_A\_\_来表述“谁做”。

- A. 角色
- B. 活动
- C. 制品
- D. workflow

4、\_\_D\_\_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; \_\_A\_\_是指子类可以自动拥有父类的全部属性和服务; \_\_C\_\_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

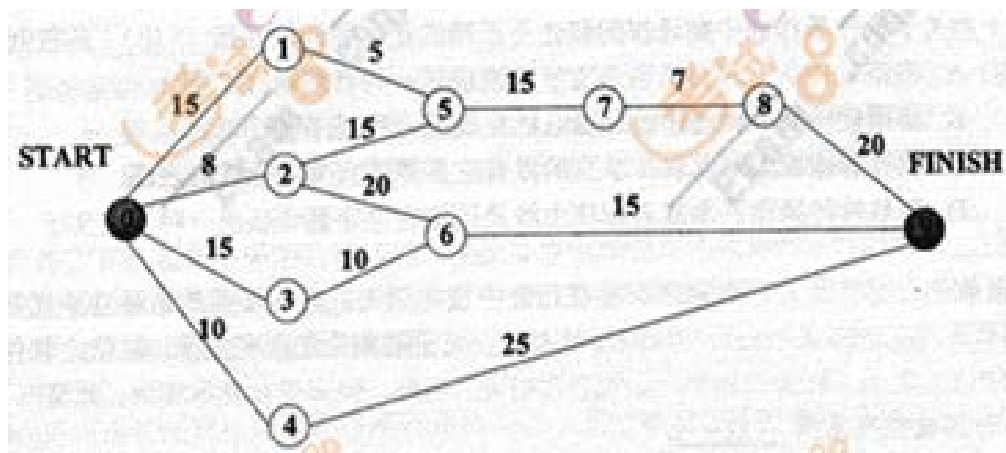
5、风险预测从两个方面评估风险, 即风险发生的可能性以及\_\_D\_\_。

- 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\_\_C\_\_。

- 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\_\_B\_\_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



\_\_\_\_ D \_\_\_\_，用例可以用 \_\_\_\_ A \_\_\_\_ 图来描述。协作图描述了协作的 \_\_\_\_ A \_\_\_\_ 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 \_\_\_\_ C \_\_\_\_ 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 \_\_\_\_ A \_\_\_\_ 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 \_\_\_\_ D \_\_\_\_。

- 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 \_\_\_\_ D \_\_\_\_。

- 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 \_\_\_\_ C \_\_\_\_。

- 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 \_\_\_\_ A \_\_\_\_ 和 \_\_\_\_ B \_\_\_\_ 表现，动态方面由交互图、 \_\_\_\_ D \_\_\_\_ 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 \_\_\_\_ A \_\_\_\_ 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 \_\_\_\_ D \_\_\_\_。

- 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 \_\_\_\_

- 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 \_\_\_\_ C \_\_\_\_。

- 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\_\_\_C\_\_\_。
- 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。

SRS, Design pattern, Acceptance Testing

补充：功能测试：Function Testing

性能测试：Performance Testing

安装测试：Installation Testing

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？

1. 确定目标、可选方案及约束
2. 评估可选方案及风险
3. 计划
4. 开发与测试

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。

瀑布模型最大的问题是它并不能反映实际的代码开发方式

- 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

注意解决的要素、文化差异问题、用户爱好问题

#### 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。

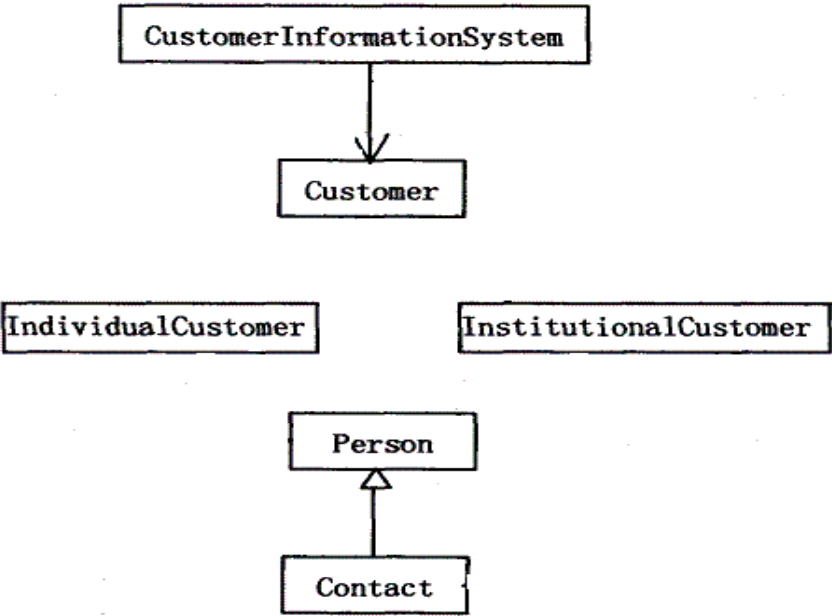
该系统除了可以保存客户信息之外，还具有以下功能：

- (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。



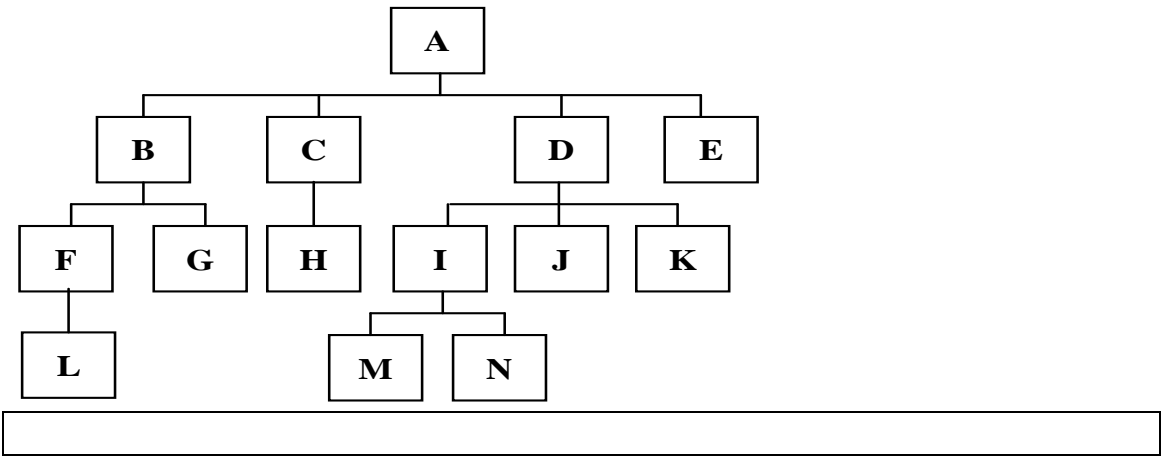
【题目要求】：

- 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程图。

(2) 设计路径覆盖的测试用例。

# 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 × 或 √）（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

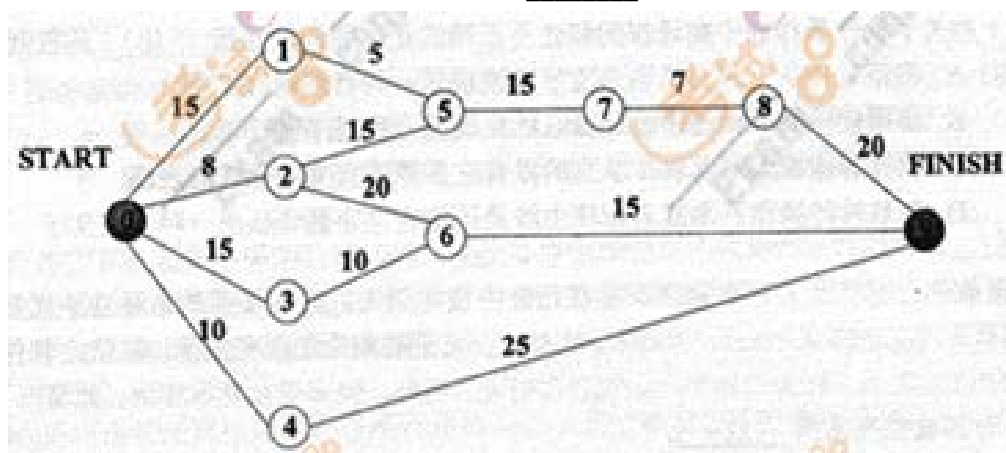
5、风险预测从两个方面评估风险，即风险发生的可能性以及\_\_\_\_\_。

- 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。

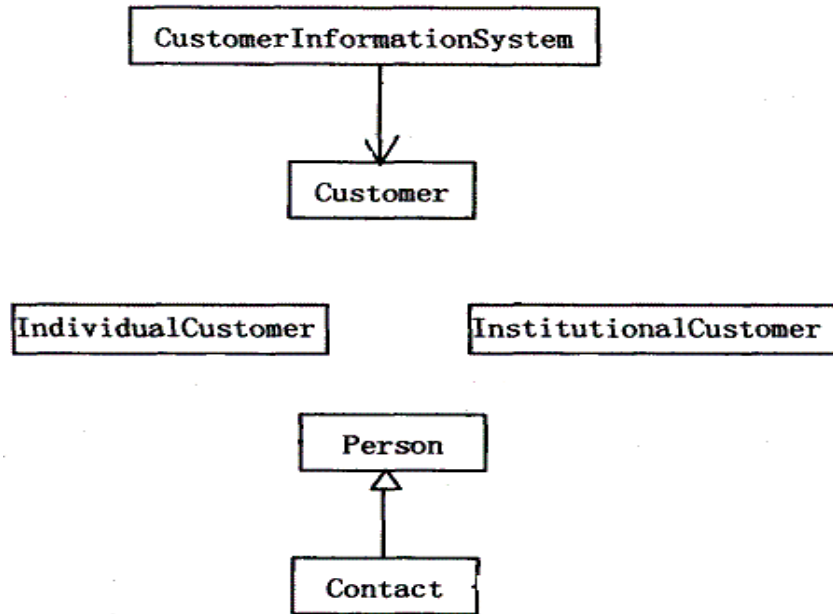
该系统除了可以保存客户信息之外，还具有以下功能：

- （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。

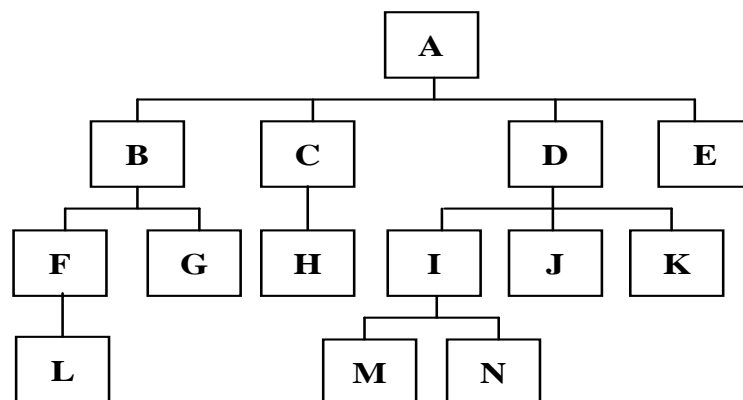


**【题目要求】:**

- 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。

# 英文版《软件工程》教学内容回顾 2016 上

(下述问题仅是课件中的主要部分回顾，问题答案以课件为主要参考)

