

第8章 面向对象设计（习题与参考答案）

[选择题]

1. () 面向对象设计是在分析模型的基础上，运用面向对象技术生成软件实现环境下的设计模型。(A)
 - A. 真
 - B. 假
2. () 系统设计的主要任务是细化分析模型，最终形成系统的设计模型。(B)
 - A. 真
 - B. 假
3. () 关系数据库可以完全支持面向对象的概念，面向对象设计中的类可以直接对应到关系数据库中的表。(B)
 - A. 真
 - B. 假
4. () 用户界面设计对于一个系统的成功是至关重要的，一个设计得很差的用户界面可能导致用户拒绝使用该系统。(A)
 - A. 真
 - B. 假
5. 内聚表示一个模块()的程度，耦合表示一个模块()的程度。(B, D)
 - A. 可以被更加细化
 - B. 仅关注在一件事情上
 - C. 能够适时地完成其功能
 - D. 联接其他模块和外部世界
6. 良好设计的特征是()。(E)
 - A. 模块之间呈现高耦合
 - B. 实现分析模型中的所有需求
 - C. 包括所有组件的测试用例
 - D. 提供软件的完整描述
 - E. 选项 B 和 D
 - F. 选项 B、C 和 D
7. () 是选择合适的解决方案策略，并将系统划分成若干子系统，从而建立整个系

统的体系结构；（ ）细化原有的分析对象，确定一些新的对象、对每一个子系统接口和类进行准确详细的说明。（A, B）

- A. 系统设计
- B. 对象设计
- C. 数据库设计
- D. 用户界面设计

8. 下面的（ ）界面设计原则不允许用户保持对计算机交互的控制。（D）

- A. 允许交互中断
- B. 允许交互操作取消
- C. 对临时用户隐藏技术内部信息
- D. 只提供一种规定的方法完成任务

[练习题]

1. 面向对象设计与面向对象分析的区别是什么？设计包括哪些活动？

答案要点：

面向对象分析是重点考虑系统“做什么”的问题，即运用面向对象方法对问题域进行分析和理解，建立系统的分析模型；面向对象设计重点考虑系统“怎样做”的问题，即在分析模型的基础上形成实现环境下的设计模型。

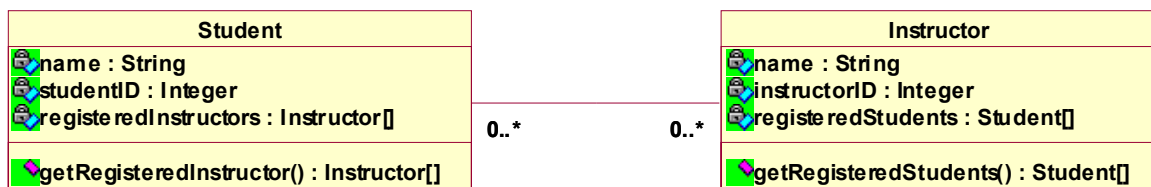
面向对象设计主要涉及系统设计、对象设计（或详细设计）、数据库设计和用户界面设计等活动。

2. 在类设计中，为什么建议使用 set 方法和 get 方法进行属性设置和读取？

答案要点：

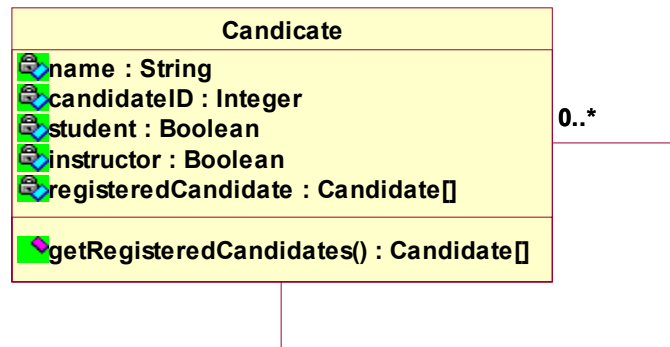
目的在于降低类之间的耦合度。

3. 下图显示了某个学校课程管理系统的部分类图，其中一个学生（student）可以知道所有注册课程的教师（instructor），一个教师也可以知道所有注册课程的学生。

