

计算机学 2014 级硕士研究生《面向对象方法》试题

(满分: 100 分, 考试时间: 2015-07-08)

一、结合 COM 组件规范和相关技术, 回答下列问题。(16 分, 每题 4 分)

1、COM 组件区别于普通可执行程序的特征有哪些?

【答】COM 组件易于动态组装, 组件的实现与语言无关, 对使用者透明; 可以以二进制形式发布; 组件的升级不影响试图继续使用它的老客户; 组件可以透明地在网络上进行位置分配。对远程组件的使用和对本地机器上的组件的使用对客户来说无差别。

模块是指整个程序中相对独立的程序单元, 每个程序单元完成和实现一个相对独立的软件功能。模块是面向应用的基本单元, 由一个或多个相同组件类型的组件组成。

为了对组件进行部署, 组件必须首先封装到模块中连同部署描述文件一起绑定到某个指定的文件中。而模块可以作为单独的软件单元分别进行部署, 也可以进一步封装为更复杂的配置, 来构成一个可以应用的软件系统。

2、用 C++ 或 Java 给出 IUnknown 接口中 QueryInterface 函数原型的设计, 并简述该函数的参数及返回结果的语义 (或含义)。

【答】(1) 函数原型: `HRESULT STDMETHODCALLTYPE QueryInterface(const IID& iid, void **ppv);` (2) 参数及返回值: 第一个参数是标识客户所需的接口, 此参数是一个“接口标识符”(IID) 结构。第二个参数是存放所请求接口指针的地址。QueryInterface 返回的是一个 HRESULT 值, 是一个具体特定结构的 32 位值, 可返回 S_OK 或 E_NOINTERFACE。

3、简述类厂接口 (工厂接口) 的作用。

【答】类厂接口 IClassFactory 是创建组件的标准接口, 客户可以通过类厂组件创建其它组件, 类厂组件的唯一功能就是创建其它组件。(某个特定的类厂将创建只同某个特定的 CLSID 相应的组件, 客户可以通过类厂所支持的接口来对类厂创建组件的过程加以控制。)

4、简述引用计数器及 IUnknown 接口中的 AddRef 和 Release 函数的作用。

【答】(1) 当客户从组件取得一个接口时, 此引用计数值将增 1。当客户使用完某个接口后, 组建的引用计数将减 1。当引用计数值为 0 时, 组建即可将自己从内存中删除。(2) AddRef 和 Release 通过计数器共同完成引用计数功能。为知道一个组件何时没有客户, 利用一个计数器来记录该组件的使用情况, 也可为每个接口安排一个计数器, 当所有接口的计数器为 0 时, 意味着该组件的不存在客户了, 可从内存中卸载了。

二、复用已有的 COM 组件, 可以设计并实现新的 COM 组件, 但对于任意一个 COM 组件, 如果支持被其他的 COM 组件复用, 需要进行合理的设计, 并在允许被其他 COM 组件复用的情况下, 叙述利用动态链接库设计实现一个 COM 组件的一般过程。(10 分)

【答】(1) 定义各业务接口, 各接口都能继承 IUnknown 接口; (2) 为每个组件和接口分配一个唯一标识符; (3) 实现各业务接口的功能; (4) 定义类厂接口并分配唯一标识符; (此接口继承自 IClassFactory 接口) (5) 实现类厂接口以实现动态生成组件; (6) 编写 DLL, 在 DLL 函数中有注册组件到注册表的函数、取消注册的函数、与类厂相连的函数。

三、结合 EJB 组件规范和相关技术，回答下列各题。(16 分，每题 4 分)

1、简述你对中间件概念及作用的理解。

【答】中间件是一种独立的系统软件或服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源，中间件位于客户机和服务器的操作系统之上，管理计算机资源和网络通信。中间件是处于应用程序及其所在应用系统的内部工作方式之间的软件，中间件不仅实现互联，而且还可以实现应用之间的互操作。也可以认为中间件是位于平台和应用之间的通用服务，这些服务具有标准的程序接口协议，针对不同的操作系统和硬件平台，它们可以有符合接口和协议的多种实现。

2、简述 EJB 组件的部署描述文件的作用。

【答】EJB 部署文件是 EJB 的一个重要部分，它是 EJB 的说明文件，服务器容器按照部署文件来管理 EJB。EJB 部署文件是标准的 XML 文件，它遵守 XML 和相关的文档类型 DTD 语法规则。