## Chapter01

### 1. SE 的定义、目的、方法及作用 (P2 / P16)

定义：在将有关软件开发与应用的概念科学化体系化的基础上，研究如何有计划、有效率、经济的开发和利用能在计算机上正确运行的软件的理论和技术的工程方法学，一些开发和维护软件的方法、过程、原则。

目的：在给定成本、进度的前提下，开发出具有适用性、有效性、可修改性、可靠性、可理解性、可维护性、可重用性、可移植性、可追踪性、可互操作性和满足用户需求的软件产品。追求这些目标有助于提高软件产品的质量和开发效率，减少维护的困难。

方向：面向对象模式、结构化模式、基于过程的模式等。

作用：付出较低的开发成本，达到要求的软件功能，取得较好的软件性能，开发的软件易于移植，需要较低的维护费用，能按时完成开发工作，及时交付使用。（是一个系统工程，既有对技术问题的分析与综合，也有对开发过程和参与者的管理）

### 2.//开发模式（paradigm）（P4）

软件开发的全部过程、活动和任务的结构框架，它能直观的表达软件开发全过程，明确要完成的主要活动、任务和开发策略。

### 3.说明错误、缺陷、失败的含义与联系。（请举例说明）（6 页）（44 页习题 3）

错误（error）：当人们在进行软件开发活动过程中出的错（在软件生产过程中人为产生的错误：需求说明的错误、代码中的错误等）

缺陷（fault 故障）：在功能实现过程中产生的问题。（一个缺陷对应多个错误，静态存在）

失败（failure 失效）：指系统违背了它应有的行为。（相对于系统指定行为的偏离：动态存在的）

联系：故障是系统的内部视图，这是从开发人员的角度看待系；而失效是系统的外部视图，它是用户所看到的问题。并非每一个故障都对应于一个失效。

### 4.软件质量应从哪几个方面来衡量？论述之。（9--12 页）

产品的质量：用户从外部特征看软件具有足够的功能并且易于学习和使用，就说明软件具有高质量，开发者从内部特征来看错误数量和类型来判断软件质量的高低。

过程的质量：有很多活动会影响到最终的产品质量。只要有活动出了差错，产品的质量就会受到影响。

商业环境背景下的质量（商业质量）：将技术价值和商业价值统一起来。

### 5.//软件系统的系统组成（P16）

活动和对象：活动（activity）是发生在系统中的某些事情，通常描述为由某个触发器引发的事件，活动通过改变某一特性将一个事物转变成另一个事物。活动中涉及的元素被称为对象（object）或实体（entity）。通常，这些对象以某种方式相互联系。

关系和系统边界。

### 6.现代软件工程大致包含的几个阶段及各个阶段文档（P23-24）

需求分析和定义（项目计划、可行性研究报告、软件需求规格说明书 SRS）

系统设计（系统设计文档如软件结构图）

程序设计（模块功能算法、数据描述文档）

编写程序/程序实现（源程序、注释）

单元测试（由模块测试和性能测试产生的测试报告）

集成测试（按照结构图进行测试产生的测试报告）

系统测试（按照 SRS 对系统总体功能进行测试产生的测试报告）

系统提交（交付的产品，用户手册或操作手册）

系统维护（修改软件的过程，为满足改错或新需求产生的维护报告）

#### **7.使现代 SE 实践发生变化的（七个）关键因素是什么？（28--29 页）**

商用产品投入市场时间的紧迫性；计算技术在经济中的转变：更低的硬件成本，更高的开发、维护成本；功能强大的桌面计算的可用性；广泛的局域网和广域网；面向对象技术的采用及其有效性；使用窗口、图标、菜单和指示器的图形用户界面；软件开发瀑布模型的不可预测性。

#### **8.什么是抽象？（30 页）**

抽象是在某种概括层次上对问题的描述，使得我们能够集中于问题的关键方面而不会陷入细节。

#### **9.什么是软件过程？软件过程的重要性是什么？包含几个阶段？（32 页）（45 页）**

定义：软件开发活动中产生某种期望结果的一系列有序任务，涉及活动、约束和资源。

重要性：

1. 通用性（在一组活动上保持一致性和结构性。一致性和结构性可以使我们知道是否已经做好了工作，还能使别人以同样地方式做工作，因而具有相对通用性）；
2. 指导性（分析、检查、理解、控制、改善活动）。
3. 可以将经验传授给他人。

阶段：需求分析和定义、系统设计、程序设计、编写程序、单元测试、集成测试、系统测试、系统交付、维护。

#### **10.什么是重用等软件工程主要概念？（34 页）**

重（复）用采用以前开发的软件系统中具有共性的部件，用到新的开发项目中去。（这里的重用不仅仅是代码的重用。）

### **Chapter02**

#### **1.什么是软件过程？软件过程的重要性是什么？（P45-46）**

#### **2.瀑布模型及各阶段文档，优缺点？（P49）**

优点：用里程碑和各个阶段的提交文档描述软件开发过程；它的简明性使得人们能够容易的向不熟悉软件开发的顾客清楚地作出解释；是其他模型的基础。

缺点：面临变动时，该模型无法处理实际过程中的重复开发的问题；文档转换有困难。

#### **3.原型的概念（P51）**

原型是一个部分开发的产品，它使客户和开发人员能够对计划开发的系统的相关方面进行检查，以决定它对最终产品是否合适或恰当。

#### **4.论述分阶段开发模型的含义，其基本分类及特点是什么？（56 页）**

含义：系统被设计成部分提交，每次用户只能得到部分功能，其他部分处在开发过程中。

分类：增量式开发（系统需求按照功能分成若干子系统，开始建造的版本是规模小的，部分功能的系统，后续版本添加包含新功能的子系统，最后版本是包含全部功能的子系统集）、迭代式开发（系统开始就提供了整体功能框架，后续版本陆续增强各个子系统，最后版本是各个子系统的功能达到最强）。

特点：培训过程可以很早开始，可以使得开发人员观察用户的反应，为后续版本提供更好的建议；可以为那些从未实现的功能提前开发市场；不断发布新的版本使得开发人员可以快速的修正系统中的问题；不同的开发版本可以针对不同的专业领域。

#### **5.螺旋模型四个象限的任务及四重循环的含义？（P58）**

任务：计划，目标/可选方案，风险评估，开发和测试。

含义：操作概念，软件需求，软件设计，系统测试。

**\*P80--81 页 习题 2、3（针对本章描述的每一种过程模型，讨论使用该模型的优点和缺点分别是什**

么，讨论该模型是如何处理开发后期重要的需求变化的。)

7.//在所有的软件开发过程模型中，你认为哪些过程给予你最大的灵活性以应对需求的变更？（81页习题 11）

阶段开发模型和螺旋模型。

## 8.什么是 UP， RUP?

统一过程（UP）：用例驱动的、以基本架构为中心的、迭代式和增量性的软件开发过程框架。它使用对象管理组织的 UML 并与对象管理组织的软件工程原模型等相兼容。

统一开发过程（RUP）：一个面相对象且基于网络的程序开发方法论。描述了如何有效地利用商业的可靠的方法开发和部署软件。统一开发过程将重复一系列生命期，每个周期包括四个阶段：开始阶段、确立阶段、构建阶段、移交阶段。每个阶段可以进一步划分为多次迭代。其定义了三个支持工序：配置变更管理工序、项目管理工序和环境配置管理工序。定义了六个核心工序：业务模型工序、需求工序、分析设计工序、实现工序、测试工序、部署工序。

## Chapter03

### 1.什么是项目进度？活动？里程碑？（83 页）

项目进度：通过列举项目的各个阶段，把每个阶段分解成离散的任务或活动，来描述特定项目的软件开发周期。进度还描绘这些活动之间的交互，并估算每项任务或活动将花费的时间。因此，进度是一个时间线，说明活动将在什么时候开始、在什么时候结束以及相关的开发产品将在什么时候完成。

活动：项目的一部分，它在一段时间内发生，具有开始和结束。

里程碑：活动的完成——某一特定的时刻，专门指定的活动的结束。

### 2.如何计算软件项目活动图的关键路径？（习题 2，3）冗余时间？最早和最迟开始时间（课堂习题讲解）

### 3.软件团队人员应该具备的能力是什么？（96 页）

完成工作的能力；对工作的兴趣；开发类似应用的经验；使用类似工具或语言的经验；使用类似技术的经验；使用类似开发环境的经验；培训；与其他人交流的能力；与其他人共同承担责任的能力；管理技能。

### 4.软件项目组织的基本结构？（101 页）

主程序员负责制组：

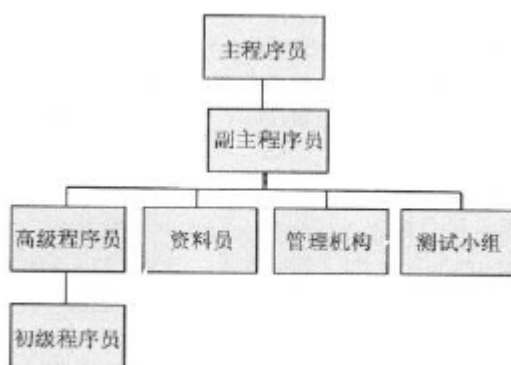


图3-11 主程序员负责制小组的组织结构

主程序员：总体负责系统的设计和开发，其他小组成员向该主程序员汇报，主程序员对每一个决定有最终决策权。主程序员监督所有其他小组成员、设计所有程序、把代码开发分配给其他小组成员。

副主程序员（后备程序员）：在必要时替代主程序员。

资料员：负责维护所有的项目文档，编译和链接代码，并对提交的所有模块进行初步测试。

在有着高度确定性、稳定性、一致性和重复性的项目中，使用像主程序员负责制这样的等级组织结构会更有效。当项目中涉及大量不确定性时，采用更为民主的方法更好。

忘我方法：不是把责任放在单个人身上，而是让每个人平等的担负责任。而且，过程与个人是分开的：批评是针对产品或结果的，并不涉及个人。忘我小组结构是民主式的，不论讨论的是设计问题还是测试技术，小组成员投票产生决策。

## 5.// 专家估算法的大致含义？（106 页），算式估算法的大致含义？（108 页）

专家估算法：工作量估算方法依赖于专家判断，预测的精确性基于估算者的能力、经验、客观性和洞察力。

算法估算法： $E = (a + bS^c)^m(X)$  其中  $S$  是估算的系统规模，而  $a$ 、 $b$  和  $c$  是常量。 $X$  是从  $x_1$  到  $x_n$  的一个成本因素的向量， $m$  是基于这些因素的一个调整因子。

## 6.试述 MIANMO 模型的三个阶段基本工作原理或含义。（111 页）

在阶段 1，项目通常构建原型以解决包含用户界面、软件和系统交互、性能和技术成熟性等方面在内的高风险问题。这时，人们对正在创建的最终产品的可能模型知之甚少，因此用应用点（其创建者对它的命名）来估算规模。