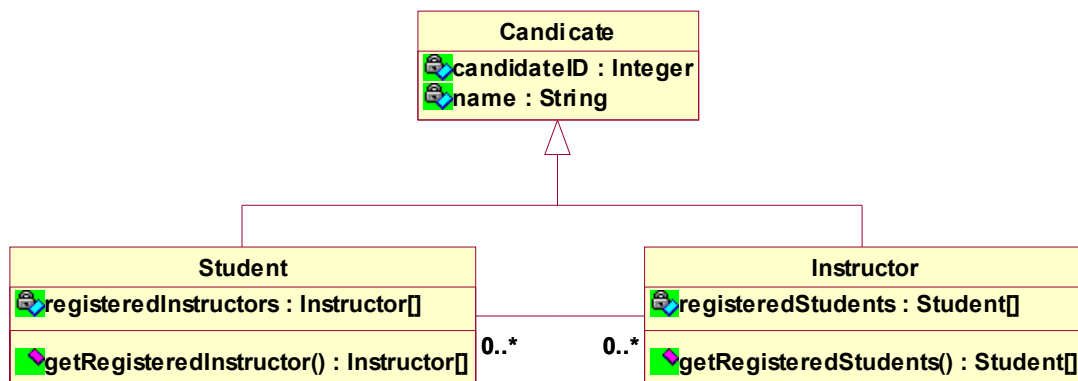


现在提出一个新的需求：“一个教师也可以是某些课程的学生”，那么下面设计 A~E 中哪一个是最好的？为什么？（请务必说明理由）

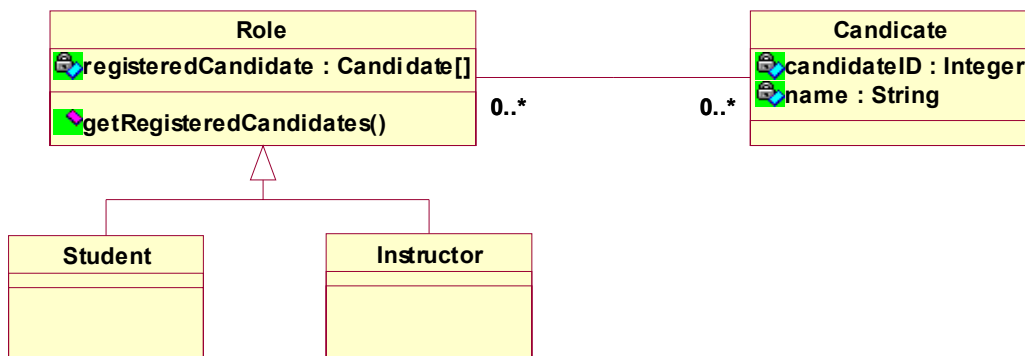
设计 A:



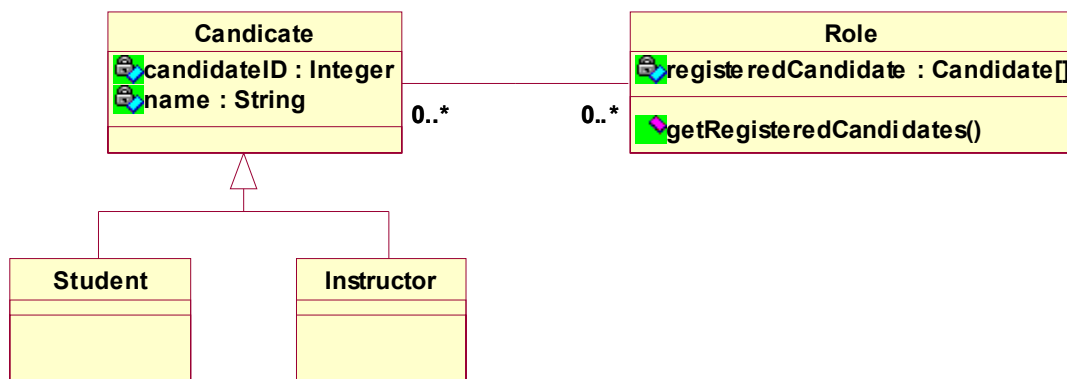
设计 B:



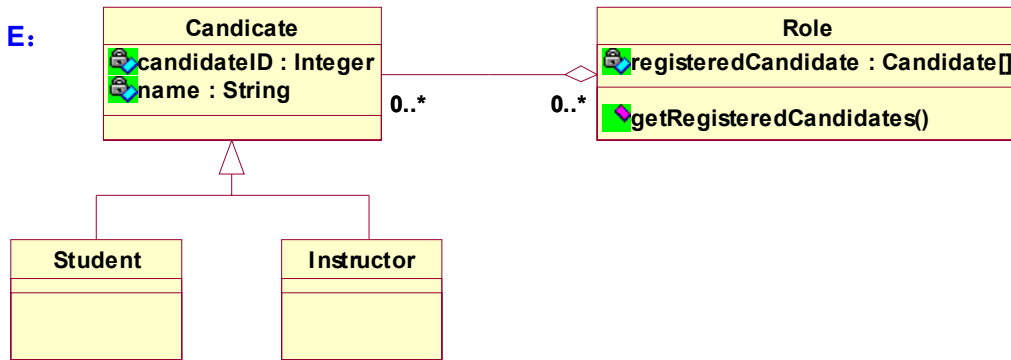
设计 C:



设计 D:



设计 E:



参考答案:

设计 A: 这个设计使用了一个自返关联，但其中的 `registeredCandidate()` 方法有问题。如果一个人既是学生又是老师，那么要想同时获得他作为学生对应的所有老师，以及他作为老师对应的所有学生，`getRegisteredCandidates()` 方法得到的 `Candidate` 数组中并不能区分上述的两种结果。

设计 B: 这个设计使用了一个泛化关系，但它并没有解决题中给出的新的要求。事实上，这个设计只是在题中已有设计之上的改进，实质还是一样的，并不能描述一个教师也可以是某些课程的学生情况。

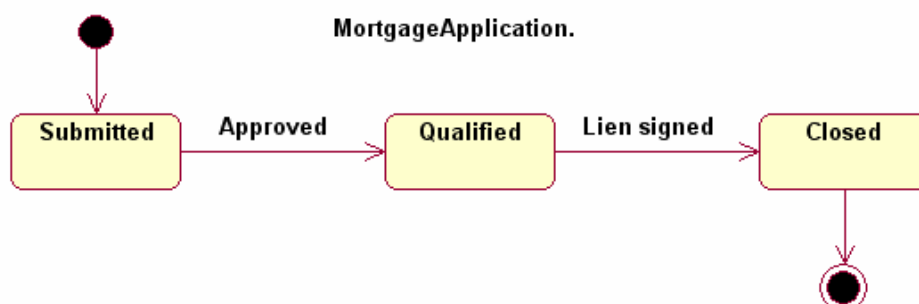
设计 C: 这个设计定义了一个 `Role` 类，学生和教师是 `Role` 类的两个子类。如果一个 `Candidate` 对象是教师，可以调用 `Instructor` 类的 `getRegisteredCandidates()` 方法获得所对应的所有学生；如果这个对象是学生，可以调用 `Student` 类的 `getRegisteredCandidates()` 方法获得所对应的所有教师；如果这个对象既是教师又是学生，那么分别调用 `Instructor` 类的 `getRegisteredCandidates()` 方法和 `Student` 类的 `getRegisteredCandidates()` 方法即可。

设计 D: 整个设计有着与 B 中类似的问题，即一个 `Candidate` 对象要么是一个学生，要么是一个老师，不能同时都是，无法满足新的需求。

设计 E: 这个设计与 D 基本一致，只是把 `Candidate` 类与 `Role` 类之间的关联关系改成了聚集关联，同样无法满足新的需求。

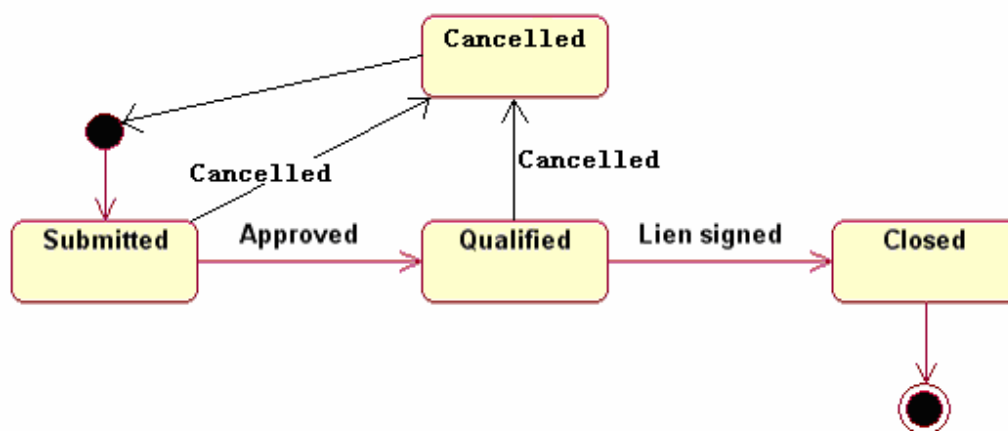
综上所述，设计 C 是最好的。

4. 下图是 `MortgageApplication` 对象的状态图，假设有一个要求，即增加一个“Cancelled”状态，图中除“Closed”状态外的任何状态都可以转换到该状态。请画出修改后的状态图（请考虑不同的修改方法中哪种方法最好）

