软件体系结构试题

# 一、简答题

1. 什么是软件体系结构？它有哪三种类型？软件产品在实现时通常分解成一些较小的模块，这种做法的意义是什么？分解时应遵循的原则是什么？

答：一个软件系统的体系结构是指构成这个系统的计算部件（computational components）、部件间的相互作用关系（interactions）。部件间的相互作用关系称为连接器（connector）。包括三种类型，分别为：概略型、是上层宏观结构的描述，反映系统最上层的部件和连接关系；需求型、对概略型体系结构的深入表达，以满足用户功能和非功能的需求；设计型、从设计实现的角度对需求结构的更深层的描述，设计系统的各个部件，描述各部件的连接关系。这是软件系统结构的主要类别。软件产品在实现时通常分解成一些较小的模块，这种做法的意义是：解决复杂问题的一种有效方法；集体分工协作的前提；产品维护的保障。原则：高内聚性和低耦合性。

2. 在C/S结构中，有状态信息服务器和无状态信息服务器相比，哪一种服务器对同时使用的用户数有限制？为什么？

答：有状态信息服务器对同时使用的用户数有限制。因为有状态信息服务器需要设计一个状态信息表，用于登记当前的用户操作请求，当状态信息表填满时，新用户的操作被推迟，意味着同时使用的用户数受限制。

3. 总结P/F结构风格的缺点。

答：1. 通常导致进程成为批处理的结构。这是因为虽然过滤器可增量式地处理数据，但它们是独立的，所以设计者必须将每个过滤器看成一个完整的从输入到输出的转换。 2. 不适合于需要共享大量数据的应用设计。 3. 不适合处理交互的应用。当需要增量地显示改变时，这个问题尤为严重。 4. 过滤器之间通过特定格式的数据进行工作，数据格式的设计和转换是系统设计的主要方面，为了确保过滤器的正确性，必须对数据的句法和语义进行分析，这增加了过滤器设计的复杂性。 5. 并行运行获得高效率往往行不通。原因，第一，独立运行的过滤器之间的数据传送的效率会很低，在网络传送时尤其如此；第二，过滤器通常是在消耗了所有输入后才产生输出；第三，在单处理器的机器上进程的切换代价是很高的；第四，通过管道对过滤器的同步控制可导致频繁启动和停止过滤器工作。

4. 在分布式令牌环互斥算法中，一个进程进入临界区执行需要令牌消息传递的个数是，解释这里的含义。

答：在分布式令牌环互斥算法中，算法保持一个令牌在环中传递，一个进程只有在得到令牌时才能进入临界区执行，以便实现互斥；即使当前所有进程都没有申请进入临界区运行，令牌也要沿环不断地传递；因此，当一个进程要进入临界区执行是，令牌消息传递的个数是不确定的，可以有任意多次，即。

5. 解释RPC透明性的含义。PRC中客户代理（client stub）实现什么主要功能？

答：RPC透明性的含义是：RPC尽可能像传统系统的本地内部调用一样。Client Stub的主要功能是：参数整理、查找服务器地址、请服务器发送请求、阻塞等待接收结果、分析结果返回并唤醒client进程。

6. 简述QDS的思想、作用及目标。

答：QDS的思想是基于设计空间（design space）和质量配置函数（quality function deployment, QFD）。QDS的作用是一种将系统需求转换为可选择的功能和结构设计，量化（quantified）分析各种的可能选择及其组合。QDS的目标为一个期望系统（desired system）制作（produce）一个设计模型，也可以用来分析和对比已有的设计。或对现有的设计提出改进建议。作为自动系统设计（automated system design）的基础。

# 二、单项选择题

1. 软件体系结构是一门新兴学科，它来源于【 C 】

A. 程序设计语言 B. 数据库技术

C. 软件工程 D. 网络通信

2. RPC中，“孤报（orphan）”的产生是源于【 B 】

A. server crash B. client crash C. lost request D. lost reply

3. 假定两台机器的CPU分别是Intel和SPARC，字长为16位的机器。现有Intel机器上的一个十进制整型参数4传递到SPARC的机器上，如果没有参数整理那么在SPARC上接收到的整数（十进制）是【 C 】

