第一章

1、请谈一下关于软件工程的整体认识和印象。

软件工程终究是服务于人的需求的，编码是实现需求的手段。软件工程就是按照工程学的管理方式，有组织、有计划的，在一定的质量基础、时间限度和成本范围内，实现功能明确的软件系统。而且，软件工程在企业范围内运行，一定需要企业资源的支持，要与企业的经营、决策、管理体系联系在一起，才能够被踏踏实实的落实下来。软件工程是一门指导软件开发和维护的工程学科，主要研究软件开发和维护的工程技术和软件项目管理等内容，其中涉及如何保证软件产品的质量和可靠性，如何提高软件开发效率和拥护满意度等。生产具有正确性、可用性以及开销合宜的产品。正确性指软件产品达到预期功能的程度。可用性指软件基本结构、实现及文档为用户可用的程度。开销合宜是指软件开发、运行的整个开销满足用户要求的程度。这些目标的实现不论在理论上还是在实践中均存在很多待解决的问题，它们形成了对过程、过程模型及工程方法选取的约束。因此，软件工程是系统地对需求的分析，服务于这个需求的系统的设计，系统的实现，以及后续的维护的总和。因此软件工程呈现出与其它的工程的共性，而不是因为编码这一实现手段的特殊而与其它类型的工程出现本质不同。

2、系统分析师和系统构架设计师有何区别?

系统分析师的工作位于系统架构设计师上层。

系统分析师要负责需求分析、系统设计等位于系统设计及以上层次的任务。相比系统架构设计师，他更贴近业务。

系统架构设计师则更贴近技术，他负责设计技术模型来解决系统分析师提出的目标和问题。

3、应用软件工程(实施分阶段原理等)会增加系统工作量吗?

使用规范化的软件工程流程必然要为此耗费设计、协调等等工作量，同时降低许多潜在的开发隐患所带来的工作量。而这种开发隐患往往随着项目的增大而增大，所以对于大型项目，应用软件工程实质上是在规避隐患，事半功倍；小型项目则需要权衡利弊。

练习题2.给出一个问题分析的例子,其中问题部分相对简单,但是解决问题的困难在于子问题之间的相互联系。

安排火车时间表。各个火车几点开车几点到达是很容易决定的，但是协调不同火车之间的冲突十分复杂。

练习题3.解释错误、故障和失效之间的区别。举出一个关于错误的例子，并且这个错误导致了需求、设计、代码的故障。举出在需求中存在故障的例子，并且这个故障导致了失效。举出在设计中存在故障的例子，并且这个故障导致了失效。举出在测试数据中存在故障的例子，并且这个故障导致了失效。

设计、代码的故障。举出在需求中存在故障的例子，并且这个故障导致了失效。举出在设计中存在故障的例子，并且这个故障导致了失效。举出在测试数据中存在故障的例子，并且这个故障导致了失效。

错误是由于各种原因而在设计和编码阶段产生的，故障是错误导致的不正确代码、不正确文档等留存在项目中的问题，失效则是在系统在运行时违背了其应有的行为。

第二章

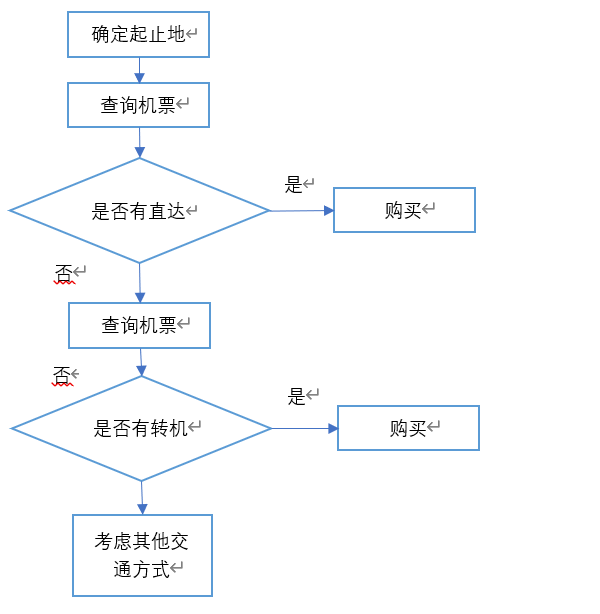
1、如何称得上一名优秀的程序员?

一个优秀的初级的程序员应该能够较好地完成一个完整过程中的某一部分活动。随着层次的逐渐提升，一个优秀的程序员能掌控的活动应当越来越多，并且对掌控的各个部分能够协调，并自如地让自己在各个不同部分的工作中灵活切换。

2、系统架构师应该具备的素质是什么?

