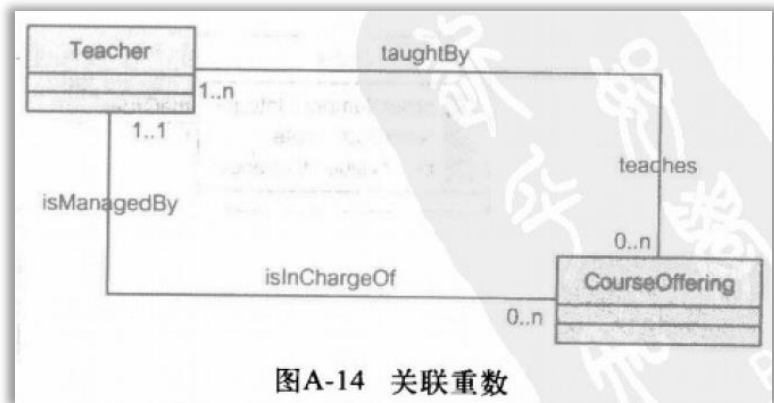
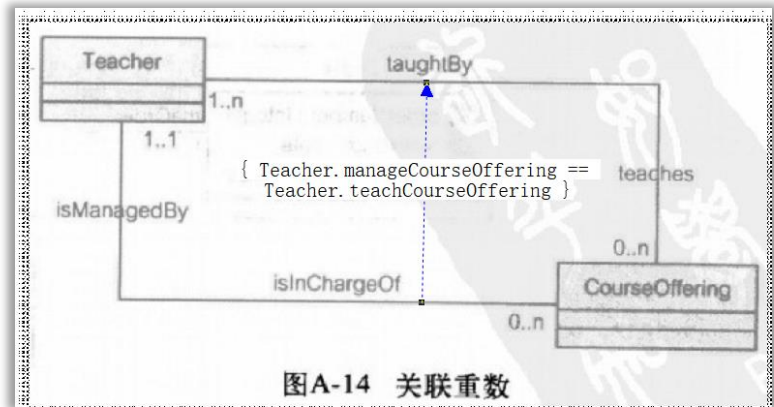


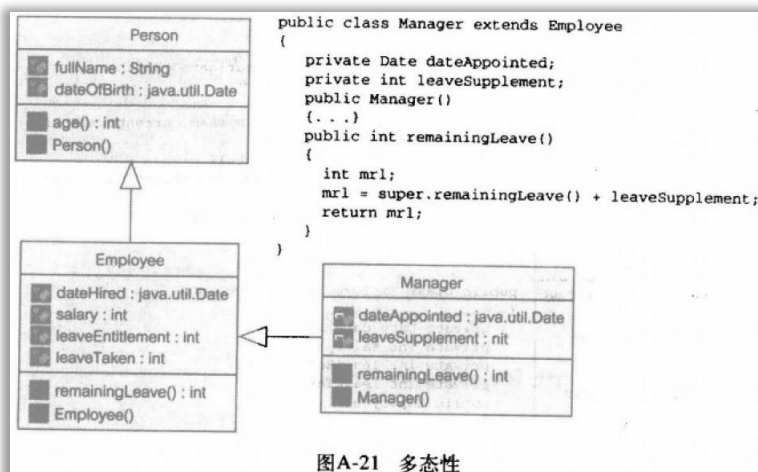
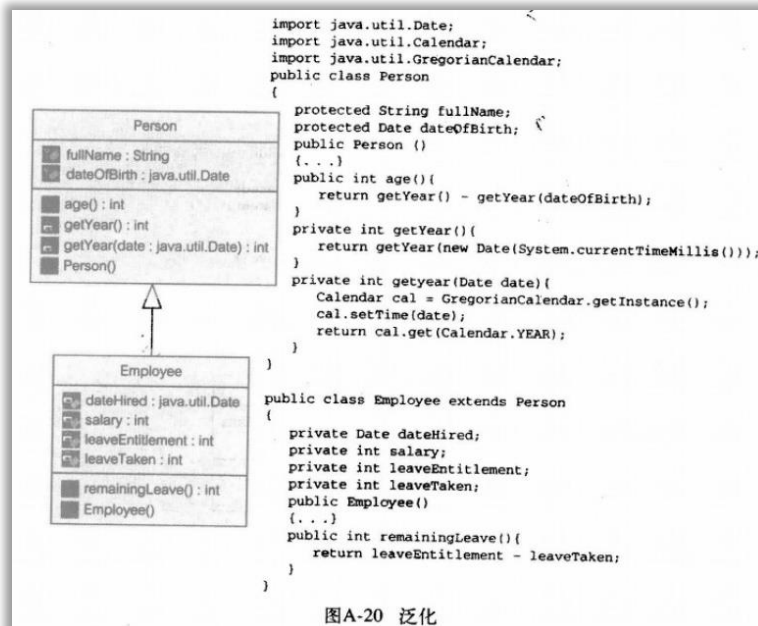
E1 参考图A-14（附录，A.5.2节），假设管理课程设置的教师也必须教这门课。修改图A-14来捕捉这一事实。



