



复旦托业
睿智所托·业则广阔

复旦托业 MSE
软件工程复习资料

MSE 测试题

一、名词解释

1. 软件能力成熟度模型 (CMM)

由美国卡内基·梅隆大学软件工程研究所提出的一个全面的过程元模型，当软件开发组织达到不同的过程能力和成熟度水平时，该模型可用来预测其所开发的系统和软件工程能力。

2. 极限编程 (XP)

属于敏捷过程模型的方法，使用面向对象方法作为推荐的开发范型，包含了策划、设计、编码和测试 4 个框架活动的规则和实践。其中包含一系列原则，如设计遵循 KIS 原则，鼓励单元测试，结对编程等。

3. 重构

以不改变代码外部行为而改进其内部结构的方式来修改软件系统的过程。这是一种净化代码以尽可能减少引入错误的严格方法。

4. 白盒测试和黑盒测试

白盒测试把测试对象看作一个透明的盒子，它允许测试人员利用程序内部的逻辑结构及有关信息，设计或选择测试用例，检查程序中所有逻辑路径是否都按预定的要求正确地工作。白盒测试又称为结构测试。黑盒测试是把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。黑盒测试又叫做功能测试。

5. 软件开发模型中的增量模型

增量模型将软件的开发过程分成若干个日程时间交错的线性序列，每个线性序列产生软件的一个可发布的“增量”版本，后一版本是对前一版本的修改和补充，重复增量发布的过程，直至产生最终的完善产品

6. 面向对象方法中的继承

继承，就是在每个由一般类和特殊类形成的一般-特殊结构中，把一般类的对

象实例和所有特殊类的对象实例都共同具有的属性和服务，一次性地在一般类中进行显式定义。在特殊类中不再重复地定义一般类中已定义的东西，特殊类却自动地、隐含地拥有它的一般类（以及所有更上层的一般类）中定义的全部属性和服务。

7. 高内聚—低耦合

一个模块内放置的相关各个元素彼此结合的紧密程度应该高；耦合是对一个软件结构内不同模块之间互连程度的度量，是模块之间接口的复杂程度，应该低。

8. 回归测试

在改正错误后或维护后要进行回归测试，即全部或部分地重复已做过的测试。

9. 软件过程中的用例驱动

用例获取系统的功能需求，它们“驱动”需求分析之后的所有阶段的开发。在分析阶段，它们被用于获取所需的功能并经客户的确认。在设计和实现阶段，用例必须被实现，在测试阶段，用例用于验证系统。

10. ER 图

实体-关系图简称为 ER 图，通常，使用 ER 图来建立数据模型。ER 图中包含了实体（即数据对象）、属性和关系等三种基本成分，其表示方法如下：

（1）实体。用矩形框表示实体。

（2）关系。用菱形框表示实体之间的关系。

（3）属性。用椭圆形或圆角矩形表示实体（或关系）的属性。

11. 模块化

模块化（Modularization）是把系统分割成能完成独立功能的模块，明确规定各模块及其输入输出规格，使模块的界面不会产生任何混乱。

12. α 测试和 β 测试

α 测试是由一个用户在开发者的场所来进行的，软件在开发者对用户的“指导”下进行测试，开发者负责记录错误和使用中出现的问题，所以， α 测试是在一个受控的环境中进行的。

β 测试是由软件的最终用户在一个或多个用户场所来进行的，用户记录下所有在 β 测试中遇到的（真正的或是想象中的）问题，并定期把这些问题报告给开发者，在接到 β 测试的问题报告后，开发者对系统进行最后的修改，然后开始准备向所有的用户发布最终的软件产品。

13. 软件构件

软件构件是一个独立发布的功能部分，可以通过其接口访问它的服务。它可以是被封装的对象类、一些功能模块、软件框架（framework）、设计模式（Pattern）等。

14. 面向对象的多态性

多态性是指同一个操作作用于不同的对象上可以有不同的解释，并产生不同的执行结果。

15. 数据字典

数据字典是描述数据信息的集合，是对系统中使用的所有数据元素的定义的集合。数据字典的作用是在软件分析和设计过程中提供数据描述，是数据流图必不可少的辅助资料。

16. 软件开发模型中的螺旋模型

B. Boehm 于 1988 年提出，是瀑布模型和演化模型的结合，并增加了风险分析。螺旋模型沿着螺线旋转，在四个象限上分别表达四个方面的活动，即：

制定计划：确定软件目标，选定实施方案，弄清项目开发的限制条件

风险分析：分析所选方案，考虑如何识别和消除风险

工程实施：实施软件开发

客户评估：评价开发工作，提出修正建议

17. Theo Mandel 界面设计三条“黄金规则”

（1）置用户于控制之中；（2）减少用户的记忆负担；（3）保持界面一致。

18. 信息隐蔽

在软件设计中，应该这样设计和确定模块，使得一个模块内包含的信息（过

程和数据)对于不需要这些信息的模块来说,是不能访问的。模块的局部化有利于实现信息隐蔽。采用信息隐蔽原理不仅支持模块的并行开发,而且有利于降低错误在软件其他部分的传播,减少测试和后期维护的工作量;

19. 内聚度 (cohesion)

一个模块内各个元素彼此结合的紧密程度用内聚来度量。理想的模块只完成一个功能,模块设计的目标之一是尽可能高的内聚。

20. 面向对象 (程序)

面向对象=对象 (object) +分类 (classification) +继承 (inheritance) +聚集+通过消息的通信 (communication with message)

其中: 1、对象是现实世界中个体或事物的抽象表示,是其属性和相关操作的封装。

2、类是一组具有相同属性和相同操作的对象的集合。一个类中的每个对象都是这个类的一个实例 (instance) .