在阶段 2（即早期设计阶段），已经决定将项目开发向前推进，但是设计人员必须研究几种可选的体系结构和操作的概念。同样，仍然没有足够的信息支持准确的工作量和工期估算，但是远比第 1 阶段知道的信息要多。在阶段 2，使用功能点对模型进行测量。

在阶段 3（后体系结构阶段），开发已经开始，而且已经知道了相当多的信息。在这个阶段，可以根据功能点或代码来进行规模估算，而且可以较为轻松的估算很多成本因素。

## 7.什么是软件风险？主要风险管理活动？有几种降低风险的策略？（119、122 页）

软件风险：一种具有负面后果的、人们不希望发生的事件。

风险管理活动：风险评价（识别、分析、优先级分配）；风险控制（降低、管理计划、化解）。

降低风险的策略：通过改变性能或功能需求避免风险；通过把风险分配到其他系统中或者购买保险以便在风险成为事实时拟补经济上的损失从而转移风险；假设风险会发生接受并用项目资源控制风险。

## 8.找出图 3.23 和 3.24（139 页）的关键路径。

## Chapter04

### 1.需求的含义是什么？（143 页）

对来自用户的关于软件系统的期望行为的综合描述，设计系统的对象、状态、约束、功能等。

需求就是对期望的行为的表达。需求处理的是对象或实体，他们可能处于的状态，以及用于改变状态或对象特性的功能。需求阶段的目标是理解客户的问题和需要。因此，需求集中于客户和问题，而不是解决方案和实现。我们通常说，需求指定客户想要什么行为，而不是如何实现这些行为。

### 2.需求作为一个工程，其确定需求的过程是什么？（144 页 图 4.1）

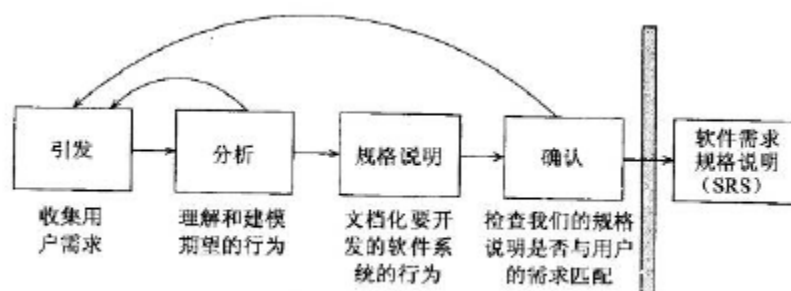


图4-1 获取需求的过程

### 3.举例说明获取需求时，若有冲突发生时，如何考虑根据优先级进行需求分类。（152 页）

（1）绝对要满足的需求（必须的）；（2）非常值得要的但并非必须的需求（值得要的）；（3）可要可不需求（可选的）。

例如，信用卡记账系统必须能够列出最近的费用，将它们加起来并要求在某个日期前支付，这是必须的需求。但是，该记账系统也可能按照购买类型区分费用，以帮助购买者理解购买的模式，这种购买类型分析是值得要的需求，但是可能不是必须的需求。最后，记账系统可能要求用黑颜色来

打印贷方账目而用红颜色打印借方账目，这种需求是有用的，但它是可选的需求。当软件开发项目受到时间或资源的限制时，这种做法也是很有用的。如果系统的成本太高或者开发的时间太长，就可以去掉可选需求，并对值得要的需求进行分析，考虑是去掉还是延迟到后续版本中。

#### **4. //如何使需求变得可测试？（151-152 页, sidebar4.4）**

（1）指定每个副词和形容词的定量描述，这样限定词的含义就清楚、明了了；（2）用特定实体的名称替换代名词；（3）要确保在需求文档的某个地方，准确地定义每个名词。

#### **5.需求文档分为哪两类？（153 页）**

需求定义：客户想要的每一件事情的完整列表。

需求规格说明：将需求重新陈述为关于要构建的系统将如何运转的规格说明。

#### **6.什么是功能性需求和非功能性需求/质量需求？ 设计约束？过程约束？（149 页）**

功能性需求：根据要求的活动来描述需要的行为。定义问题解决方案空间的边界。描述系统内部功能或系统与外界环境的交互作用。

质量需求/非功能性需求：描述一些软件解决方案必须拥有的质量特性。

设计约束：已经做出的设计决策或限制问题解决方案集的设计决策。

过程约束：对用于构建系统的技术和资源的限制。

#### **7.需求的特性？（正确性、一致性、完整性）（155 页）**

正确性、一致性、无二义性、完备性、可行性、相关性、可测试性、可跟踪性。

#### **8.了解 DFD 图的构成及画法（172 页）**

数据流图（DFD）：建模功能以及从一个功能到另一个功能的数据流。一个泡泡表示一个加工或功能，它转换数据。箭头表示数据流，其中进入泡泡的箭头表示其功能性的输入，从泡泡出去的箭头表示其功能的输出。前面步骤的结果流入后面的步骤，单个计算使用之外的持久性数据保存在数据存储中，它是一个正式的库或信息库，表示为两个平行的条。数据源或者数据接收器表示为矩形，称为参与者：提供输入数据或接收输出结果的实体。

#### **9.在需求原型化方面，什么是抛弃型原型？什么是演化型原型？（192--193 页）**

抛弃型原型：是为了对问题或者提议的解决方案有更多的了解而开发的软件，永远不会作为交付软件的一部分。

演化型原型：不仅帮助我们回答问题，而且还要演变为最终的产品。

10// 用 DFD 图简单描述 ATM 机的工作原理（主要功能和数据流）（220 页习题 7）

### **Chapter05**

#### **1.什么是软件体系结构？设计模式？设计公约？设计？概念设计？技术设计？（223-227 页）**

软件体系结构：用以解释如何将系统分解为单元以及这些单元又是如何相互关联，还描述这些单元的所有外部可见特征。

设计模式：一种针对单个软件模块或少量模块而给出的一般性解决方案，它提供相对较低层次的设计决策。

设计公约：一系列设计决策和建议的集合，采用这些设计决策和建议，能够提高系统某方面的设计质量。

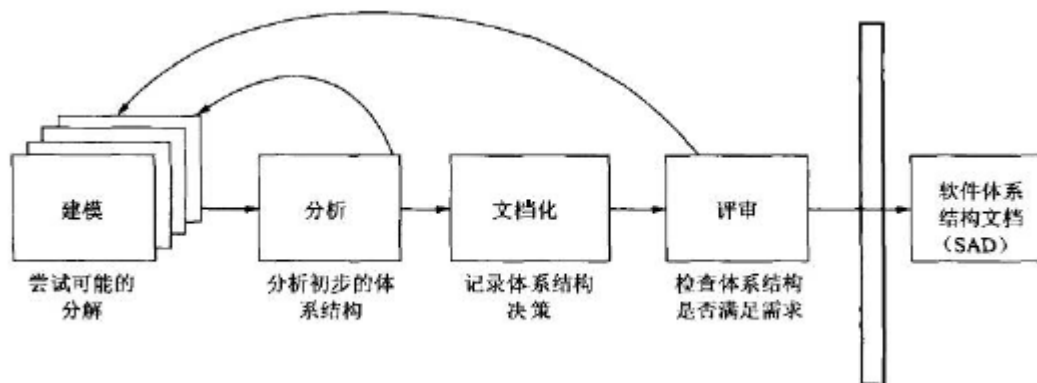
设计：是一种创造性的过程，它考虑如何实现所有的客户需求。设计所产生的计划也称为设计。

概念设计/系统设计：确切地告诉客户系统要做什么。侧重于系统的功能，用顾客理解的语言描述系统功能，通过解释看得见的系统外部特征使顾客理解系统的功能。

技术设计：将概念设计转换为更为详细的文档。描述了系统采用的工艺，用计算机行话和技术术语描述，对技术规格说明的技术性描述。

#### **2.设计过程模型的几个阶段？**





### 3.三种设计层次极其关系？（229 页）

体系结构：将需求规格说明中确定的系统能力与实现这些能力的系统构件关联起来。

代码设计：包含算法和数据结构，并且其构件是编程语言原语、数据操纵原语以及组装机制。

可执行设计：在比代码设计的层次还要低的静态层次处理代码设计，讨论内存分配、数据格式、位模式等。

关系：自顶向下设计：首先设计体系结构，然后进行代码设计，最后是可执行设计。

### 4.什么是模块化？什么是抽象？（238 页）

模块化：模块有清晰的输入和输出，设计目的明确，功能独立，可以做独立检测。

抽象：对细节的隐藏。

### 5.论述设计用户界面应考虑的问题。（242 页）

关键要素：隐喻（可以认识和学习的基本术语、图像和概念）、头脑中的模型（数据、功能、任务和角色的组织和表示）、模型的导航规则（如何在数据、功能、活动和角色中移动）、外观（系统向用户传输信息的外观特性）、感觉（向用户提供有吸引力的体验的交互技术）。

文化问题

用户偏爱

### 6.5.5 节----模块独立性----耦合与内聚的概念及各个层次划分？（248----xxx 页）

耦合度：两个部件之间的相互关联的程度。

分为：非直接耦合（模块之间没有信息传递）

数据耦合（模块之间传递的是数据）

特征耦合（模块之间传递的是数据结构）

控制耦合（模块之间传递的是控制量）

公共耦合（不同模块访问公共数据）

内容耦合（一个模块直接修改另一个模块）。

内聚度：构成构件内部的粘合程度。

分为：偶然性内聚（不相关的功能、过程、数据等出现在同一个部件中）

逻辑性内聚（逻辑上相关或相似的功能或数据放置在同一个部件中）

时间性内聚（部件各个部分要求在同一时间完成）

过程性内聚（各部分由特定的次序）

通讯性内聚（各部分访问共享数据）

顺序性内聚（各部分有输入输出关系）

功能性内聚（各部分组成单一功能）。

### 7.举例说明耦合与内聚的基本分类。以及各个分类的含义与特征（284 页习题 4，5）

## 1.什么是面向对象？（286 页）OO 有几个基本特征？如何使用高级语言实现这些基本特征？

面向对象：是一种软件开发方法，他将问题和问题的解决方案组织为离散对象的集合，数据结构和行为都包含在对象的表示中。

7 个基本特征及实现：标识（将数据组织为离散的、可区别的实体，称为对象。单个对象具有相关联的状态和行为。）、抽象（OO 系统中的抽象有助于表示正在开发的系统中的不同观点。合起来，这些抽象形成一个层次，以说明不同的系统观点是如何彼此相关的。）、分类（将具有共同属性和行为的对象分组。）、封装（类封装对象的行为和属性，隐藏实现细节。）、继承（一开始先广义的定义一个类，然后将他们细化为更具体的子类。某个子类可以继承父类的结构、行为以及属性。）、多态（同样的行为在不同点类或子类中会有不同的表现。）和持久性（对象的名称、状态和行为超越时间或空间的能力，换句话说，当转换对象时，保存对象的名称、状态和行为。）。

