2017年UML考点整理

**自己手打，仅供参考**

1. 选择题（30’）
2. 判断or填空（20’）

选择15个小题、判断(选择)填空 10个小题，主要考查以下知识点：

1. 用例实现 Actor的作用

* 发现驱动用例的用户目标；为了明确外部接口和协议；为了确保确定并满足所有必要的重要实物

用例实现描述某个用例基于协作对象如何在设计模型中实现。

例题：

1. *Which of the following are not part of a use-case realization ?*

*A. Tagged Value B. Constraint C. Concurrent D. Stereotype*

1. *Which of the following is not part of a use-case relazation ?*

*A. Sequence diagrams B. Use case C. Communication D. Class diagram*

6、Controller

控制器是UI层之上的第一个对象，它负责接收和处理系统操作信息。

对于同意用例场景上的所有系统事件使用相同的控制器类。

控制器应当把需要完成的工作委派给其他的对象。控制器只是协调或控制这些活动，本身并不完成大量工作。

当把职责分配给外观控制器会导致低内聚或高耦合的设计时，通常是当外观控制器的职责过多而变的“臃肿”时，就需要考虑使用用例控制器。

GRASP控制器是领域层的一部分，它控制或协调工作请求的处理。

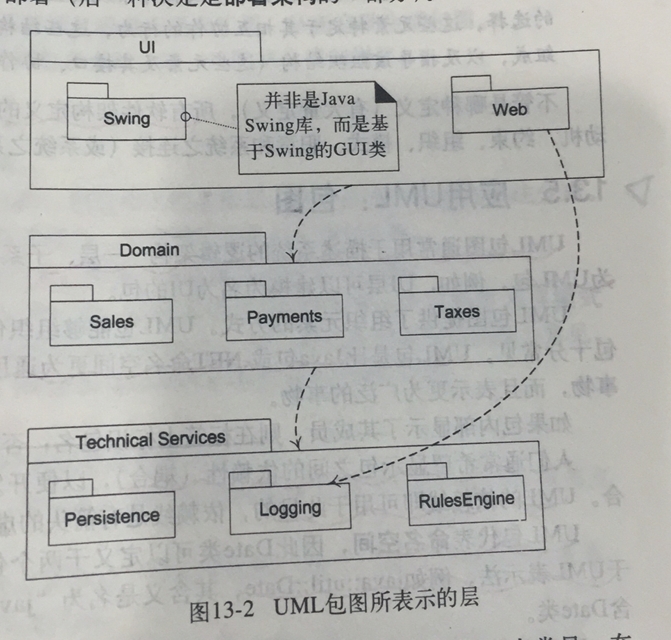
控制器优点：增加了可复用和接口可拔插的潜力。获得了推测用例状态的机会。