A. 4 B.  C.  D. 

4. 广泛用于建立虚拟机的软件体系结构风格是【 B 】

A. Pipes and Filters B. Interpreters

C. client-server D. State transition system

5. 在模块设计的耦合度模型中，下列【 A 】程序员应该尽量避免。

A. 内容耦合 B. 控制耦合

C. 特征耦合 D. 数据耦合

6. 在数据库管理系统中确保数据的一致性约束，是应用了基于【 A 】

A. 事件及级隐含激活的风格

B. 仓库的风格

7. 基于消息传递的通信中，连接器是采用可靠的通信，则可靠是指【 B 】

A. 发送进程发出的消息接收进程一定可以收到

B. 发送进程一定可以获知消息是否被接收

8. 一个软件重用（reuse）是指【 A 】

A. 指利用一个产品的构件以使开发另一个具有不同性能的产品更容易

B. 将该产品转移到新的计算机上运行所需的调整费用大大低于从头开始编写一个新产品的费用

9. 一个好的模块设计应该【 B 】

A. 使模块具有低内聚性和高耦合性

B. 使模块具有高内聚性和低耦合性

10. 如果想得到较好的User customizability （用户定制）能力，则Representation issues （表示问题）结构维最好是选择【 A 】

A. External declarative notation

B. Internal procedural notation

# 三、填空题

1. 根据服务于软件开发的不同阶段，体系结构可分为：概略型、 需求型 和设计型等三种。

2. 软件体系结构的基本内容主要有： 结构风格 、设计模式和形式化及结构描述语言。

3. 解释器结构风格可以减少 程序运行需要的机器与实际物理机器 。

4. 设计规则主要体现 维之间的关系程度或组合的好坏 。

5. 在数据抽象与面向对象的结构风格中，组件是 对象 ，连接器是 激活对象方法（method）的消息（message） 。

6. IDL的 语言独立性 ，使CORBA成为异构系统的集成技术。

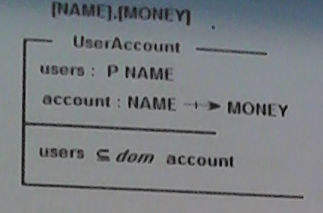
7. RPC透明性的由 客户代理 和 服务器代理 实现。

8. 形式化规范（formal specifications）就是用 数学表示法（notation） 精确地描述信息系统的属性，而不过分注重实现这些属性的方法。

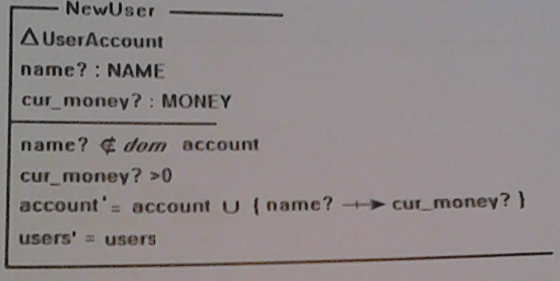
# 四、形式化设计

1. 某简单账户管理系统主要维护用户姓名和当前账目余额，提供账户新增加、查询、收入和支出等操作，要求账目余额不小于0.请应用Z Notation形式化如下的系统设计。

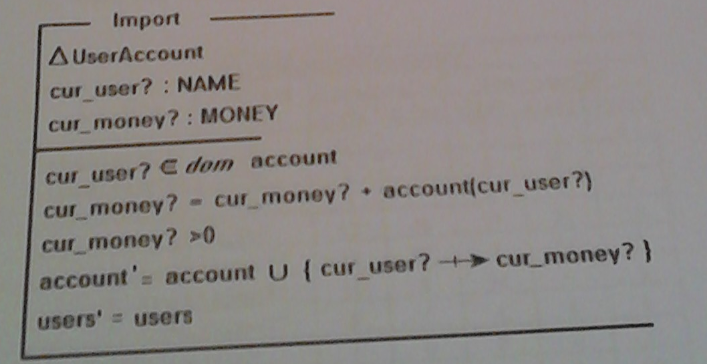
1）定义一个状态模式（UserAccount），描述账户的基本属性。



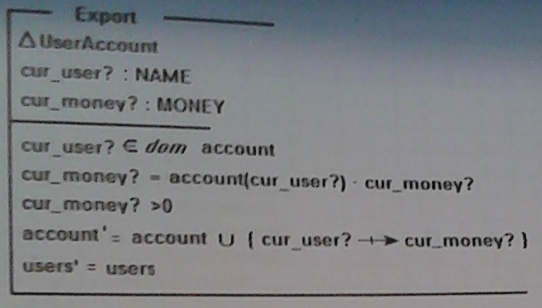
2）定义一个操作模式（NewUser），用于注册一个新的用户的用户账号和初始余额，规定同一个账户不能重复注册。