## 2.使用高级语言的 OO 基本编程方法和技巧。（286-291）

### 3.什么是设计模式？

模式：是抽象体系结构元素的模版，可以用来指导生成设计。

设计模式：是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。使用设计模式是为了可重用代码、让代码更容易被他人理解、保证代码可靠性。

### 4.OO 设计的基本原则？

开闭原则、里氏代换原则、依赖倒转原则、接口隔离原则、笛米特法则、合成/聚合复用原则。

### 5.OO 开发有何优势？（291 页）

语言的一致性：可以使用同样的术语描述问题及其解决方案：类、对象、方法、属性和行为。

过程的一致性：保持术语和视角的一致性，所有过程使用同样的术语，便于测试需求。使用封装的数据和行为，组成独立的单元。

### 6.OO 开发过程有几个步骤？（292 页）

需求、设计、编码和测试。（需求分析、高层设计、低层设计、面相对象的程序设计、测试）

### 7.熟悉用例图的组成和画法，用例的几个要素的含义，掌握用例图的实例解析方法（294 页）

用例：通过建模用户、外部系统或其他外部实体与要开发的系统之间的对话，描述系统应该具有或展现的特定功能。

要素：参与者（与系统交互的实体，可以是某个用户、某个设备或其他系统）、用例（对系统某些方面的功能性描述，对参与者可见，参与者通过这些用例反映自己对系统的观点）、扩充（扩展用例，以说明不同的或更深层次的视角）和使用（实际上对已定义用例的复用）。

### 8.用例图、类图等对面向对象的项目开发的意义是什么？

对功能的完整描述；便于用户、设计者、测试者之间的交流；是系统分析中更多正是建模的基础。

### 9.熟悉类图中各个类之间的基本关系分类（303-305）

泛化：在一个继承关系中，超类泛化子类。

关联：两个类一起出现，并且它们之间的关系必须保持一段时间。

聚合：一个类是另外一个类的一部分。松散的部分与整体的关系。聚合关系中的成员对象是整体对象的一部分，但是成员对象可以脱离整体对象存在（此时不会影响整体对象的定义，

组装：强的部分与整体的关系。组合关系中整体对象可以控制成员对象的生命周期，一旦整体对象不存在，成员对象也将不存在，两者是共生关系。

依赖：一个项的定义发生改变，则会引起另外一个项的改变变化。

### 10.熟悉类图等的组成和画法（300-308 页）

### 11.知道 UML 其他图示结构的基本用途。

## Chapter07

### 1.//为什么说编码工作是纷繁复杂甚至令人气馁？（337 页）

第一，设计人员可能没有处理平台和编程环境的所有特性。易于用图表描述的结构和关系并不是总能够直截了当地编写成代码。第二，我们必须用这样一种方式编写代码：不仅要在再次使用代码进行测试的时候便于自己理解，而且当系统随着时间演化时，也便与他人理解。第三，在创建易于复用的代码的同时，还必须利用这些特性：设计的组织结构、数据结构、程序设计语言的概念。

### 2.//一般性的编程原则应该从哪三个方面考虑？（340-344 页）

控制结构：将设计转换成代码时，我们希望能够保留构件的控制结构。在隐含调用和面相对象设计中控制是基于系统状态和变量发生的变化。程序的结构能够反映出设计的控制结构。模块化编程，强调可读性。

算法：程序设计通常会指定一类算法，用于编写某些构件。注意性能效率所花费的代价，在性能与其他质量因素之间进行平衡。

数据结构：编写程序时，应该安排数据的格式并存储数据，使得数据管理和操纵更为直观。

### 3.//论述编码阶段实现某种算法时所涉及的问题。（342 页）

使代码更快运行可能会伴随一些隐藏的代价，不要牺牲代码的清晰度和正确性，必须了解所使用的编译器是如何优化代码的。

### 4.在编写程序内部文档时，除了 HCB 外，还应添加什么注释信息？（352-354 页）

其他程序注释、有意义的变量名和语句标记、安排格式以增强理解、文档化数据。

### 5.什么是极限编程(XP)? 以及派对编程? (357 页)

极限编程 (XP)：是一种轻量级的软件开发方法论，属于敏捷开发方法。XP 从实践中来，是对实践的总结，也是经过实践检验的，其主要特征是要适应环境变化和 demand 变化，充分发挥开发人员的主动精神。XP 承诺降低软件项目风险，改善业务变化的反应能力，提高开发期间的生产力，为软件开发过程增加乐趣等等。两类参与者：客户：定义将要实现的系统之特征，描述测试计划，分配系统实现和测试的优先级；程序员：将客户的上述诉求予以编程实现。

派对编程：属于主要的敏捷开发方法，其开发方式是两个程序员共同开发程序，且角色分工明确。一个负责编写程序，另一个负责复审与测试。两人定期交换角色。

## Chapter08

### 1.// 产生软件缺陷的原因？（365 页）

规格说明可能是错误的，或者遗漏了某个需求；读于指定的硬件和软件，规格说明中可能包含不可能实现的需求；系统设计中可能包含故障；程序设计中可能包含故障；程序代码可能是错误的。

### 2.//将软件缺陷进行分类的理由？（367 页）

在编码完程序构件之后，我们通常要对代码进行检查，以找出故障并立刻去除它们。当不存在明显的故障时，我们就测试程序，创造一些条件，使代码不能像设计的那样做出反应，看一看能否发现更多的故障。

### 3.几种主要的缺陷类型？（367-368 页）

算法故障：由于处理步骤中的某些错误，使得对于给定的输入，构件的算法或逻辑没有产生适当的输出。

计算故障/精度故障：一个公式的实现是错误的，或者计算结果没有达到要求的精度。

文档故障：文档与程序实际做的事情不一致。

压力故障/过载故障：填充数据结构时超过了它们规定的容量。

能力故障/边界故障：系统活动到达指定的极限时，系统性能会变得不可接受。

计时故障/协调故障：在开发实时系统时，一个关键的考虑因素是几个同时执行的或按仔细定义的顺序执行的进程之间的协调问题，当协调这些事件的代码不适当时，会出现计时故障或协调故障。

吞吐量故障/性能故障：系统不能以需求规定的速度执行。

恢复故障：当系统遇到失效时，不能表现得像设计人员希望的或客户要求的那样。

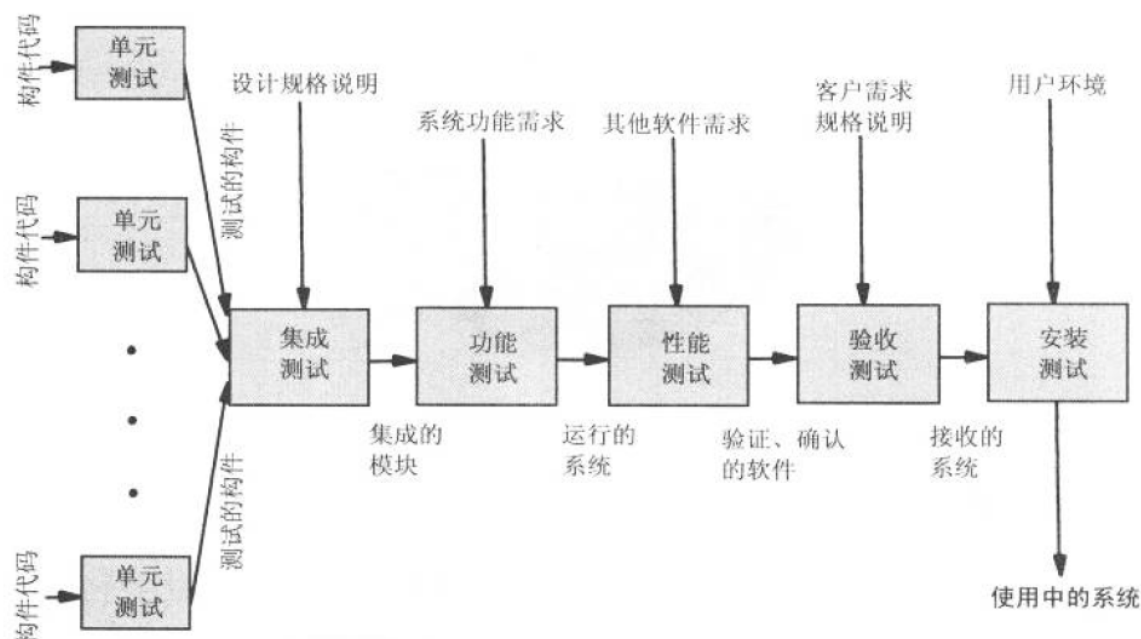
硬件和系统软件故障：提供的硬件和系统软件实际上并没有按照文档中的操作条件和步骤运作。

标准和过程故障：代码没有遵循组织机构的标准和过程。

#### 4.//什么是正交缺陷分类？（369 页）

一种故障跟踪方法，分类方案必须是产品和组织无关的，关键特征之一就是正交性（被分类的任何一项只属于一个类别）

#### 5.测试的各个阶段及其任务？（372 页图 8.3）



模块测试/构件测试/单元测试：将每个程序构件与系统中其他构件隔离，对其本身进行测试。

集成测试：验证系统构件是否能够按照系统和设计规格说明中描述的那样共同工作的过程。

功能测试：对系统进行评估，以确定集成的系统是否确实执行了需求规格说明中描述的功能。

性能测试：将系统与软件和硬件需求的剩余部分进行比较。

验收测试：根据客户的需求描述对系统进行检查。

安装测试：确保系统将按照它应该的方式来运行。

#### 6.// 测试的态度问题？（为什么要独立设置测试团队？）（373 页）

新程序员不习惯将测试看作是一个发现的过程。用户要确保系统在所有条件下都适当运行。忘我编程的态度：把程序看作一个更大系统的构件，而不是那些编写他们的人的财产。

#### 7.测试的方法---黑盒、白盒的概念？（374）

黑盒测试：从外部观察测试对象，将其看作是一个不了解其内容的闭盒/黑盒，测试就是向闭盒提供输入数据并记录产生的输出。免于受强加给测试对象内部结构和逻辑的约束，不可能完备测试。

白盒测试：将测试对象看成一个透明盒，根据测试对象的结构用不同的方式来进行测试，可以设计执行构建内所有语句或所有控制路径的测试用例，以确定测试对象是否是适当运作的。

#### 8.什么是单元测试？ 什么是走查和检查？（376 页）

单元测试：通过通读程序对代码进行检查，试着找出算法、数据以及愈发中的故障。甚至可以将代码与规格说明进行比较，与设计进行比较，以确保已经考虑了所有相关情况。接着，编译代码，排除任何剩余的语法故障。最后，开发测试用例，以证明是否将输入适当的转换成了所期望的输出。