3、继承是类间的基本关系,是基于层次关系的不同类共享数据和操作的一种机制,分为单一继承和多重继承。

4、消息传递是对象间通信的手段,一个对象通过向另一个对象发送消息来请求其服务,一个消息通常包括接受对象名、调用的操作名和适当的参数。消息完全由接受者解释

21. 验证与确认

验证指的是确保软件正确地实现了某一特定功能的一系列活动

确认指的是确保所建造的软件满足客户需求的一系列活动

22. 基线

基线是指已经通过正式评审和批准的某规约或产品 (一个版本), 可作为进一步开发的基础, 并只能通过正式的变更控制规程被改变。

23. 封装

封装 (encapsulation) 是一种信息隐蔽技术, 用户只能看见对象封装界面上的信息, 对象的内部实现对用户是隐蔽的。封装的目的是使对象的使用者和生产者分离, 使对象的定义和实现分开。

24. 软件复用

软件复用 (software reuse) 是指通过对已有软件的各种有关知识来建立新的软件, 这些知识包括: 领域知识、开发经验、设计经验、设计决定、体系结构、需求、设计、编码、测试和文档等。软件复用的目的是降低软件开发和维护的成本, 提高软件开发效率, 提高软件的质量。

25、 AOP

AOP 指一种程序设计范型。该范型以一种称为方面 (aspect) 的语言构造为基础, 方面是一种新的模块化机制, 用来描述分散在对象、类或函数中的横切关注点 (crosscutting concern)。从主关注点中分离出横切关注点是面向方面的程序设计的核心概念。分离关注点使得解决特定领域问题的代码从业务逻辑中独立出来, 易于重用和变更。

Ppt: 是目前软件开发中的一个热点, 可以通过预编译方式和运行期动态代理实现在不修改源代码的情况下给程序动态统一添加功能的一种技术。利用 AOP 可以对业务逻辑的各个部分进行隔离, 从而使得业务逻辑各部分之间的耦合度降低, 提高程序的可重用性, 同时提高了开发的效率。

26、 抽象

抽象是一种思考和解决问题的形式, 它集中注意事物某个一般性级别上的问题, 避开不必要的低层细节。抽象可以分成若干级别, 级别越高, 细节就越少。

27、 软件工程的 IEEE 定义: (1) 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, 即将工程化方法应用于软件。(2) 在 (1) 中所述方法的研究。

28、 单元测试 (unit testing) 也称模块测试, 一般在编码阶段进行, 测试对象针对模块 (功能和内部逻辑), 白盒测试方法。用于发现编码、详细设计阶段的错误。

- 29、 MVC 是一个设计模式，它强制性的使应用程序的输入、处理和输出分开。使用 MVC 应用程序被分成三个核心部件：模型、视图、控制器。它们各自处理自己的任务。视图是用户看到并与之交互的界面。模型表示企业数据和业务规则。控制器接受用户的输入并调用模型和视图去完成用户的需求。
- 30、 SOA：**面向服务的体系结构**（service-oriented architecture, SOA）是一个组件模型，它将应用程序的不同功能单元（称为服务）通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种这样的系统中的服务可以以一种统一和通用的方式进行交互。
- 31、 设计模式由 Erich Gamma 等人在 1990 年代从建筑设计领域引入到计算机科学的。它是对软件设计中普遍存在（反复出现）的各种问题，所提出的解决方案。设计模式描述了在某个特定场景与可能影响模式应用和使用方式的“影响力”中解决某个特定的设计问题的设计结构。OR
- 设计模式**是对面向对象设计中反复出现的问题的解决方案。一个设计模式命名、抽象和确定了一个通用设计结构的主要方面，这些设计结构能用来构造可复用的面向对象设计。确定了所包含的类和实例，他们的角色、协作方式以及职责分配。
- 32、 **喷泉模型**是一种支持面向对象开发的模型，是一种以用户需求为动力，以对象为驱动力的模型，主要用于描述面向对象的软件开发过程。该模型认为软件开发过程自下而上周期的各阶段是相互重叠和多次反复的。
- 33、 Web **服务**是一套标准，它定义了应用程序如何在 Web 上实现互操作性，是利用一组基于 web 的标准实现的服务，是实现 SOA 的方式之一。目标是即时装配、松散耦合以及自动集成。

二、简答题

- 1、软件过程的经典生命周期分为哪几个阶段，各个阶段的主要任务是什么？

(1) 计算机系统工程的任务是确定待开发软件的总体要求和范围，以及与之有关

的硬件、支持软件的要求。

(2) 可行性分析从经济、技术、法律等多个方面分析待开发的软件是否有可行的解决方案，并在若干个可行的解决方案中作出选择。

(3) 软件项目计划确定待开发软件的目标，并对资源分配、成本估算、进度安排等作出合理的计划。

(4) 软件需求分析确定待开发软件的功能、性能、数据、界面等要求。

(5) 软件设计分为系统设计和详细设计解决待开发软件“怎么做”的问题。

(6) 编码用某种程序语言为软件编写程序。

(7) 测试发现并纠正软件中的错误和缺陷。

(8) 运行和维护在软件运行期间对软件进行修改和维护。

4、简述 Theo Mandel 所提出的软件界面设计的三条“黄金规则”？

(1) 置用户于控制之中

(2) 减少用户的记忆负担

(3) 保持界面一致

5、请简述 CMM 的概念以及五级标准的含义。

CMM 模型是用来确定一个软件过程的成熟程度以及指明如何提高过程成熟度的参考模型。在 CMM 中把软件过程分成五个阶段，并把这些阶段排序，形成五个逐层提高的等级。