部署描述文件描述部署单位内容并相对于应用环境配置组件和应用程序。同时对它们与组件之间的关系做出一定的处理，从而可以在无需编程的情况下对这些关系加以治理。

3、简述你对容器回调函数概念及作用的理解。

【答】在应用程序使用会话 bean 时，对某些特定实例及用例，可能需要精确控制，如对象创建、对象销毁等事件，会话 bean 在创建时，可能需要执行一些数据库初始化操作；而在销毁时，需要关闭一些数据库连接，在此，通过回调方法，程序就能精确控制 bean 生命周期的每个阶段，而 EJB 容器将在 bean 生命周期的适当阶段对其进行调用。

4、定义 Bean 类时，需要完成的工作由哪些？即 Bean 类中包括哪些种类的方法（函数）。

【答】定义 Bean 类时，需要完成的工作有：(1) 编写提供业务方法的类：bean 类，需要确定 bean 的种类，实现对应的接口。(2) 编写两个接口：home 接口和组件接口。(3) 创建一个 XML 部署描述文件，用来告诉服务器如何管理，名字必须为 ejb-jar.xml。(4) 把 bean、接口和部署描述文件放到一个 ejb-jar 文件中，该文件中可能不只一个 bean，但部署描述文件只有一个。(5) 用服务器开发商提供的工具把 bean 部署到服务器上。

EJBHome 接口中的方法：getEJBMetaData、getHomeHandle、remove(Handle H)、remove(Object key)；

EJBObject 接口中的方法：getPrimaryKey()、getEJBHome()、getHandle()、remove()、isIdentical(EJBObject obj)。

四、结合 RUP 方法，回答下列问题。(16 分，每题 4 分)

1、概述你对架构机制概念及作用的理解。

【答】软件架构(software architecture)是一系列相关的抽象模式，用于指导大型软件系统各个方面的设计。在 RUP 中，软件系统的构架是指系统重要构件的组织或结构，这些重要构件通过接口与不断减小的构件与接口所组成的构件进行交互。从与目的、主题、材料和结构的联系上来说，软件架构可以和建筑物的架构相比拟。一个软件架构师需要有广泛的软件理论知识和相应的经验来实施和管理软件产品的高级设计。软件架构师定义和设计软件的模块化，模块之间的交互，用户界面风格，对外接口方法，创新的设计特性，以及高层事物的对象操作、逻辑和流程。一般而言，软件系统的架构有两个要素：(1) 它是一个软件系统从整体到部分的最高层次的划分。一个系统通常是由元件组成的，而这些元件如何形成、相互之

间如何发生作用,则是关于这个系统本身结构的重要信息。详细地说,就是要包括架构元件、联结器、任务流。所谓架构元素,也就是组成系统的核心"砖瓦",而联结器则描述这些元件之间通讯的路径、通讯的机制、通讯的预期结果,任务流则描述系统如何使用这些元件和联结器完成某一项需求。(2)建造一个系统所作出的最高层次的、以后难以更改的,商业的和技术的决定。在建造一个系统之前会有很多的重要决定需要事先作出,而一旦系统开始进行详细设计甚至建造,这些决定就很难更改甚至无法更改。显然,这样的决定必定是有关系统设计成败的最重要决定,必须经过非常慎重的研究和考察。

2、概述 RUP 方法中包与子系统的作用及区别。

【答】包是一个 UML 结构,是分组的通常手法,它使得你能够把诸如用例或类之类模型元件组织为组。子系统是一种模型元素,它具有包(其中可包含其他模型元素)和类(其具有行为)的语义。子系统的行为由它所包含的类或其他子系统提供,子系统实现一个或多个接口,这些接口定义子系统可以执行的行为。

3、分别简述你对分析类中边界类、控制类和实体类作用的理解。

【答】分析类代表系统中具备职责和行为的事物的初期概念模型。这些概念模型将演进为设计模型中的类和子系统。

边界类是一种用于对系统外部环境与其内部运作之间的交互进行建模的类,这种交互包括转换事件,并记录系统表示方式(例如接口)中的变更,是接口和系统外部事物的中间体。

控制类用于对一个或几个用例所特有的控制行为进行建模,控制对象(控制类的实例)通常控制其他对象,因此它们的行为具有协调性质,控制类将用例的特有行为进行封装,是用例行为协调器。