系统架构师应该能根据选择的过程模型的不同设计不同的系统。比如如果使用增量的阶段式开发，那么就应当注重系统的可扩展性，使用迭代的阶段式开发就应该注重系统的大致功能是否全面。

练习题4.画一个图，试描述为一次商务旅行购买一张飞机票的过程。



练习题11.考虑本章介绍的过程。哪些过程在你对需求变化做出反应时给了你最大的灵活性?

阶段化开发中的增量开发在需求变化时具有最大的灵活性。

第三章

1、如果一个案例中涉及到合同管理，项且管理控制和项目沟通等诸多方面，在项目实际运行过程中，出现了甲方随意变更、不配合验收、甲乙双方沟通存在障碍.等情形，试问如何从合同管理、过程控制和项目沟通管理三个方面来应对?

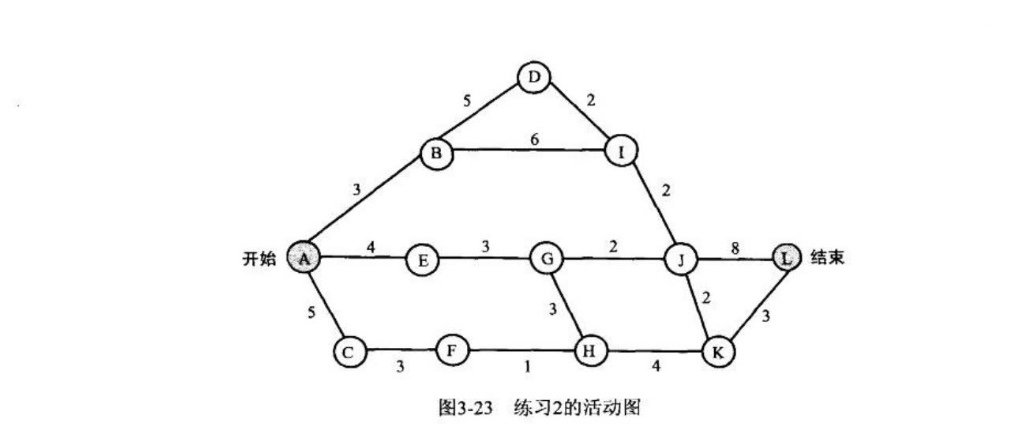
我已乙方（也就是我）的角度来思考问题

合同管理阶段：应当在合同中就说明确定需求的时间点和验收流程，并且规定改需求和更改验收流程所需要的付出的代价

项目管理阶段：写好文档，确定好里程碑，并且可以考虑分阶段验收

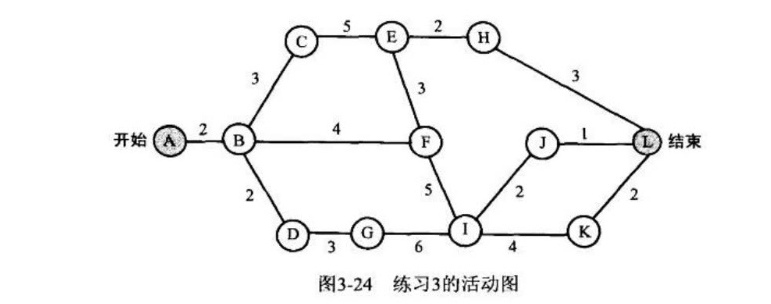
项目沟通：应当量化工作进度，让作为外行的甲方可以无疑义地确定工作进度

练习题2.图3-23是一个软件开发项目的活动图。对应于图中每条边的数字表示完成这条边代表的活动所需的天数。例如，完成终止于里程碑E的活动需要4天时间。对于每个活动，列出它的前驱，并计算最早开始时间、最晚开始时间和时差。然后确定出关键路径。





练习题3.图3-24是一张活动图。找出其关键路径



关键路径是：A-B-C-E-F-I-K-L，长度为24

练习题12.很多项目经理根据过去项目中程序员的生产率来计划项目的进度，生产率通常根据单位时间的单位规模来测量。例如，一个组织机构可能每天生产300行代码或每月生产1200个应用点。用这种方法测量生产率合适吗?根据下列事项讨论生产率的测度:

●用不同的语言实现同样的设计，可能产生的代码行数不同。

●在实现开始之前不能用基于代码行的生产率进行测量。

●程序员可能为了达到生产率的目标而堆积代码。

首先，以生产代码行数是肯定不合理的。首先要说的是，它有一个根本问题，就是软件开发的过程中，编写代码只是一部分。比如：完成同样的需求，好的设计可以降低代码量，但我们不能说它的工作量就小了，因为为了做出这样的设计，程序员可能要付出更多的精力来做好设计。

应用点计数会好一些，但是我认为应用点也是有难有易的，一味地使用数量作为衡量标准也有些不妥。还是应该以更科学的方法。

第四章

1、如何拒绝需求分析过程中某些不合理的用户需求?