1. 初始级（混沌的软件过程）

软件过程的特征是无序的，有时甚至是混乱的。几乎没有什么过程是经过定义的，项目能否成功完全取决于个人能力。

2. 可重复级（经过训练的软件过程）

建立了基本的项目管理过程，以追踪成本、进度和功能性。必要的过程规范已经建立起来了，使得可以重复以前类似项目所取得的成功。

3. 已定义级（标准、一致的软件过程）

用于管理和工程活动的软件过程已经文档化和标准化，并且已经集成到整个组织的软件过程中。所有项目都使用文档化的、组织批准的过程来开发和维护软件。

4. 定量管理级（可预测的软件过程）

已收集了软件过程和产品质量的详细度量数据，使用这些详细的度量数据，能够定量地理解和控制软件过程和产品。

5. 优化级（能持续改善的软件过程）

通过定量的反馈能够实现持续的过程改进，这些反馈是从过程及对新想法和技术的测试中获得的。这一级包含了第4级的所有特征。

处于第5级的组织的目标是持续地改进软件过程。

8、敏捷软件开发宣言是轻量级开发方式的宣言，宣言强调了它和重量级开发方式的不同，请简要叙述宣言的内容，并概述其中一个敏捷过程模型。

个体和交互胜过过程和工具；
可工作软件胜过宽泛的文档；
客户合作胜过合同谈判；
响应变化胜过遵循计划。
右边的各项很有价值，但左边的价值更大。

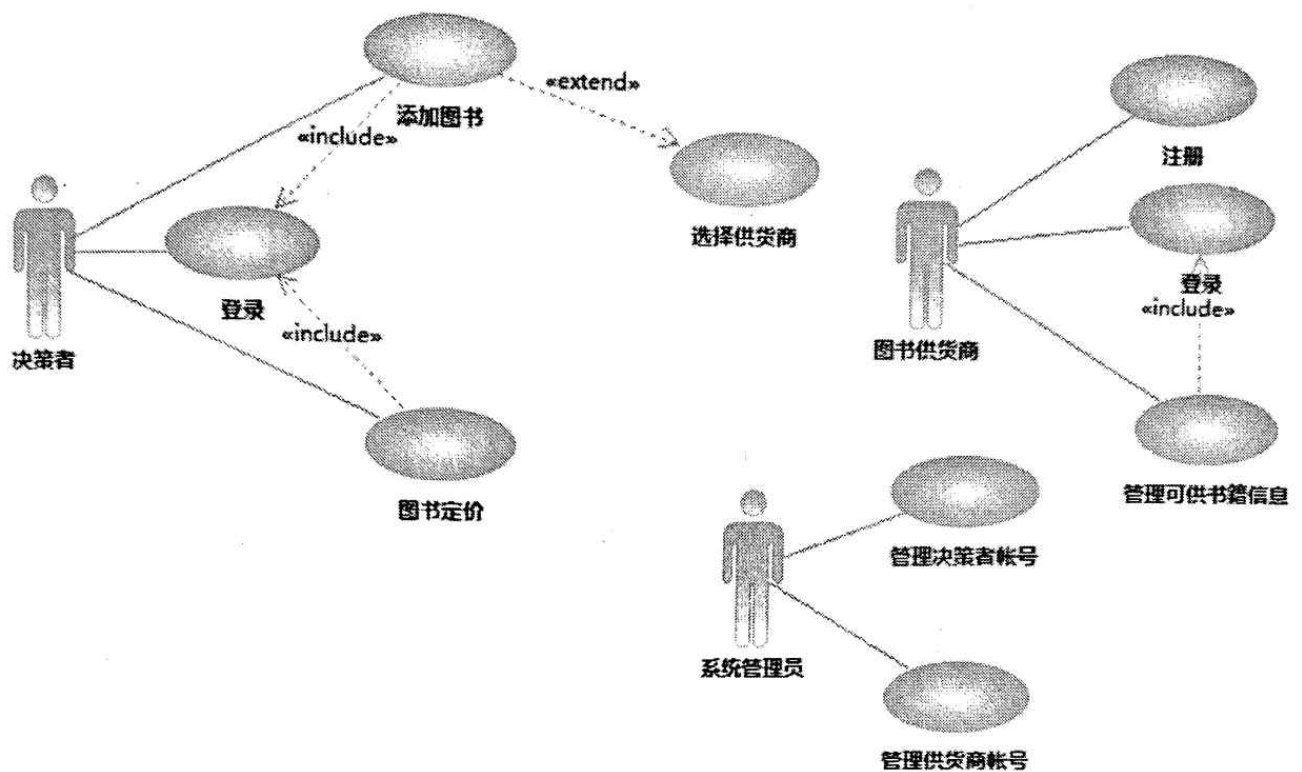
极限编程（Extreme Programming, XP）属于一种敏捷开发方式，其特征有，生命周期：策划、设计、编码、测试。XP 原则：现场客户、简单设计、测试驱动、结对编程、代码全体拥有、持续集成、小型发布。