实体类是用于对必须存储的信息和相关行为建模的类,是系统的关键抽象。实体对象(实体类的实例)用于保存和更新一些现象的有关信息,例如:事件、人员或者一些现实生活中的对象。

4、分别简述你对软件体系结构(架构)、设计框架和设计概念的理解。

【答】软件体系结构是具有一定形式的结构化元素,即构件的集合,包括处理构件、数据构件和连接构件。处理构件负责对数据进行加工,数据构件是被加工的信息,连接构件把体系结构的不同部分组合连接起来。这一定义注重区分处理构件、数据构件和连接构件,这一方法在其他的定义和方法中基本上得到保持。

软件框架,通常指的是为了实现某个业界标准或完成特定基本任务的软件组件规范,也指为了实现某个软件组件规范时,提供规范所要求之基础功能的软件产品。

软件设计是从软件需求规格说明书出发,根据需求分析阶段确定的功能设计软件系统的整体结构、划分功能模块、确定每个模块的实现算法以及编写具体的代码,形成软件的具体设计方案。软件设计是把许多事物和问题抽象起来,并且抽象它们不同的层次和角度。将问题或事物分解并模块化使得解决问题变得容易,分解的越细模块数量也就越多,它的副作用就是使得设计者考虑更多的模块之间耦合度的情况。

五、根据你对 CORBA 组件技术中在 IDL 编译器的理解,利用 RUP 方法中的需求模型描述形式,给出 IDL 编译器的需求模型,说明及要求包括:把整个 IDL 编译器看成一个黑盒,不需要考虑其内部构成:术语表中的术语个数不能少于 6 个,每个术语给出清晰的解释,所

有的用例的参与者总和不能少于2个，用例的个数不能少于2个。(12分)

六、对软件系统进行分解的基本原则是“高内聚，低耦合”，请在组件技术和 RUP 方法范畴内，任意列举出5中支持“低耦合”的概念，方法或技术，并分别简述它们支持“低耦合”的基本原理。(10分)

【答】(1) 三层架构：通常意义上的三层架构就是将整个业务应用划分为：表现层、业务逻辑层、数据访问层，区分层次的目的即为了“高内聚、低耦合”的思想。所谓三层体系结构，是在客户端与数据库之间加入了一个“中间层”，也叫组件层。

(2) Web service：是一个平台独立的，低耦合的，自包含的、基于可编程的 web 的应用程序，可使用开放的 XML 标准来描述、发布、发现、协调和配置这些应用程序，用于开发分布式的互操作的应用程序。

(3) MVC 全名是 Model View Controller，是模型(model)-视图(view)-控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。

(4) 控制反转(Inversion of Control, IoC)是一个重要的面向对象编程的法则来削减计算机程序的耦合问题，也是轻量级的 Spring 框架的核心。控制反转一般分为两种类型，依赖注入(Dependency Injection, 简称 DI)和依赖查找(Dependency Lookup)。依赖注入应用比较广泛。

(5) Hibernate 是一个开放源代码的对象关系映射框架，它对 JDBC 进行了非常轻量级的对象封装，使得 Java 程序员可以随心所欲的使用对象编程思维来操纵数据库。

(比如，使用 struts 我们可以应用 MVC 模型，使页面展现与业务逻辑分离，做到了页面展现与业务逻辑的低耦合。当我们的页面展现需要变更时，我们只需要修改我们的页面，而不影响我们的业务逻辑；同样，我们的业务逻辑需要变更的时候，我们只需要修改我们的 java 程序，与我们的页面无关。使用 spring 我们运用 IoC (反向控制)，降低了业务逻辑中各个类的相互依赖。假如类 A 因为需要功能 F 而调用类 B，在通常的情况下类 A 需要引用类 B，因而类 A 就依赖于类 B 了，也就是说当类 B 不存在的时候类 A 就无法使用了。使用了 IoC，类 A 调用的仅仅是实现了功能 F 的接口的某个类，这个类可能是类 B，也可能是另一个类 C，由 spring 的配置文件来决定。这样，类 A 就不再依赖于类 B 了，耦合度降低，重用性提高了。使用 hibernate 则是使我们的业务逻辑与数据持久化分离，也就是与将数据存储到数据库的操作分离。我们在业务逻辑中只需要将数据放到值对象中，然后交给 hibernate，或者从 hibernate 那里得到值对象。至于用 Oracle、MySQL 还是 SQL Server，如何执行的操作，与我无关。)