我认为首先要分类

-技术上不现实的需求：以专业角度否定不合理的需求，因为这种问题是没有余地的

-可实现但是从我方角度来讲不合适：面对这种情况时，其实是不能确定需求最终的合理性的，应当以需求为唯一准绳，不能涉及技术方面。

2、分析阶段类的大致种类有边界类、实体类、控制类。请简述其具体含义(包括画法)

边界类：两个对象之间彼此交互使用的接口，系统与外界交互的门户

实体类：系统中的实体，存储系统中具体数据的类

控制类：将来自用户的操作转化为对系统中其他类的操作的类

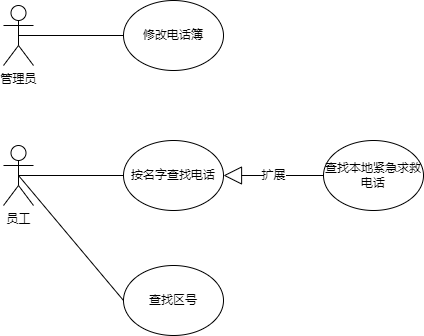
3、以下三题任选其一用UML描述格式

练习题12.针对用联机电话号码簿替换电话公司给你的号码簿这样的问题，编写一套UML模型(用例图、MSC图、类图)。给出名字时，要求该号码簿应该能够提供电话号码;它还应该能够列出国家不同地区的区号，并给出你所在地区的紧急求救电话号码。

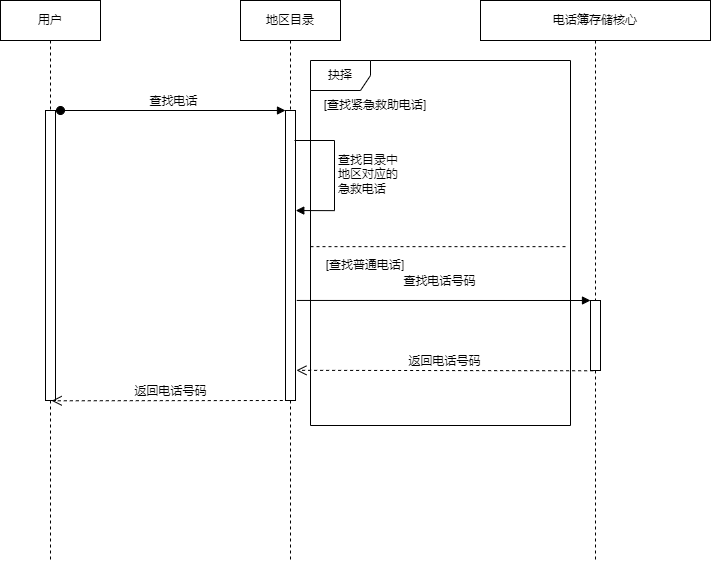
练习题17.为讲座调度系统编写一个Z规格说明。系统保存哪一个演讲者将在哪一天进行演讲的相关记录。对每一个演讲者，不应该安排多于一次的演讲;任何一天不能安排多于4次的演讲;应该提供从安排中增加或删除演讲的操作，以及下列操作:交换两个演讲的日期，列出安排在特定日期的演讲，列出安排特定演讲者演讲的日期，以及向每一个演讲者发送关于其演讲日期的提示信息。你可以定义任何有助于简化规格说明的额外操作。

网上实例1.小米校招产品作业解读:设计一款网络日记APP。(画像课堂作业)