10、根据以下对网络书店采购、定价管理信息系统需求的简要描述，区分其中的用户和系统功能，使用UML 画出系统用例图：

系统功能简述如下，“图书供货商”注册被审批后，可以登录系统发布其可提供的书籍信息；网络书店经营“决策者”登录后可以添加待购书，其中采购时可以选择以前系统中注册过的“图书供货商”作为供货渠道；另外，还可以为图书定价。“系统管理员”进行账户管理。

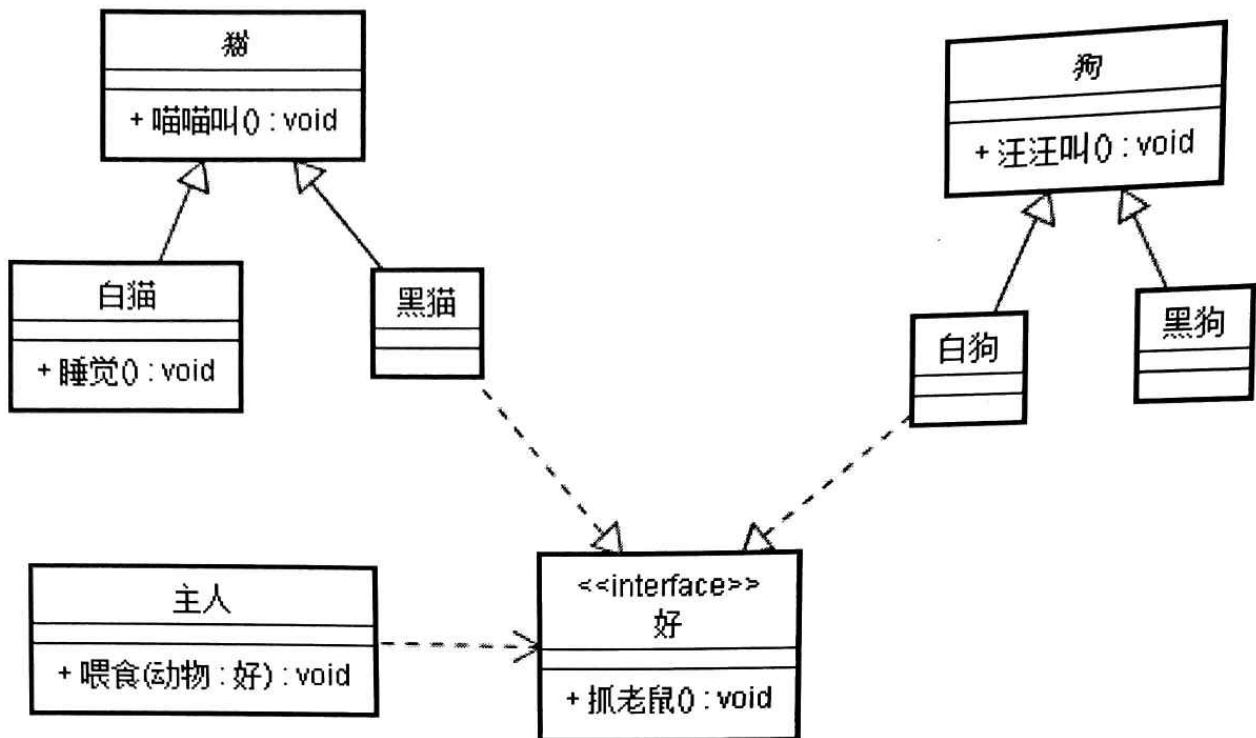
具体来说，本系统提供给供应链上的“图书供货商”登录后输入可供货图书商品信息；系统还提供网络书店经营“决策者”添加待购图书功能，其中可选的输入图书采购渠道，其采购渠道可以根据“图书供应商”在注册后输入的供书信息进行选择；另外，还可以进行图书价格确定等功能；“系统管理员”可以审批注册的“图书供应商”，日后也可以删除该供应商帐号；同时，为“决策者”添加和删除相应账户。“决策者”无需注册，直接获得帐号和密码，可以在系统中修

改密码。



13、假定以下场景：猫分为白猫和黑猫，都会喵喵叫，但是黑猫还能抓老鼠，白猫不抓老鼠只会睡觉；狗也分为白狗和黑狗，都会汪汪叫，而白狗还会多管闲事抓老鼠；无论猫和狗，会抓老鼠的就是好的，主人认为会抓老鼠的动物就是好动物，在喂食的时候只给会抓老鼠的猫和狗喂食。请用面向对象的观点分析以上场景，并用 UML 里的类图来描绘其中的类以及之间的关系。