7、活动图、交互图

8、三层架构图 Layer 特征

较高层可以调用较低层的服务。在严格的分层架构（网络协议栈）中，层只能调用与其相邻的下层的服务。在宽松的分层架构（信息系统）中，较高层可以调用其下任何层的服务。

OO系统通常包含的层有：用户界面，应用逻辑和领域对象，技术服务。



1. 信用卡服务是什么

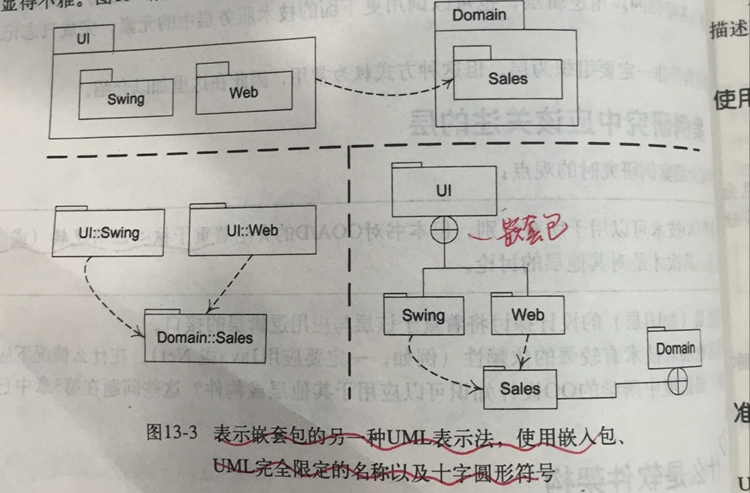
存在冗余→包含关系

用例在其他用例中重复使用

用例非常复杂并且冗长，将其分解为子单元便于理解。

1. 系统架构图→包图

UML包图显示了包之间的依赖性，UML的依赖线（有箭头的虚线指向被依赖的包）用来表示系统内大型事物之间的耦合。

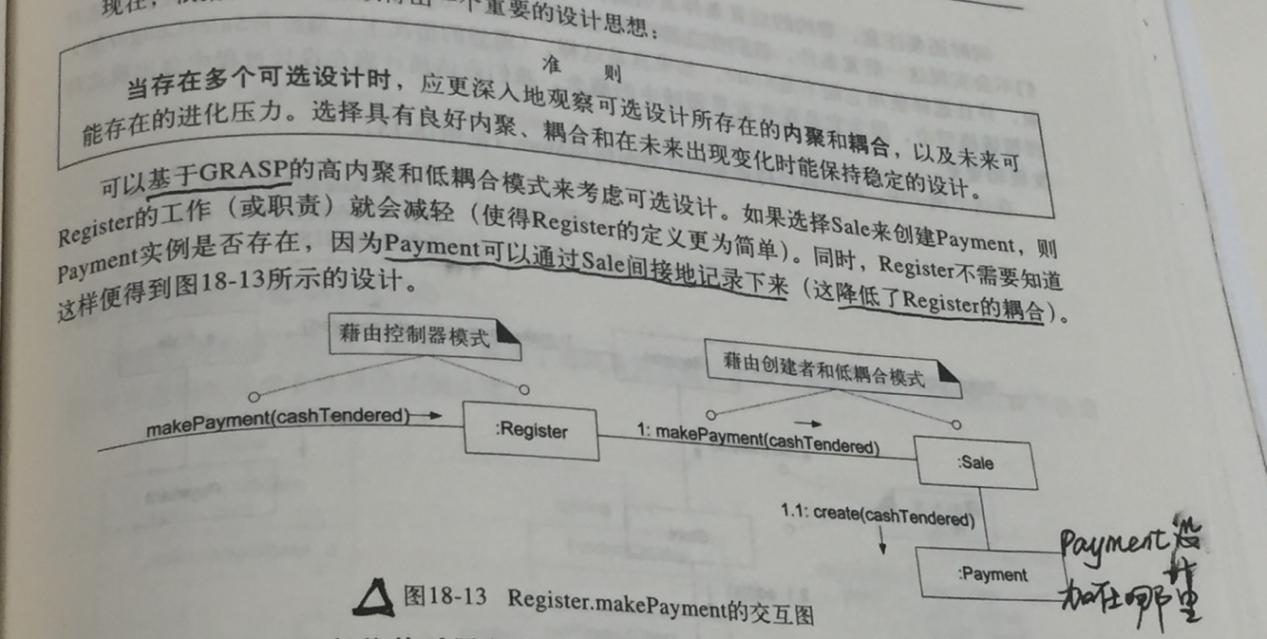


1. 分析建模题

画一个具体应用的类图，类图的关系要描述正确，如依赖关系，类的属性和方法等。

【P245 Payment、sale那段话】用GROSP分析 高内聚低耦合的应用，降低耦合程度。

可以基于GRASP高内聚和低耦合模式来考虑可选设计。如果选择Sale来创建Payment，则Register的工作就会减轻。同时Register不需要知道Payment实例是否存在，因为Payment可以通过Sale间接地记录下来，这降低了Register的耦合。



1. 综合设计题（30’）

4个小题（用例图、活动图、时序图、状态机图(不必描述所有可能的事件)）

以下三种中考一个：

在线租车系统（book、还、计费）

智能牙刷或温度计（对应app，把信息传到app上）

图书管理系统

**P.S. 建模题没有标准答案，所以一定要画图！画图就有分拿！**

补充：每种图的细节