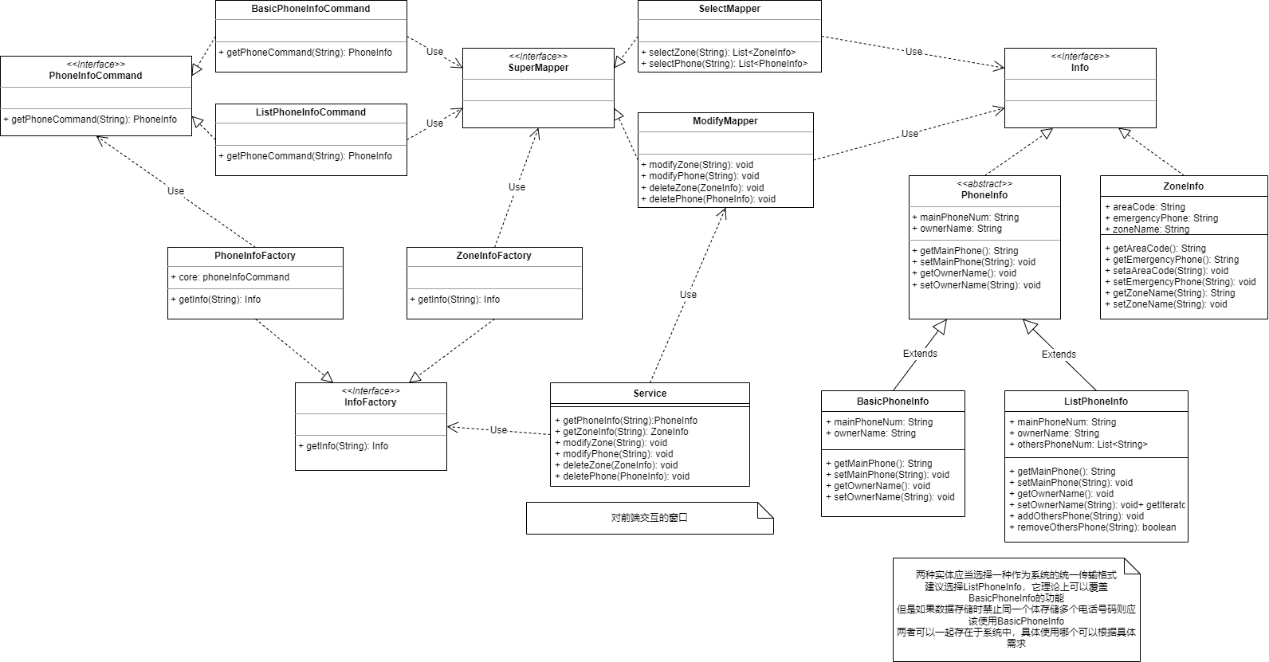
用例图



MSC图-查询号码的过程



类图



第五章

练习题4. 给出一个用原型开发但并没有节省大量开发时间的例子。

高精运算的软件，各个运算逻辑紧密耦合，难以使用原型开发进行分块原型化，整体原型化又会使原型过程过于冗长

练习题5. 列举出最适合原型设计的系统的特征。

- 系统可以被分为比较小的模块，分开攻坚

- 系统有不确定之处，比如有多种方案需要权衡，或有未确定的功能需要测试

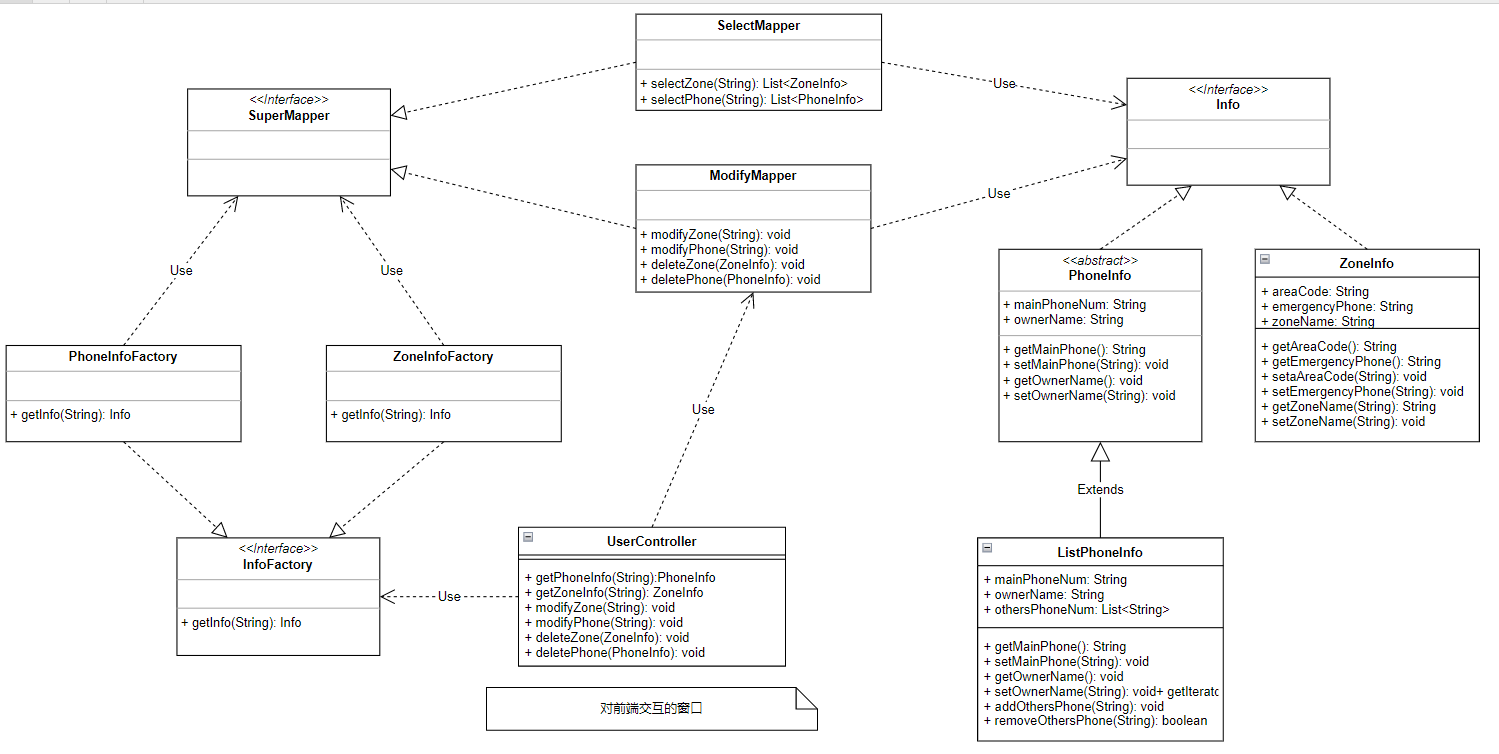
2、上一章第12题的延续:继续进行设计阶段的工作。

或者:上一章第17题的延续:继续进行设计阶段的工作。

或者:上一章小米校招的延续:继续进行设计阶段的工作。

12续

分块介绍类的设计，和最开始的类图已经有不同了



DAO：

<<Interface>>SuperMapper：没有实际内容，所有DAO层对象的接口，我认为各个Mapper间可能会在将来有需要实现的共性，所以预先设计了

SelectMapper implements SuperMapper：用于实现各种查询方法，不允许有任何修改操作

方法：

+ selectZone(String): List<ZoneInfo>

选择区域信息

+ selectPhone(String): List<PhoneInfo>

选择电话信息

ModifyMapper implements SuperMapper：用于实现各种修改方法，不允许有任何查询操作

方法：

+ modifyZone(ZoneInfo): void

修改区域信息，如果不存在则新增，存在则修改

+ modifyPhone(PhoneInfo): void

修改电话信息，逻辑同上

+ deleteZone(ZoneInfo): void

删除区域信息

+ deletePhone(PhoneInfo): void

删除电话信息

POJO：

统一使用lombok，具有get、set等方法

<<Interface>>Info：没有实际内容，所有POJO对象的接口，理由同SuperMapper

<<abstract>>PhoneInfo implements Info：抽象类，作为后续电话信息类的父类(当然，后续可以有结构更复杂的存储类)