提示：主人，猫，狗都将出现在类中；可以采用抽象类，继承以及接口的设计。



14、请阐述面向对象程序设计的三个基本特征：封装性、继承性、多态性的含义。并从程序设计的角度阐述面向对象与面向过程的区别。

封装是一种信息隐蔽技术，用户只能看见对象封装界面上的信息，对象的内部实现对用户是隐蔽的。封装的目的是使对象的使用者和生产者分离，使对象的定义和实现分开。

继承是类间的基本关系，它是基于层次关系的不同类共享数据和操作的一种机制。

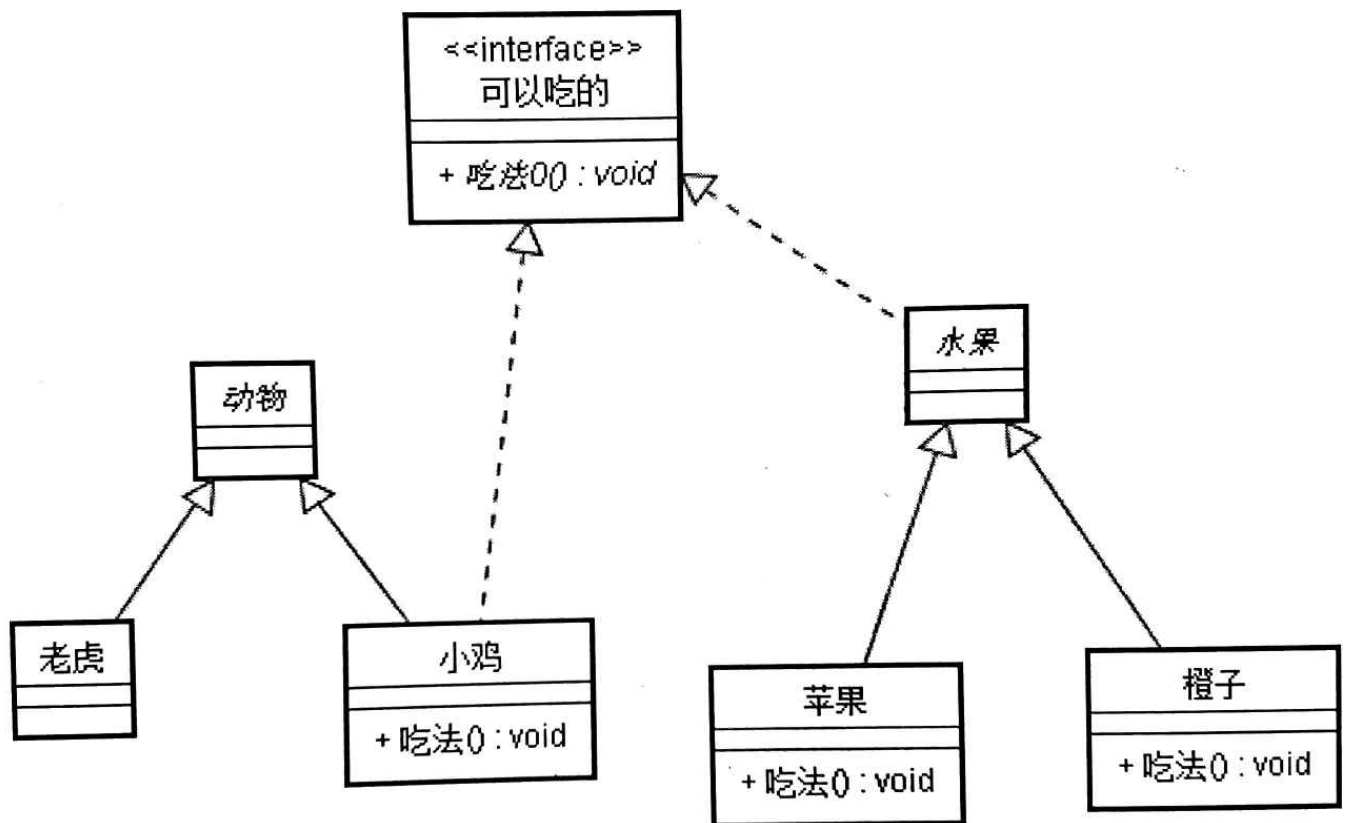
多态性是指同一个操作作用于不同的对象上可以有不同的解释，并产生不同的执行结果。

面向对象的程序设计更加以数据为中心，将状态和行为封装为类和对象。Coad 和 Yourdon 给出了一个定义：面向对象=对象+分类+继承+基于消息的通信。

面向过程更多的以功能和步骤为中心。面向过程的程序设计主要是数据结构+算法。

16、给出以下阐述：**小鸡**和**老虎**都是**动物**，其中小鸡是可以**油煎吃**的，而老虎不能吃。**水果**都可以**鲜吃**，其中**苹果**可以**做成苹果汁吃**，**橙子**可以**做成橙汁吃**。请用面向对象的观点分析以上阐述，并用 UML 里的类图来描绘其中的类、接口以及之间的关系。