七、为了给本校师生提供更丰富的图书管理自愿，某高校图书馆与一家大型书店联合，在图书馆提供图书租借业务，为适应该业务，图书馆在已有的软件系统基础上，开发一个“图书租借管理系统”。已有的软件系统如下：
软件系统

(1) 学校图书馆已有“图书借阅系统”，该校师生可以利用已经注册在“校园一卡通管理系统”中的校园卡（学生卡，教师卡）对图书馆中的图书在一定期限内进行无偿借阅。它包括结束，幻术，查询，罚款，赔偿等基本功能。该系统由图书管理员为师生提供图书借阅服务时使用。

(2) 学校图书馆已有“图书库存管理系统”，该系统用来对图书骨干购入的图书进行管理。

包括：图书分类，图书上架，图书下架，图书下架等基本功能。系统由图书管理员对图书进行管理时使用。

(3) 学校的“校园网网站”可以为学校各部门发布一些信息，图书馆通过该网站的链接可以发布图书馆资源信息。

软件开发团队对拟开发的“图书租借管理系统”进行需求分析之后，获得的零散信息如下：

(1) 任意能够访问“校园网”的游客，可以通过查询功能获得图书馆可供租借的图书信息。

(2) 持校园卡的师生租借图书时，可以到图书馆，在图书馆的管理人员帮助下，通过该系统进行图书租借。

(3) 图书租借的基本含义是，在一定的期限内，租借者可以免费阅读图书：超出该期限，按照超出天数计算租金，类似于“图书借阅系统”中的罚款，但金额要小得多；而超出最后的给定期限，则认为是购买，类似于“图书借阅系统”中的赔偿，但金额可能要低于图书价钱（相当于图书打折销售）。

(4) 一方面，图书馆用以出租的图书只能来源书店，而不能把自己的固有图书出租；另一方面。图书借阅是无偿的，而图书租借可能是有偿的。所以对于同样的图书，租借者应该能够依据系统所提供的相关信息进行选择；正常借阅图书馆固有图书，还是租借书店为图书馆提供的图书。

上述信息并不完整，请根据你对这个“图书馆租借管理系统”的理解和自己的构思，完成如下问题：（共 20 分）

1. 利用 RUP 方法，给出你所理解的该系统的需求模型，其中，用例模型中的每个用例只需要给出用例的名称及简短描述：相关的参与者要出现在术语表中，术语表中的术语给出关键的词汇和对应的简单解释即可。（7 分）

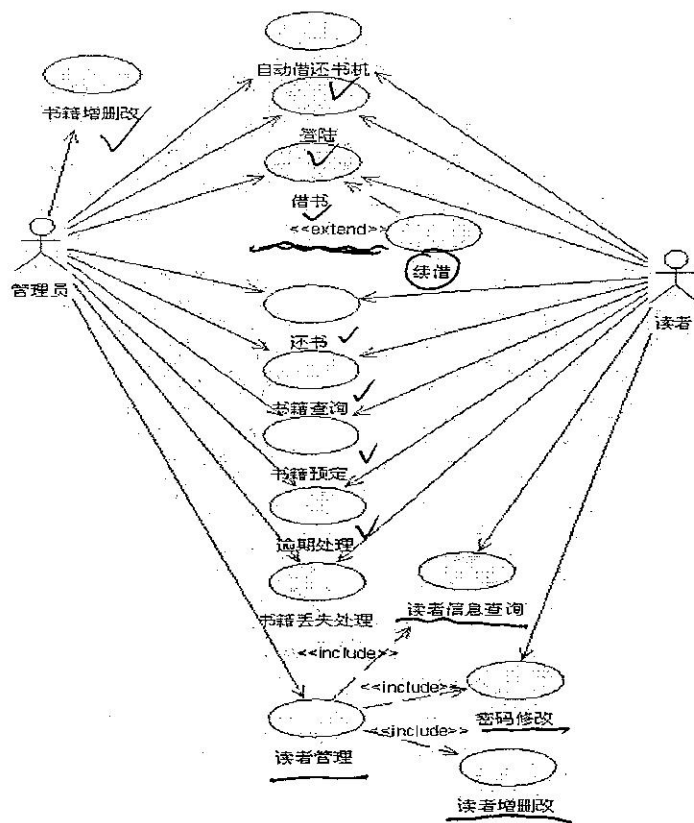
从用例图中我们可以看出管理员和读者之间对本系统所具有的用例。

管理员所包含的用例有：

- (1) 登录系统：管理员可以通过登录该系统进行各项功能的操作
- (2) 书籍管理：包括对书籍的增删改等。
- (3) 书籍借阅管理：包括借书、还书、预订、书籍逾期处理和书籍丢失处理等等。
- (4) 读者管理：包含对读者的增删改等操作。
- (5) 自动借书机的管理。