属性：

+ mainPhoneNum: String

主电话号码

+ ownerName: String

电话的主人

ListPhoneInfo extends PhoneInfo：具有一个List，存储主电话主人所具有的其他非主要号码

属性：

+ mainPhoneNum: String

+ ownerName: String

+ othersPhoneNum: List<String>

存储更多电话的List

方法：

+ addOthersPhone(String): void

+ removeOthersPhone(String): Boolean

ZoneInfo implements Info：存储这个城市的区号、急救电话、名称

属性：

+ areaCode: String

区号

+ emergencyPhone: String

急救电话

+ zoneName: String

城市名

工厂：

<<Interface>> InfoFactory：获取Info对象的所有工厂的接口

方法：

+ getInfo(String): Info

规定的获取信息的方法

PhoneInfoFactory implements InfoFactory：获取电话信息的工厂，依赖于Mapper实现getInfo方法

ZoneInfoFactory implements InfoFactory：获取区域信息的工厂，依赖于Mapper实现getInfo方法

Controller层：

使用springboot

UserContoller：提供给用户，用于获取信息。get方法依赖于Factory获取信息并返回，但剩下的修改方法目前还直接依赖于mapper

方法：

+ getPhoneInfo(String):PhoneInfo

+ getZoneInfo(String): ZoneInfo

+ modifyZone(String): void

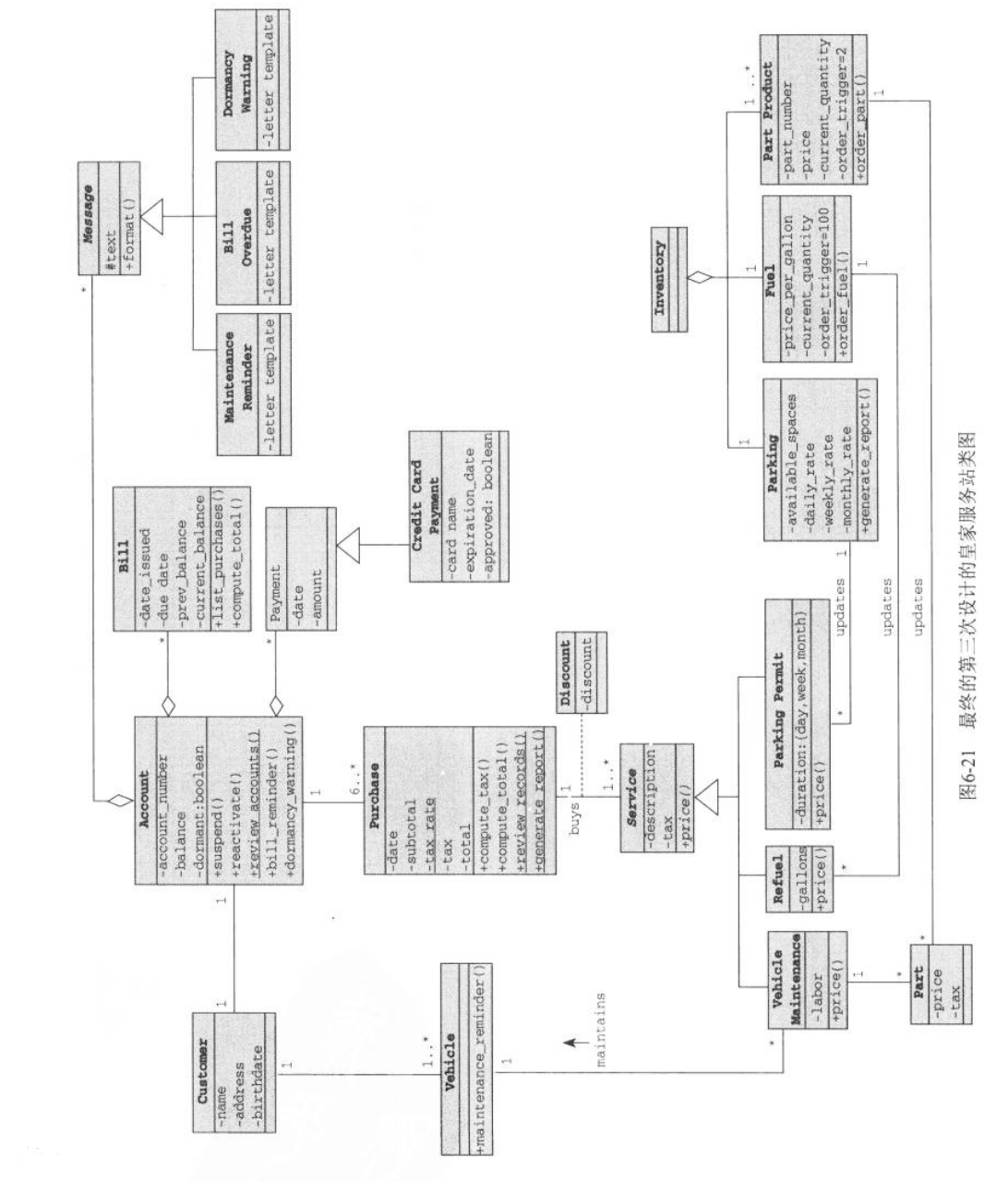
+ modifyPhone(String): void

+ deleteZone(ZoneInfo): void

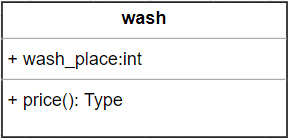
+ deletePhone(PhoneInfo): void

第六章

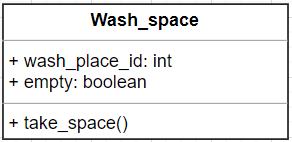
练习题1. 皇家汽车服务站经理Manny准备扩展他们的服务，使之包括洗车服务。这将是一个自动化系统。客户选择洗车的方式并注明车的类型。系统计算费用并在控制面板上显示应付金额。然后客户支付洗车费。支付结束后，如果洗车正忙的话，系统就会提示客户必须等待，否则，系统提示客户将车驶进洗车间。请问，如何对图6-21所示的类图进行修改才能提供这种新服务。