提示：文中粗斜体可以考虑设计成类和接口，以及其中的方法；可以采用抽象类，继承以及接口的设计。



18、给出 UML 的 9 种典型的视图名字，并且将其列在其属于的相应的视图分类右边。

请列出 9 种 UML 视图名字在右边：

将上述 UML 视图列在下面分类视图的右边

用例视图：

结构视图：

行为视图：

实现视图：

环境视图：

答案：

用例视图：用例图

结构视图：类图、对象图

行为视图：状态图、活动图、顺序图、协作图

实现视图：构件图

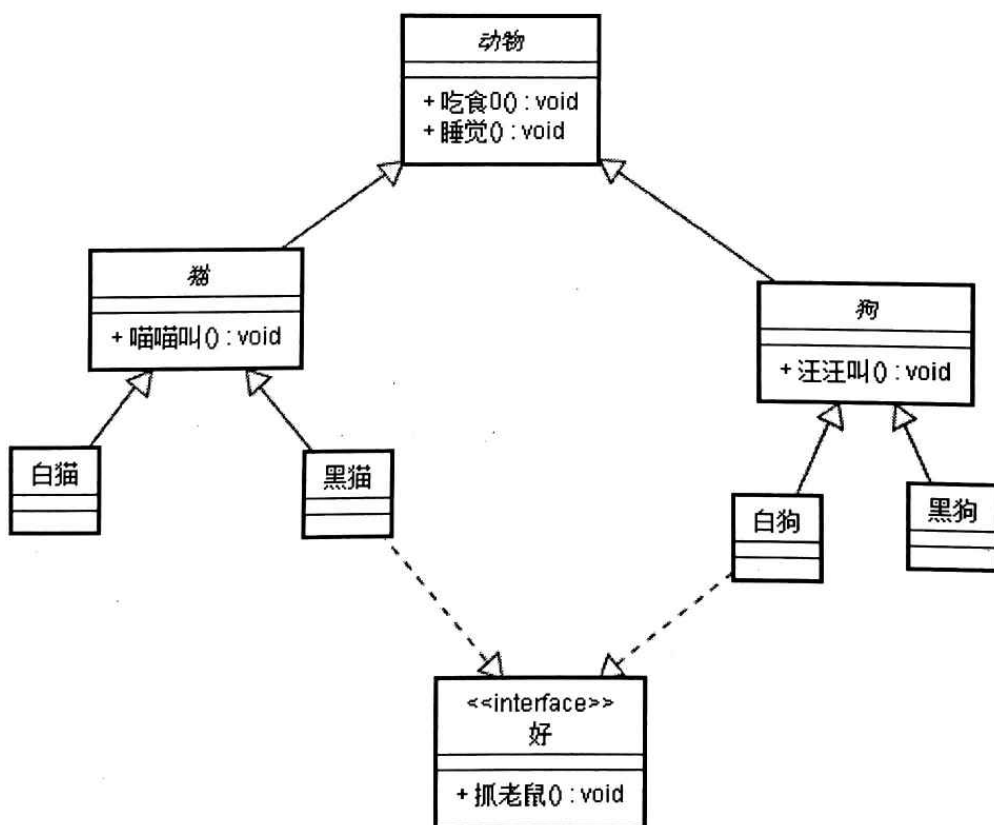
环境视图：部署图

19、请简要比较面向过程开发和面向对象开发。

面向过程的开发更多的以功能和步骤为中心。面向过程的程序设计主要是数据结构+算法，一般应用于较为简单和底层的系统开发；**面向对象**的开发更加以数据为中心，将状态和行为封装为类和对象。oad 和 Yourdon 给出了一个定义：面向对象=对象+分类+继承+基于消息的通信，一般应用于较为复杂和数据为中心的开发。

21、**动物**都能**吃**以及**睡觉**；**猫**和**狗**都属于**动物**，猫分为**白猫**和**黑猫**，都会**喵喵叫**，但是黑猫还能**抓老鼠**，白猫不抓老鼠只会睡觉；狗也分为**白狗**和**黑狗**，都会**汪汪叫**，而白狗还会多管闲事抓老鼠；无论猫和狗，会抓老鼠的就是**好的**。请用 UML 里的类图来描绘其中的类以及之间的关系。

提示：动物，猫，狗都将出现在类中；可以采用继承以及接口的设计。



22. 请给出 MVC（模型-视图-控制器）架构的含义以及优势。

MVC 是一个设计模式，它强制性的使应用程序的输入、处理和输出分开。使用 MVC 应

用程序被分成三个核心部件：模型、视图、控制器。它们各自处理自己的任务。视图是用户看到并与之交互的界面。模型表示企业数据和业务规则。控制器接受用户的输入并调用模型和视图去完成用户的需求。

其优势在于：在模型层、视图层和控制层之间划分责任可以减少代码的重复度，并使应用程序维护起来更简单；由于数据和商务逻辑的分开，在新的数据源加入和数据显示变化的时候，数据处理也会变得更简单；实现了功能模块和显示模块的分离，同时它还提高了应用系统的可维护性、可扩展性、可移植性和组件的可复用性。