读者所包含的用例有：

- (1) 登录系统
- (2) 借书：进行借书业务。
- (3) 还书：读者具有的还书业务。
- (4) 查询：包含对个人信息和书籍信息的查询业务
- (5) 预订：读者对书籍的预订业务。
- (6) 逾期处理：就是书籍过期后的缴纳罚金等。
- (7) 书籍丢失处理：对书籍丢失后的不同措施进行处理。
- (8) 自动借书机的使用等。

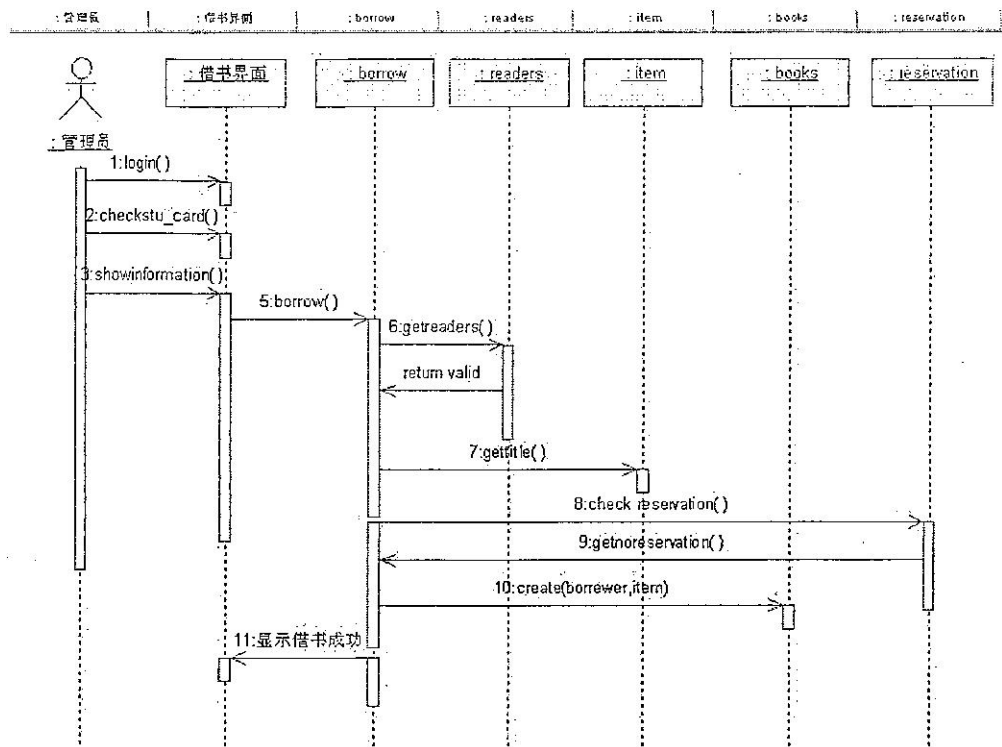


2.对能够完成图书馆租借业务功能的用例给出用例分析,给出得到的分析类,并利用顺序图描述该用例,对于得到的每个分析类,给出类的名字并列出主要的操作即可。说明:其他业务功能相关的用例不用分析(不必画出参与类图) (7分)

【时序图说明】

- (1) login (): 登录系统。
- (2) checkstu_card():对读者信息进行验证,检查是否符合本图书馆借书条件。
- (3) showinformation (): 显示该读者的基本信息函数。
- (4) borrow (): 读者借书函数。
- (5) getreaders (): 取得读者信息函数。看该读者是否符合借书条件,若符合,则返回可借信息。
- (6) gettitle ():取得书目信息。
- (7) getreservation (): 检验书籍是否被预订函数。
- (8) getnoreservation (): 书籍没被预订或取消预订函数。
- (9) create (borrower, item): 创建书籍外借函数。

借书时,读者先将书拿予管理员,管理员对书籍和读者进行检验,若书籍和读者都符合借书条件,则借书成功。



3.根据你的理解和设计构思，与该系统相关的系统如何描述？给出你对这些相关系统的描述。（6分）