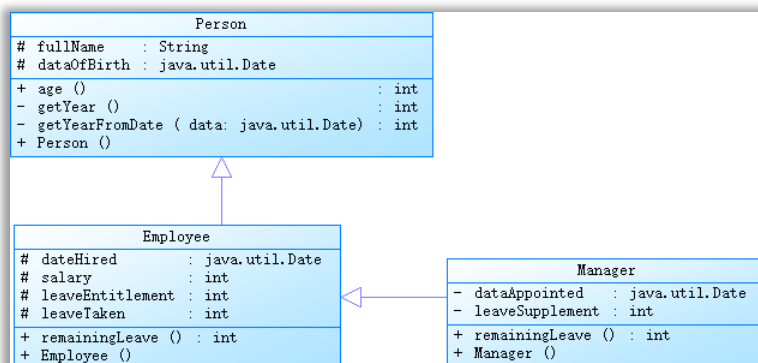
可以在两个类的关联线上添加一个约束，表示管理课程设置的教室也必须教这门课。约束绑定在箭头上，如下图所示。



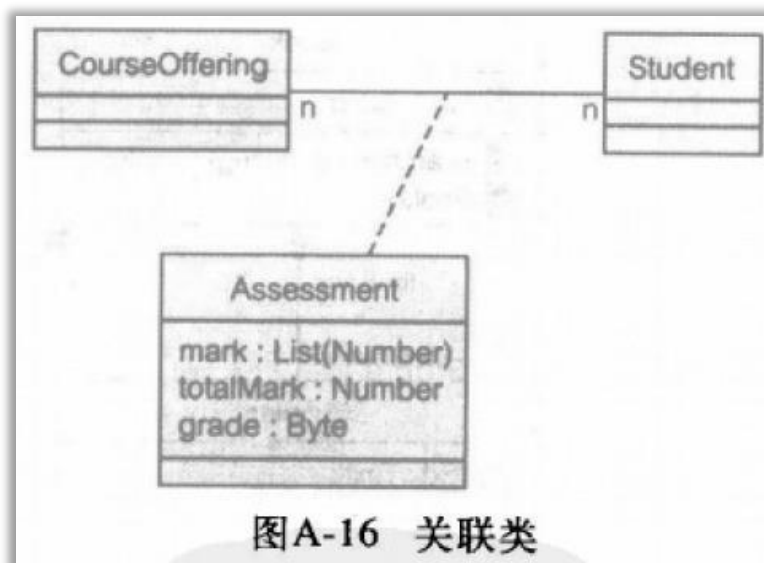
E2 参考图A-20（附录，A.7节）和图A-21（附录，A.7.1节），将这两个图合并成一个类模型。设计类模型中的可见性。解释你的答案。