三、分析题

一、对你参与开发的一个软件项目作简介，然后结合这个项目，阐述以下相关方面

：

A、采用的软件过程有哪些阶段，介绍这些阶段的任务、所花时间等情况，有何经验？（5分）

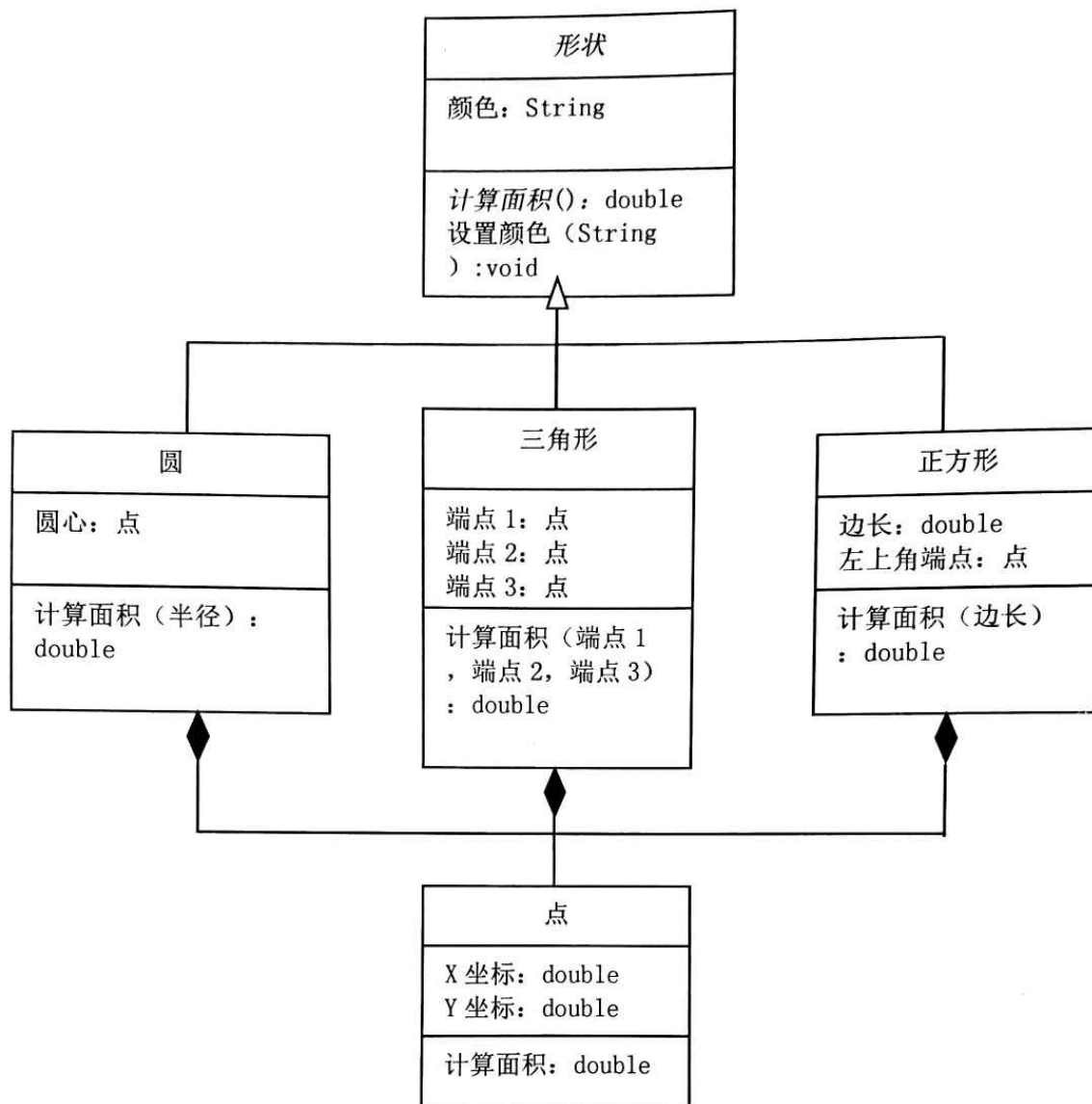
B、使用的开发平台、语言以及开发方法是什么？选择的依据是什么？（5分）

C、采用了哪些测试措施，你认为他们分别属于什么测试？（5分）

二、用 UML 的类图分别表示下列描述中的实体（带下划线的斜体字）的属性和行为，以及类之间的关系（关联，泛化，聚集或者组合聚集等），并简单描述你的设计思路和理由，不需要给出具体方法的实现。（15分）

在一个二维绘图系统中，具有一些形状，包括圆，三角形和正方形，该绘图系统显示它们的时候，可以采用不同的颜色表现，同时，可以计算它们的面积，其中将分别根据点，半径，边长等信息进行计算。由于这些形状都要用到点，所以也将其中点设计为一个类，采用点的 x, y 坐标描述。

提示：你将根据这段描述采用 UML 进行简单的面向对象的分析和设计，分辨出系统中需要设计哪些类，类需要具有哪些属性和方法，类之间的关系如何，如何设计高层类，哪些属性和方法应该放在高层类中，哪些类和方法应该设计为抽象的，哪些方法需要重写。



练习题：订货系统

假设一家工厂的采购部每天需要一张定货报表，报表按零件编号排序，表中列出所有需要再次定货的零件。对于每个需要再次定货的零件应该列出下述数据：零件编号，零件名称，定货数量，目前价格，主要供应者，次要供应者。零件入库或出库称为事务，通过放在仓库中的CRT终端把事务报告给定货系统。当某种零件的库存数量少于库存量临界值时就应该再次定货。

要求进行需求分析，并分别画出0层、1层和2层数据流图。

练习题：订货系统

数据流图成分：源点或终点，处理，数据存储和数据流。

- 第一步可以从问题描述中提取数据流图的4种成分：

- ✓ 首先考虑数据的源点和终点：

- 仓库管理员是数据源点

- 采购员是数据终点

- ✓ 接下来考虑处理：

- 用于产生报表的处理

- 对事务的加工

- ✓ 考虑数据流和数据存储

- 定货报表、事务

练习题：订货系统

源点/终点	处理
仓库管理员 采购员	产生报表 处理事务
数据流	数据存储
定货报表 零件编号、名称 定货数量、目前价格 主要供应者、次要供应者 事务 零件编号、事务类型、数量	定货信息 （见定货报表） 库存清单* 零件编号* 库存量 库存量临界值