走查：程序员向评审小组提交代码及其相关文档，然后评审小组评论它们的正确性，非正式的，注意力集中在代码上，发现故障但不必修改它们。

检查：评审小组按照一个事先准备好的关注问题清单来检查代码和文档。正式的。

### 9.黑盒白盒方法各自的分类？测试用例的设计和给出方法（结合补充材料）

黑盒测试分类：等价分类法、边界值分析法、错误猜测法、因果图法。

白盒测试分类：逻辑覆盖法、路径覆盖法。

### 10.黑盒白盒方法的分类，各种覆盖方法等。（课件和补充课件）

### 11.考虑如何面对一个命题，设计和给出测试用例的问题。（课件）-课堂练习的测试题目和讲解内容

### 12.集成测试及其主要方法的分类？（390-392）（驱动，桩的概念）

集成测试：将单个构件组合成一个运转的系统进行测试。

分类：自底向上集成（通过合并构件来测试较大型系统。每一个处于系统层次中最低层的构件首先被单独测试，接着要测试的是那些调用了前面已测构件的构件。反复采用此方法，知道所有的构件都被测试完毕。）、自顶向下集成（顶层构件通常是一个控制构件，是独立进行测试的。然后将被测构件调用的所有构件组合起来，作为一个更大的单元进行测试。重复执行这种方法，直到所有的构件都被测试。）

构件驱动程序：调用特定构件并向其传递测试用例的程序。

桩：一种专用程序，用于模拟（测试时）缺少构件时的活动。

### 13.//传统测试和 OO 测试有何不同？OO 测试有何困难？（398-399 页）

区别：传统测试当系统改变时，需要新老测试用例（回归测试）；OO 测试必须对重载的子类进行测试，可能会使用不同的测试用例，单元测试比较简单，但是集成测试比较广泛。

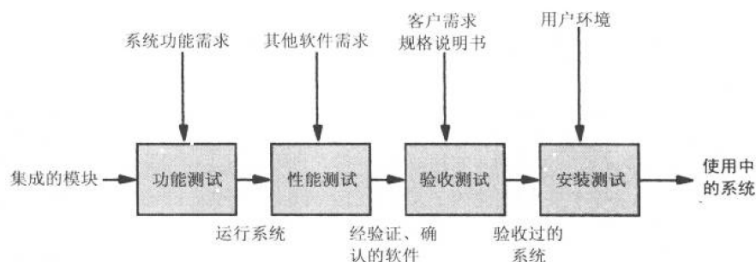
困难：没有什么工具支持表述为对象和方法的需求的验证；帮助测试用例生成的大多数工具并不能处理用对象和方法表述的模型；大多数源代码测度的定义是针对过程代码而不是针对对象和方法；代码覆盖测度和工具在面相对象测试中的作用要比在传统测试中的作用小。

### 14.// 测试计划涉及的几个步骤？（400 页）

制定测试目标、设计测试用例、编写测试用例、测试测试用例、执行测试、评估测试结果。

## Chapter09

### 1.系统测试的主要步骤及各自含义？（420 页，图 9.2）



功能测试：检查集成的系统是否按照需求中指定的那样执行它的功能。

性能测试：将集成的构件与非功能系统需求进行比较。

验收测试：向客户保证构件的系统就是客户想要的系统。

安装测试：使用户能够执行系统功能并记录在实际环境中可能引起的其他问题。XIANG

### 2.//什么是系统配置？软件配置管理？基线？（423 页）（或见课件）

系统配置：向特定客户交付的一组系统构件

软件配置管理：开发和测试不同的配置需要配置管理。配置管理控制不同系统配置之间的差别，将风险和错误减少到最低程度。

基线：是指软件文档和其他资料的集合，它们代表了产品在某一时间点的情况(以及其他参考点)。

### 3.什么是回归测试？（425 页）

用于识别在改正当前故障的同时可能引入的新故障，用于新的版本或发布的一种测试，以验证与旧版本或发布相比，它是否仍然以同样的方式执行相同的功能。

#### **4.功能测试的含义极其作用？（430 页）**

含义：对需求规格说明书中的功能描述进行测试。

作用：相对于一个大的构件集，在一个小的构件集中更容易找出问题的原因。

#### **5.功能测试的基本指导原则？（431）**

具有很高的故障检测概率；使用独立于测试人员和程序员的测试小组；了解期望的动作和输出；既要测试合法输入，也要测试不合法输入；永远不要为了使测试更加容易而去修改系统；制定停止测试标准。

#### **6.性能测试的含义与作用？（436 页）**

含义：测试非功能需求。

作用：测试功能执行的方式，根据客户规定的性能目标来测量，这些性能目标表示非功能需求。

#### **7.性能测试的主要分类？（436 页）**

压力测试：当系统在短时间内达到压力极限时，对系统进行的测试。

容量测试：强调的是处理系统中的大量数据。

配置测试：分析需求中指定的各种软件和硬件配置。

兼容性测试：检查接口功能是否按照需求执行。

回归测试：保证新系统的表现至少与老系统一样好。

安全性测试：确保安全性需求得到满足。

计时测试：评估涉及对用户的响应时间和一个功能的执行时间的相关需求。

环境测试：考察系统在安装场所的执行能力。

质量测试：评估系统的可靠性、可维护性和可用性。

恢复测试：强调的是系统对出现故障或丢失数据、电源、设备或服务时的反应。

维护测试：为了帮助人们发现问题的根源提供诊断工具和过程的需要。

文档测试：确保已经编写了必需的文档。

人为因素测试：检查设计系统用户界面的需求。

#### **8.// 什么是可靠性、可用性和可维护性？（438 页）**

可靠性：指一个系统对于给定的时间间隔内、在给定的条件下无失效运作的概率。

可用性：指在给定的时间点上，一个系统能够按照规格说明正确运作的概率。

可维护性：在给定的使用条件下，在规定的时间内，使用规定的过程和资源完成维护活动的概率。

#### **9.确认测试，确认测试分类？（基准测试和引导测试）（447-448 页）**

确认测试：在确信软件开发的初始阶段系统满足了所有指定的需求后让用户对系统进行的测试。

分类：

基准测试：顾客准备测试用例，这些测试用例代表了系统实际安装投入运行的典型条件，顾客针对每一个测试用例评价系统的性能。

引导测试：在试验的基础上安装该系统，它依赖日常的工作来测试所有的功能。

#### **10.什么是 alpha 测试？β 测试？（448 页）**

α 测试：在向客户发布一个系统之前，先让来自自己组织机构或公司的用户来测试这个系统。

β 测试：客户的试验。

#### **11.什么是安装测试？（450 页）**

在用户的环境下测试，解决由于开发环境和用户环境的不同而引起的错误。集中于两件事：安装系统的完备性；验证任何可能受场所条件影响的功能和非功能特新。

注意：

每一章节的开头中，大的概念性问题是怎样引入的？，其讨论请见课件。

试卷答题须知：

1. 软件工程课程出题覆盖范围比较广泛，考察手段有灵活和多样化特点。
2. 有考察学生软件工程运用能力的题目存在，难题占的分数很少。其他很多是送分的题目。
3. 有难度的题目占极少部分，可以根据时间安排，先回答别的题目。
4. 所有题目都要写到试卷的指定位置，以免流水阅卷时有遗漏。

# 英文版《软件工程》教学内容回顾 2015 上

(下述问题仅是课件中的主要部分回顾，问题答案以课件为主要参考)

## Chapter01

SE 的定义、目的、方法及作用 (P2 / P16)

//开发模式 (paradigm) (P4)

说明错误、缺陷、失败的含义与联系。(请举例说明) (6 页) (44 页习题 3)

软件质量应从哪几个方面来衡量? 论述之。(9--12 页)

//软件系统的系统组成 (P16)

现代软件工程大致包含的几个阶段及各个阶段文档 (P23-24)

使现代 SE 实践发生变化的 (七个) 关键因素是什么? (28--29 页)

什么是抽象? (30 页)

什么是软件过程? 软件过程的重要性是什么? 包含几个阶段? (32 页) (45 页)

什么是重用等软件工程主要概念? (34 页)

## Chapter02

什么是软件过程? 软件过程的重要性是什么? (P45-46)

瀑布模型及各阶段文档, 优缺点? (P49)

原型的概念 (P51)

论述分阶段开发模型的含义, 其基本分类及特点是什么? (56 页)

螺旋模型四个象限的任务及四重循环的含义? (P58)

P80--81 页 习题 2, 3。

//在所有的软件开发过程模型中, 你认为哪些过程给予你最大的灵活性以应对需求的变更? (81 页习题 11)

什么是 UP, RUP?

## Chapter03

什么是项目调度? 活动? 里程碑? (83 页)

如何计算软件项目活动图的关键路径? (习题 2, 3) 冗余时间? 最早和最迟开始时间 (课堂习题讲解)

软件团队人员应该具备的能力是什么? (96 页)

软件项目组织的基本结构? (101 页)

// 专家估算法的大致含义? (106 页), 算式估算法的大致含义? (108 页)

试述 COCOMO 模型的三个阶段基本工作原理或含义。(111 页)

什么是软件风险? 主要风险管理活动? 有几种降低风险的策略? (119、122 页)

找出图 3.23 和 3.24 (139 页) 的关键路径。

## Chapter04



需求的含义是什么? (143 页)

需求作为一个工程, 其确定需求的过程是什么? (144 页 图 4.1)

举例说明获取需求时, 若有冲突发生时, 如何考虑根据优先级进行需求分类。(152 页)

//如何使需求变得可测试? (151-152 页, sidebar4.4)

需求文档分为哪两类? (153 页)

什么是功能性需求和非功能性需求/质量需求? 设计约束? 过程约束? (149 页)

需求的特性? (正确性、一致性、完整性) (155 页)

了解 DFD 图的构成及画法 (172 页)

在需求原型化方面, 什么是抛弃型原型? 什么是演化型原型? (192--193 页)

// 用 DFD 图简单描述 ATM 机的工作原理 (主要功能和数据流) (220 页习题 7)

## Chapter05

什么是软件体系结构? 设计模式? 设计公约? 设计? 概念设计? 技术设计? (223-224 页)

设计过程模型的几个阶段?

三种设计层次及其关系? (229 页)

什么是模块化? 什么是抽象? (238 页)

论述设计用户界面应考虑的问题。(242 页)

5.5 节----模块独立性----耦合与内聚的概念及各个层次划分? (248----xxx 页)

举例说明耦合与内聚的基本分类。以及各个分类的含义与特征 (284 页习题 4, 5)

## Chapter06

//什么是面向对象? (286 页) OO 有几个基本特征? 如何使用高级语言实现这些基本特征? 了解并使用高级语言的 OO 基本编程方法和技巧。(286-291)

什么是设计模式?

//OO 设计的基本原则?

OO 开发有何优势? (291 页)

OO 开发过程有几个步骤? (292 页)

熟悉用例图的组成和画法, 用例的几个要素的含义, 掌握用例图的实例解析方法 (294 页)

用例图、类图等对面向对象的项目开发的意义是什么?