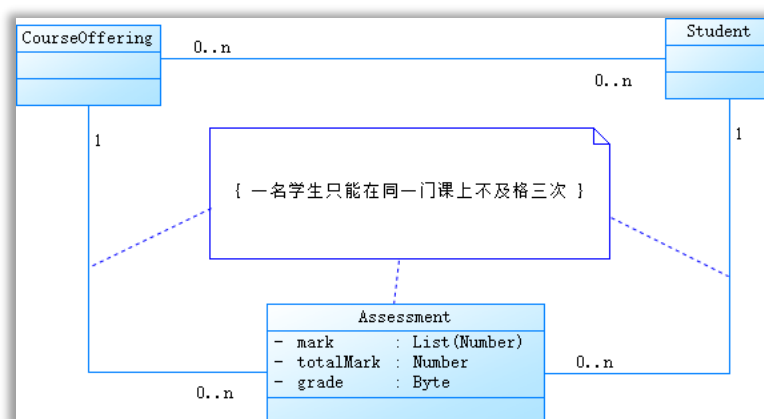
由原模型，Person 是 Employee 类的父类，Manager 是 Employee 类的子类，三个类之间是泛化关系。合并两图结果如下，其中 Person 类和 Employee 类的数据均采用 protected，而 Manager 类的数据采用 private。



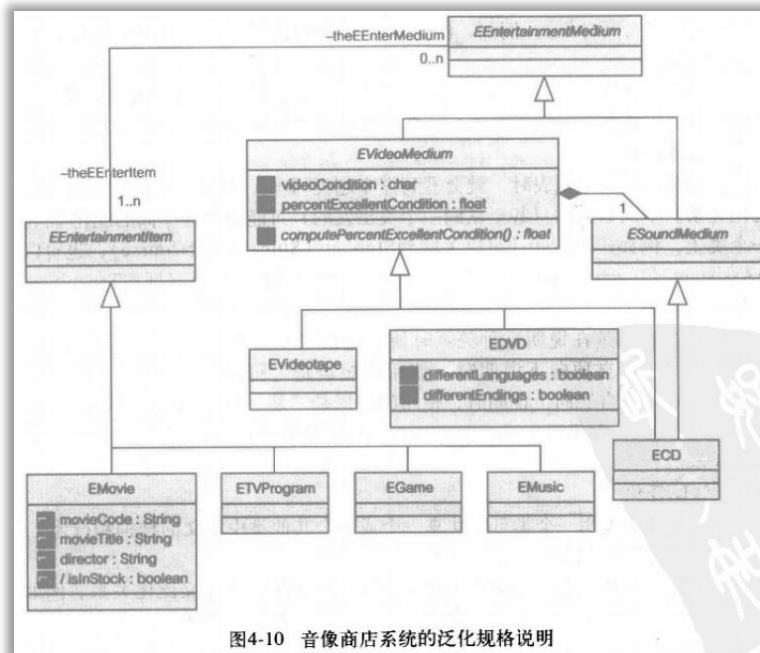
- E3 参考图A-16（附录，A.5.4节），假设系统不得不在同一门课的多个课程设置中监控对学生的评价。这是因为有这样的限制，即一名学生只能在同一门课上不及格3次（不允许第4次注册）。扩展图A-16来对上述约束建模，使用一个具体化类，建模并/或解释所有假设。