练习题：订货系统

- 第二步：画出基本系统模型

任何计算机系统实质上都是信息处理系统，也就是说计算机系统本质上都是把输入数据变换成输出数据。

因此，任何系统的基本逻辑模型都由若干个数据源点/终点以及一个处理组成，这个处理就代表了系统对数据加工变换的基本功能。

练习题：订货系统

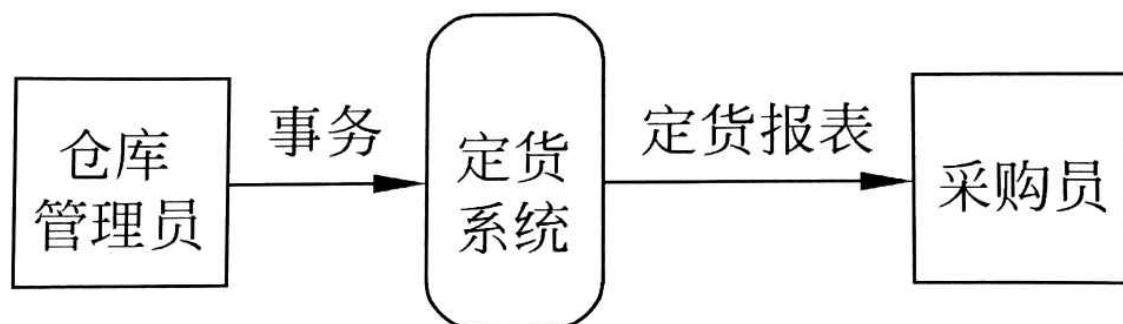


图2.5 定货系统的基本系统模型

练习题：订货系统

- 第三步：画出系统的功能级数据流图
把基本系统模型细化，描绘系统的主要功能。
- ✓ “产生报表”和“处理事务”，代替“定货系统”。
- ✓ 增加两个数据存储
 - 处理事务需要“库存清单”数据；
 - 一天只产生一份报表，因此需要存储“定货信息”
- ✓ 增加两个数据流，与数据存储相同
- ✓ 给处理和数据存储加编号

练习题：订货系统

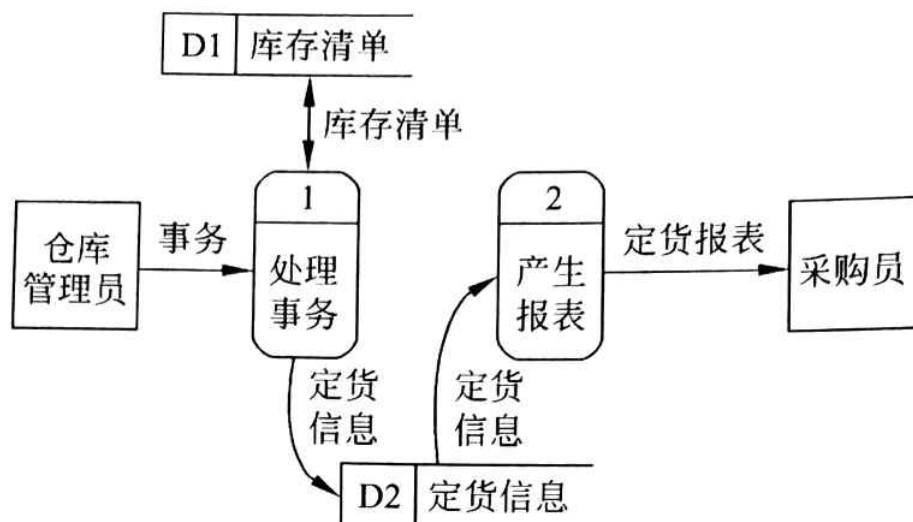


图2.6 定货系统的功能级数据流图

练习题：订货系统

- 第四步：对功能级数据流图中描绘的系统主要功能进一步细化

考虑通过系统的逻辑数据流：

- 当发生一个事务时必须首先接收它
- 随后按照事务的内容修改库存清单
- 最后如果更新后的库存量少于库存量临界值时，则应该再次定货，也就是需要处理定货信息。

把“处理事务”这个功能分解为下述3个步骤：

“接收事务”、“更新库存清单”和“处理定货” (图2.7)。

注意：当对数据流图分层细化时必须保持信息连续性，也就是说，当把一个处理分解为一系列处理时，分解前和分解后的输入输出数据流必须相同。

练习题：订货系统

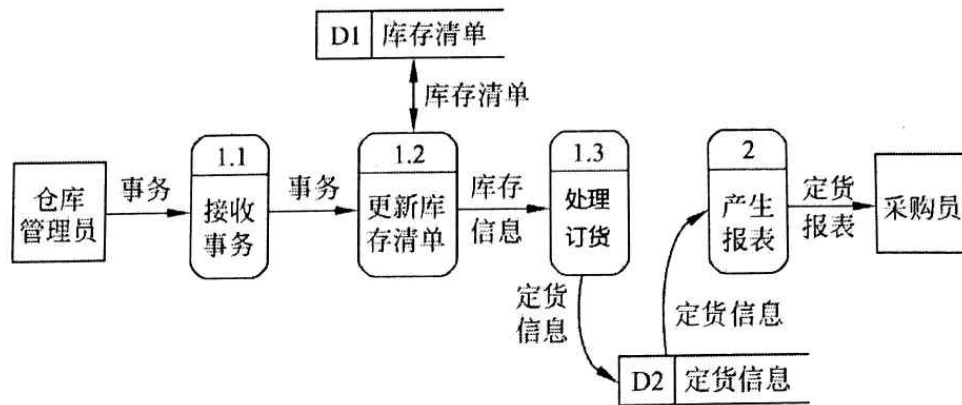


图2.7 功能进一步分解后的数据流图