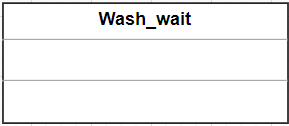
增加service的子类Wash，表示洗车服务



增加Inventory下的类Wash\_space，表示洗车位



增加message的子类Wash\_wait，表示洗车等待信息



2、 learn about the useness of all UML diagrams

用例图(Use Case Diagram)是由软件需求分析到最终实现的第一步，它描述人们如何使用一个系统。用例视图显示谁是相关的用户、用户希望系统提供什么样的服务，以及用户需要为系统提供的服务，以便使系统的用户更容易理解这些元素的用途，也便于软件开发人员最终实现这些元素。用例图在各种开发活动中被广泛的应用，但是它最常用来描述系统及子系统。

当用例视图在外部用户出现以前出现时，它捕获到系统、子系统或类的行为。它将系统功能划分成对参与者（即系统的理想用户）有用的需求。而交互部分被称作用例。用例使用系统与一个或者多个参与者之间的一系列消息来描述系统中的交互。

用例图包含六个元素，分别是：参与者(Actor)、用例(Use Case)、关联关系(Association)、包含关系(Include)、扩展关系(Extend)以及泛化关系(Generalization)。

用例图可一个包含注释和约束，还可一个包含包，用于将模型中的元素组合成更大的模块。有时，可以将用例的实例引入到图中。用例图模型如下所示，参与者用人形图标来标识，用例用椭圆来表示，连线表示它们之间的关系。

3、 learn about how to design a rough class diagram of a small scale project.

类图一般包含3个组成部分。第一个是类名；第二个是属性（attributes）；第三个是该类提供的方法（ 类的性质可以放在第四部分；如果类中含有内部类，则会出现第五个组成部分）。类名部分是不能省略的，其他组成部分可以省略。

类名书写规范：正体字说明类是可被实例化的，斜体字说明类为抽象类。

属性和方法书写规范：修饰符 [描述信息] 属性、方法名称 [参数] [：返回类型|类型]

属性和方法之前可附加的可见性修饰符：

如果属性或方法具有下划线，则说明它是静态的。

类的性质是由一个属性、一个赋值方法和一个取值方法组成。

没有类是单独存在的，他们通常和别的类协作，创造比单独工作更大的语义。因此，除了捕获系统的词汇以外，还要将注意力集中到这些类是如何在一起工作的。使用类图来表达这种协作。

第八章

1、 模拟面试问答:如果让您来带领一个测试团队，您会做哪些工作?

组织通常将其业务目标,具体落实到每个团队和个人身上，从而进行高层级管理。当提出需求文档时,测试领导和他的团队的测试人员就开始着手去写测试计划，包括定义范围,硬件,软件,功能测试,进度等。在所需的努力和适当的工作分配的基础上进行大小评估。经常情况下，测试领导人有意突出他们的团队技能,强调他们的团队忙得团团转。现在，忙的团团转的可以意味着很多领域的事情。沟通不管是在电子邮件、电话或在人际关系上都应该排在第一位,而不是说——把最好的留到最后。

-分析需求，根据需求、项目规模规划整体测试计划，规定测试和文档的规范

-为不同的部分规划不同的测试方法：是否用自动化测试？黑盒or白盒？集成的顺序？

-根据重要程度的不同安排人手，并投入到最重要的部分的测试工作

2、 对于需求文档，如何进行测试? .(从什么方面考虑?有什么样的原则? )