熟悉类图中各个类之间的基本关系分类 (303-305)

熟悉类图等的组成和画法 (300-308 页)

了解 UML 其他图示结构的基本用途。

## Chapter07

//为什么说编码工作是纷繁复杂甚至令人气馁? (337 页)

一般性的编程原则应该从哪三个方面考虑? (340-344 页)

//论述编码阶段实现某种算法时所涉及的问题。(342 页)

在编写程序内部文档时, 除了 HCB 外, 还应添加什么注释信息? (352-354 页)

什么是极限编程(XP)? 以及派对编程? (357 页)

## Chapter08

// 产生软件缺陷的原因？（365 页）  
//将软件缺陷进行分类的理由？（367 页）  
几种主要的缺陷类型？（367-368 页）  
什么是正交缺陷分类？（369 页）  
测试的各个阶段及其任务？（372 页图 8.3）  
// 测试的态度问题？（为什么要独立设置测试团队？）（373 页）  
测试的方法----黑盒、白盒的概念？（374）  
什么是单元测试？什么是走查和检查？（376 页）  
黑盒白盒方法各自的分类？测试用例的设计和给出方法（结合补充材料）  
黑盒白盒方法的分类，各种覆盖方法等。（课件和补充课件）  
考虑如何面对一个命题，设计和给出测试用例的问题。（课件）  
-----课堂练习的测试题目和讲解内容  
集成测试及其主要方法的分类？（390-392）（驱动，桩的概念）  
传统测试和 OO 测试有何不同？OO 测试有何困难？（398-399 页）  
// 测试计划涉及的几个步骤？（400 页）（了解）

## Chapter09

系统测试的主要步骤及各自含义？（420 页，图 9.2）  
//什么是系统配置？软件配置管理？基线？（423 页）（或见课件）  
// 什么是回归测试？（425 页）  
功能测试的含义及其作用？（430 页）  
功能测试的基本指导原则？（431）  
性能测试的含义与作用？（436 页）  
性能测试的主要分类？（436 页）  
// 什么是可靠性、可用性和可维护性？（438 页）  
确认测试，确认测试分类？（基准测试和引导测试）（447-448 页）  
什么是  $\alpha$  测试？ $\beta$  测试？（448 页）  
什么是安装测试？（450 页）

### 注意：

每一章节的开头中，大的概念性问题是引入的？，其讨论请见课件。

### 试卷答题须知：

1. 软件工程课程出题覆盖范围比较广泛，考察手段有灵活和多样化特点。
2. 有考察学生软件工程运用能力的题目存在，难题占的分数很少。其他很

多是送分的题目。

3. 有难度的题目占极少部分，可以根据时间安排，先回答别的题目。

4. 所有题目都要写到试卷的指定位置，以免流水阅卷时有遗漏。

# 英文版《软件工程》教学内容回顾 2016 上

(下述问题仅是课件中的主要部分回顾，问题答案以课件为主要参考)

## Chapter01

SE 的定义、目的、方法及作用 (P2 / P16)

//开发模式 (paradigm) (P4)

说明错误、缺陷、失败的含义与联系。(请举例说明) (6 页) (44 页习题 3)

软件质量应从哪几个方面来衡量? 论述之。(9--12 页)

//软件系统的系统组成 (P16)

现代软件工程大致包含的几个阶段及各个阶段文档 (P23-24)

使现代 SE 实践发生变化的 (七个) 关键因素是什么? (28--29 页)

什么是抽象? (30 页)

什么是软件过程? 软件过程的重要性是什么? 包含几个阶段? (32 页) (45 页)

什么是重用等软件工程主要概念? (34 页)

## Chapter02

什么是软件过程? 软件过程的重要性是什么? (P45-46)

瀑布模型及各阶段文档, 优缺点? (P49)

原型的概念 (P51)

论述分阶段开发模型的含义, 其基本分类及特点是什么? (56 页)

螺旋模型四个象限的任务及四重循环的含义? (P58)

P80--81 页 习题 2, 3。

//在所有的软件开发过程模型中, 你认为哪些过程给予你最大的灵活性以应对需求的变更? (81 页习题 11)

什么是 UP, RUP?

## Chapter03

什么是项目调度? 活动? 里程碑? (83 页)

如何计算软件项目活动图的关键路径? (习题 2, 3) 冗余时间? 最早和最迟开始时间 (课堂习题讲解)

软件团队人员应该具备的能力是什么? (96 页)

软件项目组织的基本结构? (101 页)

// 专家估算法的大致含义? (106 页), 算式估算法的大致含义? (108 页)

试述 COCOMO 模型的三个阶段基本工作原理或含义。(111 页)

什么是软件风险? 主要风险管理活动? 有几种降低风险的策略? (119、122 页)

找出图 3.23 和 3.24 (139 页) 的关键路径。

## Chapter04

需求的含义是什么? (143 页)

需求作为一个工程, 其确定需求的过程是什么? (144 页 图 4.1)

举例说明获取需求时, 若有冲突发生时, 如何考虑根据优先级进行需求分类。(152 页)

//如何使需求变得可测试? (151-152 页, sidebar4.4)

需求文档分为哪两类? (153 页)

什么是功能性需求和非功能性需求/质量需求? 设计约束? 过程约束? (149 页)

需求的特性? (正确性、一致性、完整性)(155 页)

了解 DFD 图的构成及画法 (172 页)

在需求原型化方面, 什么是抛弃型原型? 什么是演化型原型? (192--193 页)

// 用 DFD 图简单描述 ATM 机的工作原理 (主要功能和数据流) (220 页习题 7)

## Chapter05

什么是软件体系结构? 设计模式? 设计公约? 设计? 概念设计? 技术设计? (223-224 页)

设计过程模型的几个阶段?

三种设计层次及其关系? (229 页)

什么是模块化? 什么是抽象? (238 页)

论述设计用户界面应考虑的问题。(242 页)

5.5 节----模块独立性----耦合与内聚的概念及各个层次划分? (248----xxx 页)

举例说明耦合与内聚的基本分类。以及各个分类的含义与特征 (284 页习题 4, 5)

## Chapter06

//什么是面向对象? (286 页) OO 有几个基本特征? 如何使用高级语言实现这些基本特

//征? 了解并使用高级语言的 OO 基本编程方法和技巧。(286-291)

什么是设计模式?

OO 设计的基本原则?

OO 开发有何优势? (291 页)

OO 开发过程有几个步骤? (292 页)

熟悉用例图的组成和画法, 用例的几个要素的含义, 掌握用例图的实例解析方法 (294 页)

用例图、类图等对面向对象的项目开发的意义是什么?

熟悉类图中各个类之间的基本关系分类 (303-305)

熟悉类图等的组成和画法 (300-308 页)

知道 UML 其他图示结构的基本用途。

## Chapter07

//为什么说编码工作是纷繁复杂甚至令人气馁? (337 页)

//一般性的编程原则应该从哪三个方面考虑? (340-344 页)

//论述编码阶段实现某种算法时所涉及的问题。(342 页)

在编写程序内部文档时, 除了 HCB 外, 还应添加什么注释信息? (352-354 页)

什么是极限编程(XP)? 以及派对编程? (357 页)

## Chapter08

// 产生软件缺陷的原因？（365 页）  
//将软件缺陷进行分类的理由？（367 页）  
几种主要的缺陷类型？（367-368 页）  
//什么是正交缺陷分类？（369 页）  
测试的各个阶段及其任务？（372 页图 8.3）  
// 测试的态度问题？（为什么要独立设置测试团队？）（373 页）  
测试的方法----黑盒、白盒的概念？（374）  
什么是单元测试？ 什么是走查和检查？（376 页）  
黑盒白盒方法各自的分类？测试用例的设计和给出方法（结合补充材料）  
黑盒白盒方法的分类，各种覆盖方法等。（课件和补充课件）  
考虑如何面对一个命题，设计和给出测试用例的问题。（课件）  
-----课堂练习的测试题目和讲解内容  
集成测试及其主要方法的分类？（390-392）（驱动，桩的概念）  
//传统测试和 OO 测试有何不同？OO 测试有何困难？（398-399 页）  
// 测试计划涉及的几个步骤？（400 页） （了解）

## Chapter09

系统测试的主要步骤及各自含义？（420 页，图 9.2）  
//什么是系统配置？软件配置管理？基线？（423 页）（或见课件）  
什么是回归测试？（425 页）  
功能测试的含义及其作用？（430 页）  
功能测试的基本指导原则？（431）  
性能测试的含义与作用？（436 页）  
性能测试的主要分类？（436 页）  
// 什么是可靠性、可用性和可维护性？（438 页）  
确认测试，确认测试分类？（基准测试和引导测试）（447-448 页）  
什么是  $\alpha$  测试？ $\beta$  测试？（448 页）  
什么是安装测试？（450 页）

**注意：**

每一章节的开头中，大的概念性问题是引入的？，其讨论请见课件。

**试卷答题须知：**

1. 软件工程课程出题覆盖范围比较广泛，考察手段有灵活和多样化特点。
2. 有考察学生软件工程运用能力的题目存在，难题占的分数很少。其他很

多是送分的题目。

3. 有难度的题目占极少部分，可以根据时间安排，先回答别的题目。

4. 所有题目都要写到试卷的指定位置，以免流水阅卷时有遗漏。

# 英文版《软件工程》教学内容回顾 2017 上

(下述问题仅是课件中的主要部分回顾，问题答案以课件为主要参考)

## Chapter01

SE 的定义、目的、方法及作用 (P2 / P16)

//开发模式 (paradigm) (P4)

说明错误、缺陷、失败的含义与联系。(请举例说明) (6 页) (44 页习题 3)

软件质量应从哪几个方面来衡量? 论述之。(9--12 页)

//软件系统的系统组成 (P16)

现代软件工程大致包含的几个阶段及各个阶段文档 (P23-24)

//使现代 SE 实践发生变化的 (七个) 关键因素是什么? (28--29 页)

什么是抽象? (30 页)

什么是软件过程? 软件过程的重要性是什么? 包含几个阶段? (32 页) (45 页)

什么是重用等软件工程主要概念? (34 页)

## Chapter02

什么是软件过程? 软件过程的重要性是什么? (P45-46)

瀑布模型及各阶段文档, 优缺点? (P49)

原型的概念 (P51)

论述分阶段开发模型的含义, 其基本分类及特点是什么? (56 页)

螺旋模型四个象限的任务及四重循环的含义? (P58)

P80--81 页 习题 2, 3。

// 在所有的软件开发过程模型中, 你认为哪些过程给予你最大的灵活性以应对需求的变更? (81 页习题 11)

什么是 UP, RUP?

## Chapter03

什么是项目进度? 活动? 里程碑? (83 页)

如何计算软件项目活动图的关键路径? (习题 2, 3) 冗余时间? 最早和最迟开始时间 (课堂习题讲解)

软件团队人员应该具备的能力是什么? (96 页)

软件项目组织的基本结构? (101 页)

// 专家估算法的大致含义? (106 页), 算式估算法的大致含义? (108 页)

试述 COCOMO 模型的三个阶段基本工作原理或含义。(111 页)

什么是软件风险? 主要风险管理活动? 有几种降低风险的策略? (119、122 页)

找出图 3.23 和 3.24 (139 页) 的关键路径。

## Chapter04



需求的含义是什么? (143 页)

需求作为一个工程, 其确定需求的过程是什么? (144 页 图 4.1)

举例说明获取需求时, 若有冲突发生时, 如何考虑根据优先级进行需求分类。(152 页)

// 如何使需求变得可测试? (151-152 页, sidebar4.4)

需求文档分为哪两类? (153 页)

什么是功能性需求和非功能性需求/质量需求? 设计约束? 过程约束? (149 页)

// 需求的特性? (正确性、一致性、完整性) (155 页)

**知道** DFD 图的构成及画法 (172 页)

在需求原型化方面, 什么是抛弃型原型? 什么是演化型原型? (192--193 页)

// 用 DFD 图简单描述 ATM 机的工作原理 (主要功能和数据流) (220 页习题 7)

## Chapter05

什么是软件体系结构? 设计模式? 设计公约? 设计? 概念设计? 技术设计? (223-224 页)

软件设计过程模型的几个阶段?