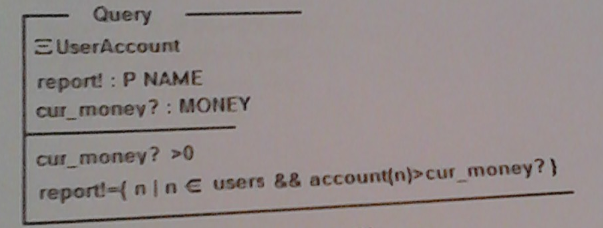


3）定义一个操作模式（Import），向给定用户cur\_user的账目中添加指定的金额cur\_money。



4）定义一个操作模式（Export），从给定用户cur\_user的账目中扣除指定的金额cur\_money。



5）定义一个操作模式（Query），查询系统中账目余额小于指定金额cur\_money的所有用户。

# 五、结构分析题

1. 已知某产品的设计空间的功能维子空间、结构维子空间以及规则（如表1、表2、表3）。完成下面两个问题：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表1：通用功能维子空间及量化 | | | | | | |
| Concepts | Customer Weight | Functional#1 | | Fuctional#2 | | |
| Alt#1 | Alt#2 | Alt#1 | Alt#2 | Alt#3 |
| Requirement#1 | 9 | 2 | 3 | 1 | 3 | 3 |
| Requirement#2 | 8 | 2 | 1 | 3 | 0 | 1 |
| Requirement#3 | 2 | 3 | 2 | 2 | 2 | 2 |
| Calculated Weight | | **40** | **39** | **37** | **31** | **39** |
|  | Requirement#1 | **18** | **27** | **9** | **27** | **27** |
|  | Requirement#2 | **16** | **8** | **24** | **0** | **8** |
|  | Requirement#3 | **6** | **4** | **4** | **4** | **4** |
| Design decision | | **1** | **0** | **0** | **0** | **1** |
| Effective weight | | **40** | **0** | **0** | **0** | **39** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表2：通用结构维子空间及量化 | | | | | | | | |
| Concepts | | Alternatives | Effective Weight | Structural#1 | | | Structural#2 | |
| Alt#1 | Alt#2 | Alt#3 | Alt#1 | Alt#2 |
| Functional#1 | | Alt#1 | **40** | 2 | 1 | 0 | 2 | 1 |
| Alt#2 | **0** | 1 | 3 | 2 | 3 | 2 |
| Functional#2 | | Alt#1 | **0** | 1 | 0 | 1 | 0 | 0 |
| Alt#2 | **0** | 2 | 1 | 5 | 4 | 0 |
| Alt#3 | **39** | 5 | 8 | 3 | 9 | 5 |
| Calculated Weight | | | | **275** | **352** | **117** | **431** | **235** |
|  | Requirement#1 | | | **171** | **234** | **81** | **279** | **153** |
|  | Requirement#2 | | | **72** | **80** | **24** | **104** | **56** |
|  | Requirement#3 | | | **32** | **38** | **12** | **48** | **26** |
| Design decision | | | | **0** | **1** | **0** | **1** | **0** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 表3：结构维子空间内部规则 | | | | | | |
| Concept | Alternatives | Structural#1 | | | Structural#2 | |
| Alt#1 | Alt#2 | Alt#3 | Alt#1 | Alt#2 |
| Structural#1 | Alt#1 |  | 1 | 0.5 | 0 | 1 |
| Alt#2 |  |  | 5 | 0.5 | 3 |
| Alt#3 |  |  |  | 4 | 1 |
| Structural#2 | Alt#1 |  |  |  |  | 5 |
| Alt#2 |  |  |  |  |  |

1）采用QFD量化算法，完成表1和表2的剩余单元格的数据，并给出量化过程的设计选择结果。

答：经量化，得到系统的一个设计：功能维选择F1.Alt#1、F2.Alt#3，结构维选择S1.Alt#2、S2.Alt#1。

2）结合表3，计算该设计的得分（含总分和各需求的得分，要求写出计算过程）。

答：该设计的得分：

Request#1: 0.5\*(234+279)=256.5

Request#2: 0.5\*(80+104)=92

Request#3: 0.5\*(38+48)=43

总得分：391.5

1. 什么是UIPC？他有哪些优点？P44