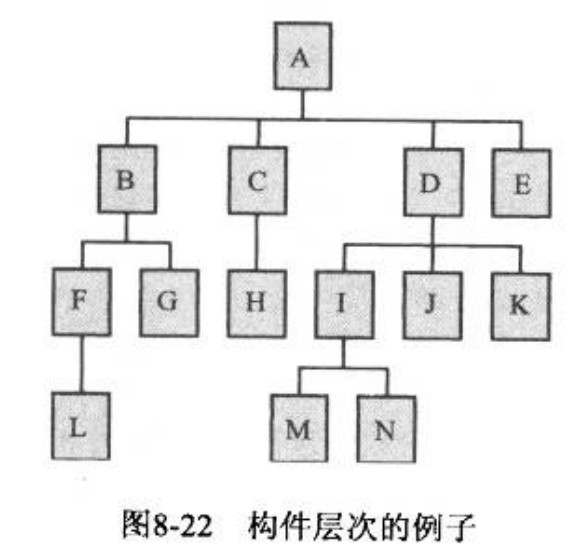
-根据需求文档本身确定流程图、输入域

-“小”用白盒，“大”用黑盒

-抽离地看待系统，我们的目的是找到错误并修正它，要尽可能地找到各种各样的用例，覆盖种种情况，努力找错。

-科学地、理性地进行测试用例设计（黑盒白盒都是），尽量根据测试设计原则进行设计而不是凭感觉，这有助于我们覆盖各种情况

练习题7. 图8-22说明了一个软件系统中的构件层次。分别描述用自底向上方法、自顶向下方法、改进的自顶向下方法、一次性集成方法、三明治方法和改进的三明治方法对构件进行集成测试的顺序。



\*每行代表一步，代表一批可以同时进行的测试，每一个框代表一个测试

自底向上：

[ L ] [ G ] [ H ] [ M ] [ N ] [ J ] [ K ] [ E ]

[ F L ] [ M N I ] [ C H ]

[ B F G ] [ D I J K ]

[ A B C D E ]

自顶向下：

[ A ]

[ A B C D E ]

[ A ~ K ]

[ A ~ N ]

改进的自顶向下：

[ A ]

[ B ] [ C ] [ D ] [ E ]

[ A B C D E ]

[ F ] [ G ] [ H ] [ I ] [ J ] [ K ]

[ A ~ K ]

[ L ] [ M ] [ N ]

[ A ~ N ]

一次性集成方法

[ A ] ~ [ N ]

[ A ~ N ]

三明治方法

\*目标层为BCDE

[ L ] [ G ] [ H ] [ M ] [ N ] [ J ] [ K ] [ E ] [ A ]

[ F L ] [ I M N ] [ C H ]

[ B F G ] [ D I J K ]

[ A B C D E ]

改进的三明治方法

\*目标层为BCDE

[ L ] [ G ] [ H ] [ M ] [ N ] [ J ] [ K ] [ E ] [ A ] [ I ] [ F ] [ C ]

[ F L ] [ I M N ] [ C H ] [ B ] [ D ]

[ B F G ] [ D I J K ]

[ A B C D E ]

第九章

练习题5. 在第4章中，我们讨论了需求的可测试性的必要性。解释为什么可测试性对性能测试是重要的。举例说明你的解释。

软件的可测试性是指在一定的时间和成本前提下，进行测试设计、测试执行以此来发现软件的问题，以及发现故障并隔离、定位其故障的能力特性。简单的说，软件的可测试性就是一个计算机程序能够被测试的容易程度。测试人员可以更容易定位问题，可以更快地对其进行调试，并且应用程序能够更快地到达用户手中，并且没有隐藏的故障。

练习题6. 字处理系统、工资单系统、自动银行连线柜员系统、水质监控系统、核电站控制系统分别需要哪种类型的性能测试?

1-功能测试、性能测试

2-功能测试、性能测试、验收测试

3-功能测试、性能测试

4-可靠性、可用性及可维护性测试、验收测试

5-验收测试、自动化系统测试

练习题8. 一个制导系统将安装在一架飞机上。当设计安装测试时必须考虑什么问题?

要考虑兼容性问题，以内飞机知道心痛能否正确工作取决于它能否与飞机系统之间很好的传递信息，因此要做兼容性测试。

练习题12. 一个工资单系统的设计要求在该公司工作的每名员工都有一个雇员信息记录。每个星期对雇员记录更新一次，记录该雇员本星期工作的小时数。每两星期打印一次汇总报表，显示从会计年度开始的总工作小时数。每月每名雇员的该月收入被转到他的银行账户中。针对照本章所介绍的每一种类型的性能测试，介绍它们是否适用于该系统。

压力测试，如何保证更新大量数据时系统正确准确的完成任务。

兼容性测试，能否与银行等系统兼容，进行转账等操作

可靠性测试：能否长期记录员工的工作时长并保存

验收测试：工作领导要首先查看并确定系统是否满足要求。