// 三种设计层次极其关系? (229 页)

什么是模块化? 什么是抽象? (238 页)

论述设计用户界面应考虑的问题。(242 页)

5.5 节----模块独立性----耦合与内聚的概念及各个层次划分? (248----xxx 页)

举例说明耦合与内聚的基本分类。以及各个分类的含义与特征 (284 页习题 4, 5)

## Chapter06

// 什么是面向对象? (286 页) OO 有几个基本特征? 如何使用高级语言实现这些基本

// 特征? 了解并使用高级语言的 OO 基本编程方法和技巧。(286-291)

什么是设计模式?

OO 设计的基本原则?

OO 开发有何优势? (291 页)

OO 开发过程有几个步骤? (292 页)

**熟悉用例图**的组成和画法, 用例的几个要素的含义, 掌握用例图的实例解析方法 (294 页)

用例图、类图等对面向对象的项目开发的意义是什么?

**熟悉类图**中各个类之间的基本关系分类 (303-305)

熟悉类图等的组成和画法 (300-308 页)

**知道** UML 其他图示结构的基本用途。

## Chapter07

//为什么说编码工作是纷繁复杂甚至令人气馁? (337 页)

//一般性的编程原则应该从哪三个方面考虑? (340-344 页)

//论述编码阶段实现某种算法时所涉及的问题。(342 页)

在编写程序内部文档时, 除了 HCB 外, 还应添加什么注释信息? (352-354 页)

什么是极限编程(XP)? 以及派对编程? (357 页)

## Chapter08

// 产生软件缺陷的原因？（365 页）  
// 将软件缺陷进行分类的理由？（367 页）  
几种主要的缺陷类型？（367-368 页）  
什么是正交缺陷分类？（369 页）  
测试的各个阶段及其任务？（372 页图 8.3）  
// 测试的态度问题？（为什么要独立设置测试团队？）（373 页）  
**掌握**测试的方法----黑盒、白盒的概念？（374）  
什么是单元测试？什么是走查和检查？（376 页）  
黑盒白盒方法各自的分类？测试用例的设计和给出方法（结合补充材料）  
黑盒白盒方法的分类，各种覆盖方法等。（课件和补充课件）  
如何面对一个命题，设计和给出测试用例的问题。（课件）  
-----课堂练习的测试题目和讲解内容  
集成测试及其主要方法的分类？（390-392）（驱动，桩的概念）  
// 传统测试和 OO 测试有何不同？OO 测试有何困难？（398-399 页）  
// 测试计划涉及的几个步骤？（400 页）（了解）

## Chapter09

系统测试的主要步骤及各自含义？（420 页，图 9.2）  
什么是系统配置？软件配置管理？ // 基线？（423 页）（或见课件）  
// 什么是回归测试？（425 页）  
功能测试的含义及其作用？（430 页）  
功能测试的基本指导原则？（431）  
性能测试的含义与作用？（436 页）  
性能测试的主要分类？（436 页）  
// 什么是可靠性、可用性和可维护性？（438 页）  
确认测试，确认测试分类？（基准测试和引导测试）（447-448 页）  
什么是  $\alpha$  测试？ $\beta$  测试？（448 页）  
// 什么是安装测试？（450 页）

**注意：**

每一章节的开头中，大的概念性问题是引入的？，其讨论请见课件。

**试卷答题须知：**

1. 软件工程课程出题覆盖范围比较广泛，考察手段有灵活和多样化特点。
2. 有考察学生软件工程运用能力的题目存在，难题占的分数很少。其他很

多是送分的题目。

3. 鉴于有难度的题目占极少部分，可以根据时间安排，先回答别的题目。

4. 所有题目都要写到试卷的指定位置，以免流水阅卷时有遗漏。

# 英文版《软件工程》教学内容回顾 2017 下

(下述问题仅是课件中的主要部分回顾，问题答案以课件为主要参考)

## Chapter01

SE 的定义、目的、方法及作用

开发模式 (paradigm)

说明错误、缺陷、失败的含义与联系。(请举例说明)

软件质量应从哪几个方面来衡量? 论述之。

//软件系统的系统组成。

现代软件工程大致包含的几个阶段及各个阶段文档。

//使现代 SE 实践发生变化的 (七个) 关键因素是什么?

//什么是抽象?

什么是软件过程? 软件过程的重要性是什么? 包含几个阶段?

什么是重用等软件工程主要概念?

## Chapter02

什么是软件过程? 软件过程的重要性是什么?

瀑布模型及各阶段文档, 优缺点?

原型的概念与用途。

论述分阶段开发模型的含义, 其基本分类及特点是什么?

螺旋模型四个象限的任务及四重循环的含义?

// ----- 习题 2, 3。

// 在所有的软件开发过程模型中, 你认为哪些过程给予你最大的灵活性以应对需求的变更?

什么是 UP, RUP?

## Chapter03

什么是项目进度? 活动? 里程碑?

如何计算软件项目活动图的关键路径? (习题 2, 3) 冗余时间? 最早和最迟开始时间 (课堂习题讲解)

// 软件团队人员应该具备的能力是什么?

软件项目组织的基本结构?

专家估算法的大致含义? 算式估算法的大致含义?

试述 COCOMO 模型的三个阶段基本工作原理或含义。

什么是软件风险? 主要风险管理活动? 有几种降低风险的策略?

找出图 3.23 和 3.24 (139 页) 的关键路径。

## Chapter04

需求的含义是什么？

需求作为一个工程，其确定需求的过程是什么？

举例说明获取需求时，若有冲突发生时，如何考虑根据优先级进行需求分类。

// 如何使需求变得可测试？（sidebar4.4）

需求文档分为哪两类？

什么是功能性需求和非功能性需求/质量需求？ 设计约束？过程约束？如何区分？

// 需求的特性？（正确性、一致性、完整性）。

知道 DFD 图的构成及画法。

// 在需求原型化方面，什么是抛弃型原型？什么是演化型原型？

// 用 DFD 图简单描述 ATM 机的工作原理（主要功能和数据流）（习题 7）

## Chapter05

什么是软件体系结构？设计模式？设计公约？设计？概念设计？技术设计？

软件设计过程模型的几个阶段？

// 三种设计层次极其关系？

// 什么是模块化？什么是抽象？

论述设计用户界面应考虑的问题。

5.5 节----模块独立性----耦合与内聚的概念及各个层次划分？

举例说明耦合与内聚的基本分类。以及各个分类的含义与特征

## Chapter06

// 什么是面向对象？OO 有几个基本特征？如何使用高级语言实现这些基本// 特征？

了解并使用高级语言的 OO 基本编程方法和技巧。

什么是设计模式？

OO 设计的基本原则？

OO 开发有何优势？

OO 开发过程有几个步骤？

了解用例图的组成和画法，用例的几个要素的含义。 //掌握用例图的实例解析方法

用例图、类图等对面向对象的项目开发的意义是什么？

了解熟悉类图中各个类之间的基本关系分类

熟悉用例图、类图、通讯图等的组成和画法。

知道 UML 其他图示结构的基本用途。

## Chapter07

//为什么说编码工作是纷繁复杂甚至令人气馁？

//一般性的编程原则应该从哪三个方面考虑？

//论述编码阶段实现某种算法时所涉及的问题。

在编写程序内部文档时，除了 HCB 外，还应添加什么注释信息？

什么是极限编程(XP)？以及派对编程？

## Chapter08

// 产生软件缺陷的原因？  
// 将软件缺陷进行分类的理由？  
有几种主要的缺陷类型？  
什么是正交缺陷分类？  
测试的各个阶段及其任务？（图 8.3）  
// 测试的态度问题？（为什么要独立设置测试团队？）  
**掌握**测试的方法---黑盒、白盒的概念？  
什么是单元测试？ 什么是走查和检查？  
黑盒白盒方法各自的分类？测试用例的设计和给出方法（结合补充材料）  
黑盒白盒方法的分类，各种覆盖方法等。（课件和补充课件）  
如何面对一个命题，设计和给出测试用例的问题。（课件）  
-----课堂练习的测试题目和讲解内容  
集成测试及其主要方法的分类？（驱动，桩的概念）  
// 传统测试和 OO 测试有何不同？OO 测试有何困难？  
// 测试计划涉及的几个步骤？ （了解）

## Chapter09

系统测试的主要步骤及各自含义？（图 9.2）  
什么是系统配置？软件配置管理？ // 基线？（或见课件）  
// 什么是回归测试？  
功能测试的含义及其作用？  
功能测试的基本指导原则？  
性能测试的含义与作用？  
性能测试的主要分类？  
// 什么是可靠性、可用性和可维护性？  
确认测试，确认测试分类？（基准测试和引导测试）  
什么是 alpha 测试？ $\beta$  测试？  
// 什么是安装测试？

**注意：**

每一章节的开头中，大的概念性问题是引入的？，其讨论请见课件。

**试卷答题须知：**

1. 软件工程课程出题覆盖范围比较广泛，考察手段有灵活和多样化特点。
2. 有考察学生软件工程运用能力的题目存在，难题占的分数很少。其他很多是送分的题目。

3. 鉴于有难度的题目占极少部分，可以根据时间安排，先回答别的题目。
4. 所有题目都要写到试卷的指定位置，以免流水阅卷时有遗漏。

# 软件工程模拟试卷 2014 年下 (A)

注意 1: 请使用中文回答问题。

注意 2: 所有题目都要写到试卷的指定位置 (例如: 试卷上), 以免流水阅卷时有遗漏

注意 3: 复习时以课件为主, 英文教材为辅。

## 一. 解释下列名词的含义 (每个小题 2 分, 共 X 分)

1. 软件工程 (Software Engineering)
2. 过程 (Process)
3. 功能性需求 (Functional Requirement)
4. 黑盒测试 (Black-box Test)

## 二. 判断 (填写 × 或 √) (每个小题 1 分, 共 X 分)

1. ( ) 当软件系统的效率与可维护性产生抵触时, 应强调效率。
2. ( ) 开发软件时可随便选择一种语言进行开发。
3. ( ) 抽象是面向对象的开发方法中独有的策略, 在传统的开发方法中不需使用。
4. ( ) 桩模块的编写比驱动模块更困难。

## 三. 从供选择的答案中, 选出正确的答案填入 ( ) 内。(每个题空 1 分, 共 X 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

3. 软件是计算机系统中与硬件相互依存的部分，它是包括 (A)、(B) 及 (C) 的完整集合。其中，(A) 是按事先设计的功能和性能要求执行的指令序列，(B) 是使程序能够正确操纵信息的数据结构，(C) 是与程序开发、维护和使用有关的图文资料。