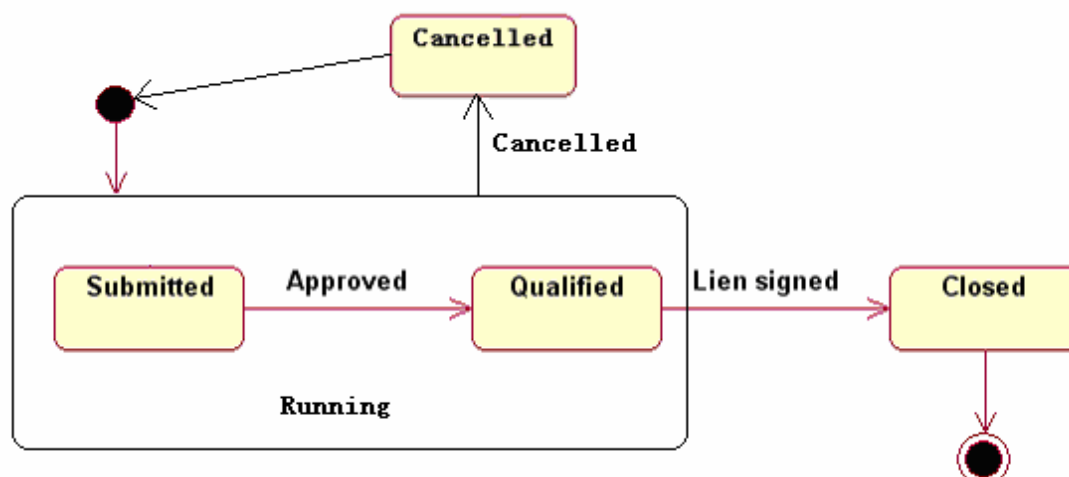


参考答案：

方法 1：



方法 2：



方法 2 引入了组合状态和子状态，从而可以在组合状态 Running 中加入更多的子状态，以保证组合状态的可扩展性。所以说，方法 2 相对于方法 1 更好一些。

5. 请举例说明用户界面设计的基本原则。

参考答案：

- (1) 用户控制：用户应当感觉系统的运行在自己的控制之下。在图形界面或基于 Web 的界面中，用户指导程序的每一步执行；即使在程序进行某些处理或用户等待输出结果时，用户同样保持对控制的敏感度。
举例：当程序进行某些需要占用较长时间的处理时，需要为用户提供及时的反馈信息，诸如一个沙漏、一个等待的指示器或其他类似的东西。
- (2) 界面一致性：一致性要求用户界面遵循标准和常规的方式，让用户处在一个熟悉的和可预见的环境之中，这主要体现在命名、编码、缩写、布局以及菜单、按钮和键盘功能在内的控制使用等。

举例：一个运行在 Windows 平台上的三维几何造型系统应当采用 Windows 图形窗口的“外观和感觉”，与 Office 类型软件保持一致的界面风格和操作方式。

- (3) 界面容错性：一个好的界面应该以一种宽容的态度允许用户进行实验和出错，使用户在出现错误时能够方便地从错误中恢复。

举例：Microsoft Word 系统允许撤消用户对文档的许多最近操作，从而处理用户的误操作问题。

- (4) 界面美观性：界面美观性是视觉上的吸引力，主要体现在具有平衡和对称性、合适的色彩、各元素具有合理的对齐方式和间隔、相关元素适当分组、使用户可以方便地找到要操作的元素等。

举例：清华大学软件学院综合信息服务系统的 Web 界面在平衡和对称、色彩、元素分组、用户方便地查找信息等方面进行了良好的设计。



- (5) 界面可适应性：界面可适应性是指用户界面应该根据用户的个性要求及其对界面的熟知程度而改变，即满足定制化和个性化的要求。所谓定制化是在程序中声明用户的熟知程度，用户界面可以根据熟知程度改变外观和行为；所谓个性化是让用户按照自己的习惯和爱好设置用户界面元素。

举例：系统允许用户选择语言类型（中文、英文、意大利文等），并系统根据用户定制的语言类型显示不同语种的用户界面。