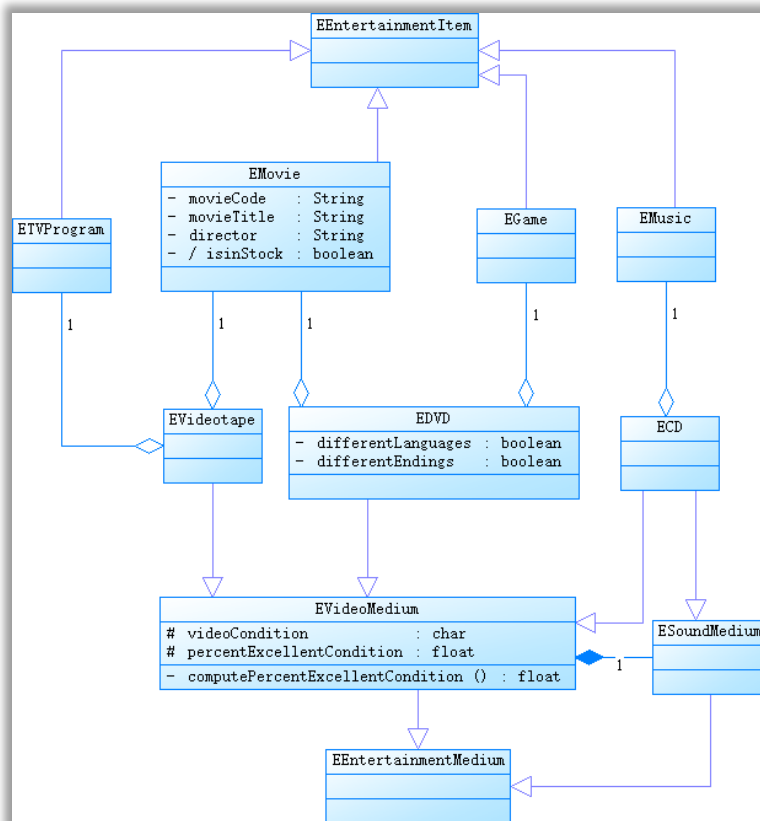
首先将关联类 **Assessment** 转化为具体类，将三者关联为如下图模型。最后添加约束注释：“一名学生只能在同一门课上不及格三次”，并链接到 **Assessment** 类和它与另外两个类的关联上。



E4 参考例4.10 (4.2.4.3节)。用聚合代替泛化重画图4-10，解释新模型的利与弊。



新的模型利在：更清楚直观表现出类模型部分与整体的关系，降低类耦合性，有利于面向对象分析；弊在：降低了系统模型的表现力、可理解性和抽象程度，增加了模型中聚合关系的总数与理解难度。



E5 参考例4.18 (4.3.3.3节)，利用组合片段和其他高级交互建模概念来改进图4-18中的顺序图。

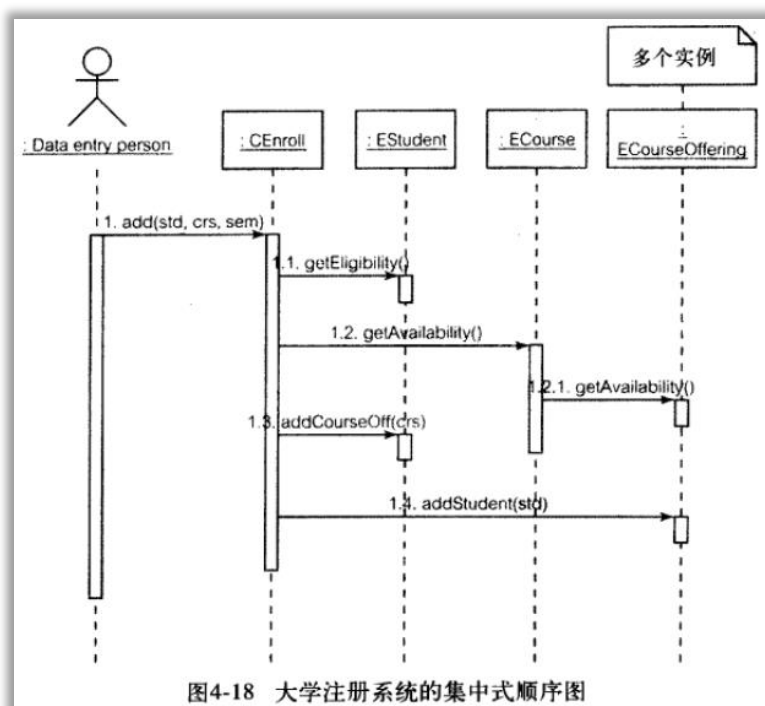


图4-18 大学注册系统的集中式顺序图

如下图所示，选择了可选片段、循环片段和选择片段三个组合片段来改进上图。当方法 `getEligibility()` 返回值为真时，使用 `getAvailability()` 从 `ECourseOffering` 类中递归获取目标 `ECourse` 是否可用。若可用，则添加课程，并将学生添加到课程中；若不可用，则退出；当方法 `getEligibility()` 返回值为假时，退出。

SequenceDiagram\_1