A, B, C: ① 软件      ② 程序      ③ 代码      ④ 硬件  
             ⑤ 文档      ⑥ 外设      ⑦ 数据      ⑧ 图表

填入答案 (A:              B:              C:              )

#### 四. 简述题 (每个小题 5 分, 共 X 分)

1. 某个开发模型的含义及特点。
2. 列出你所知道的模块间的各种内聚 (Cohesion) 关系
3. 说明软件的测试过程的主要步骤及含义。
4. 简述如何使需求变得更具测试性。

#### 五. 综合应用题 (共 X 分)

1. 根据给定的命题或分析图, 添加内容或回答问题。
2. 测试方面的题目 (基本概念, 单元测试, 集成测试概念等)。
- // 3. 高级语言程序设计的基本技术。
- // 4. OO 的基本设计方法和技巧等。

# 软件工程模拟试卷 2016 年上 (A)

备注 1: 请使用中文回答问题。

备注 2: 所有题目都要写到试卷的指定位置 (例如: 试卷上), 以免流水阅卷时有遗漏

备注 3: 复习时以课件为主, 英文教材为辅。

一. 解释下列名词的含义 (每个小题 2 分, 共 X 分)

2. 过程 (Process)
3. 功能性需求 (Functional Requirement)
4. 黑盒测试 ( Black-box Test)

二. 判断 (填写 × 或 √) (每个小题 1 分, 共 X 分)

1. ( ) 当软件系统的效率与可维护性产生抵触时, 应强调效率。
2. ( ) 开发软件时可随便选择一种语言进行开发。
3. ( ) 抽象是面向对象的开发方法中独有的策略, 在传统的开发方法中不需使用。
4. ( ) 桩模块的编写比驱动模块更困难。

三. 从供选择的答案中, 选出正确的答案填入 ( ) 内。(每个题空 1 分, 共 X 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

3. 软件是计算机系统中与硬件相互依存的部分，它是包括 (A)、(B) 及 (C) 的完整集合。其中，(A) 是按事先设计的功能和性能要求执行的指令序列，(B) 是使程序能够正确操纵信息的数据结构，(C) 是与程序开发、维护和使用有关的图文资料。

A, B, C: ① 软件      ② 程序      ③ 代码      ④ 硬件  
             ⑤ 文档      ⑥ 外设      ⑦ 数据      ⑧ 图表

填入答案 (A:                B:                C:                )

#### 四. 简述题 (每个小题 6 分, 共 X 分)

1. 某个开发模型的含义及特点。
2. 列出你所知道的模块间的各种内聚 (Cohesion) 关系
3. 说明软件测试过程的主要步骤及含义。
4. 简述如何使需求变得更具测试性。

#### 五. 综合应用题 (共 X 分)

1. 根据给定的命题或分析图, 添加内容或回答问题或画图。
2. 测试方面的题目 (基本概念, 或单元测试, 或集成测试概念等)。
3. 高级语言程序设计的基本技术用以实现软件工程的某个概念。
- // 4. OO 的基本设计方法和技巧等。

# 软件工程模拟试卷 2016 年下 (A)

备注 1: 请使用中文回答问题。

备注 2: 所有题目都要写到试卷的指定位置 (例如: 试卷上), 以免流水阅卷时有遗漏

备注 3: 复习时以课件为主, 英文教材为辅。

一. 解释下列名词的含义 (每个小题 2 分, 共 X 分)

2. 过程 (Process)
3. 功能性需求 (Functional Requirement)
4. 黑盒测试 ( Black-box Test)

二. 判断 (填写 × 或 √) (每个小题 1 分, 共 X 分)

1. ( ) 当软件系统的效率与可维护性产生抵触时, 应强调效率。
2. ( ) 开发软件时可随便选择一种语言进行开发。
3. ( ) 抽象是面向对象的开发方法中独有的策略, 在传统的开发方法中不需使用。
4. ( ) 桩模块的编写比驱动模块更困难。

三. 从供选择的答案中, 选出正确的答案填入 ( ) 内。(每个题空 1 分, 共 X 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

3. 软件是计算机系统中与硬件相互依存的部分，它是包括 (A)、(B) 及 (C) 的完整集合。其中，(A) 是按事先设计的功能和性能要求执行的指令序列，(B) 是使程序能够正确操纵信息的数据结构，(C) 是与程序开发、维护和使用有关的图文资料。

A, B, C: ① 软件      ② 程序      ③ 代码      ④ 硬件  
             ⑤ 文档      ⑥ 外设      ⑦ 数据      ⑧ 图表

填入答案 (A:                B:                C:                )

#### 四. 简述题 (每个小题 5 分, 共 X 分)

1. 某个开发模型的含义及特点。
2. 列出你所知道的模块间的各种内聚 (Cohesion) 关系
3. 说明软件测试过程的主要步骤及含义。
4. 简述如何使需求变得更具测试性。

#### 五. 综合应用题 (共 X 分)

1. 根据给定的命题或分析图, 添加内容或回答问题或画图。
2. 测试方面的题目 (基本概念, 或单元测试, 或集成测试概念等)。
- // 3. 高级语言程序设计的基本技术用以实现软件工程的某个概念。
- // 4. OO 的基本设计方法和技巧等。

# 软件工程模拟试卷 2017 年上 (A)

备注 1: 请使用中文回答问题。

备注 2: 所有题目都要写到试卷的指定位置 (例如: 试卷上), 以免流水阅卷时有遗漏

备注 3: 复习时以课件为主, 英文教材为辅。

一. 解释下列名词的含义 (每个小题 2 分, 共 X 分)

2. 过程 (Process)
3. 功能性需求 (Functional Requirement)
4. 黑盒测试 ( Black-box Test)

二. 判断 (填写 × 或 √) (每个小题 1 分, 共 X 分)

1. ( ) 当软件系统的效率与可维护性产生抵触时, 应强调效率。
2. ( ) 开发软件时可随便选择一种语言进行开发。
3. ( ) 抽象是面向对象的开发方法中独有的策略, 在传统的开发方法中不需使用。
4. ( ) 桩模块的编写比驱动模块更困难。

三. 从供选择的答案中, 选出正确的答案填入 ( ) 内。(每个题空 1 分, 共 X 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

3. 软件是计算机系统中与硬件相互依存的部分，它是包括 (A)、(B) 及 (C) 的完整集合。其中，(A) 是按事先设计的功能和性能要求执行的指令序列，(B) 是使程序能够正确操纵信息的数据结构，(C) 是与程序开发、维护和使用有关的图文资料。

A, B, C: ① 软件      ② 程序      ③ 代码      ④ 硬件  
             ⑤ 文档      ⑥ 外设      ⑦ 数据      ⑧ 图表

填入答案 (A:                B:                C:                )

#### 四. 简述题 (每个小题 6 分, 共 X 分)

1. 名词解释。
2. 某个开发模型的含义及特点。
3. 列出你所知道的模块间的各种耦合关系
4. 说明软件测试过程的主要步骤及含义。

#### 五. 综合应用题 (共 X 分)

1. 根据给定的命题或分析图, 添加内容或回答问题。
  2. 测试方面的题目 (基本概念, 或单元测试, 或集成测试概念等)。
  3. 高级语言程序设计的基本技术用以实现软件工程的某个概念。
- // 4. OO 的基本设计方法和技巧等。

# 软件工程模拟试卷 2017 年下 (A)

备注 1: 请使用中文回答问题。

备注 2: 所有题目都要写到试卷的指定位置 (例如: 试卷上), 以免流水阅卷时有遗漏

备注 3: 复习时以课件为主, 英文教材为辅。

一. 解释下列名词的含义 (每个小题 2 分, 共 X 分)

2. 过程 (Process)
3. 功能性需求 (Functional Requirement)
4. 黑盒测试 ( Black-box Test)

二. 从供选择的答案中, 选出正确的答案填入 ( ) 内。(每个题空 1 分, 共 X 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
3. 软件是计算机系统中与硬件相互依存的部分, 它是包括 (A)、(B) 及 (C) 的完整集合。其中, (A) 是按事先设计的功能和性能要求执行的指令序列, (B) 是使程序能够正确操纵信息的数据结构, (C) 是与程序开发、维护和使用有关的图文资料。

A, B, C: ① 软件      ② 程序      ③ 代码      ④ 硬件  
             ⑤ 文档      ⑥ 外设      ⑦ 数据      ⑧ 图表

填入答案 (A:                B:                C:                )



### 三. 简述题（每个小题 6 分，共 X 分）

1. 术语解释-----英文全称或缩写。
2. 某个开发模型的含义及原理特点。
3. 列出你所知道的模块间的各种聚合关系
4. 说明软件测试过程的主要步骤及含义。

### 四. 综合应用题（共 X 分）

1. 根据给定的文字命题或分析图， 添加内容或回答问题。
2. 根据文字说明，画出某种图示。
- //3. 高级语言程序设计的基本技术用以实现软件工程的某个概念。
- // 4. OO 的基本设计方法和技巧等。

# 软件工程模拟试卷 2019 年下 (A)

备注 1: 请使用中文回答问题。

备注 2: 所有题目都要写到试卷的指定位置 (例如: 试卷上), 以免流水阅卷时有遗漏

## 一. 解释下列名词的含义 (每个小题 2 分, 共 X 分)

1. 软件工程 (Software Engineering)
2. 过程 (Process)
3. 黑盒测试 ( Black-box Test)

## 二. 判断 (填写 × 或 √) (每个小题 1 分, 共 X 分)

1. ( ) 当软件系统的效率与可维护性产生抵触时, 应强调效率。
2. ( ) 开发软件时可随便选择一种语言进行开发。
3. ( ) 抽象是面向对象的开发方法中独有的策略, 在传统的开发方法中不需使用。
4. ( ) 桩模块的编写比驱动模块更困难。

## 三. 从供选择的答案中, 选出正确的答案填入 ( ) 内。(每个题空 1 分, 共 X 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
3. 软件是计算机系统中与硬件相互依存的部分, 它是包括 (A)、(B) 及 (C) 的完整集合。其中, (A) 是按事先设计的功能和性能要求执行的指令序列, (B) 是使程序能够正确操纵信息的数据结构, (C) 是与程序开发、维护和使用有关的图文资料。

A, B, C: ① 软件      ② 程序      ③ 代码      ④ 硬件

⑤ 文档      ⑥ 外设      ⑦ 数据      ⑧ 图表

填入答案 (A:              B:              C:              )

#### 四. 简述题 (每个小题 6 分, 共 X 分)

- //    1. 名词解释。
2. 某个开发模型的含义及特点。
  3. 列出你所知道的模块间的各种耦合关系
  4. 说明软件测试过程的主要步骤及含义。
  5. 提出某种观点, 请予以评价并说明理由。

#### 五. 综合应用题 (共 X 分)

1. 根据给定的文字命题, 画出 UML 分析图以及描述。
2. 测试方面的题目  
(提出实际命题, 要求给出测试用例)。  
(基本概念, 或单元测试, 或集成测试概念等)。
3. 高级语言程序设计的基本技术用以实现软件工程的某个设计概念。
4. OO 的基本设计方法和技巧等。