

# 软件工程作业 2

2024 年 11 月 21 日

## 1 概念题

1. 请解释以下术语，并举例说明。术语有对象，操作，类，关联，链，泛化，聚合，依赖。
2. 描述类之间的关系所使用的概念都有哪些？请列出并解释它们。
3. 在什么情况下需要建立状态图？
4. 面向对象方法与结构化方法在控制信息组织复杂性方面引入的机制都有哪些？请列出并解释它们。
5. 面向对象为什么要从多个侧面建立系统模型？

解答：

1. 解释与举例如下：

- (a) 对象：对象是系统中用来描述客观事物的一个实体，它是构成系统的一个基本单位。一个对象由一组属性和对这组属性进行操作的一组服务构成。e.g. 我们可以在计算机世界中建立一个汽车对象来描述现实世界中的汽车这个实体。
- (b) 操作：对象的操作是用来表示事物的行为的。e.g. 系统中的汽车对象有启动操作，是为了表示现实世界中汽车的启动这个行为的。
- (c) 类：对事物进行分类，把具有相同属性和相同操作的对象归为一类，类是这些对象的抽象描述，每个对象是它的类的一个实例。e.g. 洗衣机属于家用电器这个分类，那么我们可以从系统中抽象出家用电器类来表示现实世界中家用电器这一类实体，洗衣机就是家用电器类的一个示例。
- (d) 关联：关联是表达类（一组对象）之间的静态关系的（即通过对象属性体现的联系）。e.g. 例如教师和学生之间的任课关系就是一种关联，在实现中可以通过对象的属性表达出来。
- (e) 链：链是关联的一个实例。或者，链是对象（或者类目的实例）之间特殊的语义关系的抽象。e.g. 

Student
---------

 — 

Course
--------

。学生和选课之间有选课的关联关系，那么我们就说这两个类的对象的关系将会构成一条链。

Müller
--------

 — 

Software Engineering
----------------------

 就是这个关系的一个实例，构成了一条链，表示学生的一个对象实例 Müller 选了课程的一个对象实例 Software Engineering。
- (f) 泛化：泛化是一般性类目（称为超类或父类）和它的较为特殊性类目（称为子类）之间的一种关系，有时称为“is-a-kind-of”关系。e.g. 洗衣机是家用电器的一个子类，洗衣机和家用电器之间是泛化关系。

- (g) 聚合：一种特殊形式的关联，表达“整体/部分”关系。e.g. 教师和大学是聚合的关系，教师是部分，大学是整体。
- (h) 依赖：依赖是一种使用关系，用于描述一个类目使用另一类目的信息和服务。e.g. GitHub 系统中用户类可能会需要使用仓库类的一些信息和服务。

2. 描述类之间的关系的概念有：关联、泛化、实现和依赖。

- (a) 关联：关联是类目之间的一种结构关系，描述了两个或多个类的实例之间的连接关系。聚合是一种特殊的关联关系，是一种“整体/部分”的关系。组合又是聚合的一种特殊形式，如果整体类的实例负责创建和消除部分类的实例，而且整体类的实例和部分类的实例具有相同的生命周期。
- (b) 泛化：泛化是一般性类目（称为超类或父类）和它的较为特殊性类目（称为子类）之间的一种关系，有时称为“is-a-kind-of”关系。泛化是一种支持复用的机制。如果两个类具有泛化关系，那么：
  - i. 子类可以继承父类的属性和操作，并可有自己的属性和操作；
  - ii. 子类可以替换父类的声明；
  - iii. 若子类的一个操作的实现覆盖了父类同一个操作的实现，这种情况称为多态性，但两个操作必须具有相同的名字和参数；
  - iv. 泛化可以存在于其他类目之间，例如在节点之间、类和接口之间等。
- (c) 实现：实现是类目之间的语义关系，其中一个类目规定了保证另一个类目执行的契约。在实际应用中，一般在下面 2 个地方会使用实现关系：
  - i. 接口与实现它们的类和构件之间；
  - ii. 用况与实现它们的协作之间。
- (d) 依赖：依赖是一种使用关系，用于描述一个类目使用另一类目的信息和服务。例如，一个类使用另一个类的操作，显然在这种情况下，如果被使用的类发生改变，那么另一个类的操作也会受到一定的影响。有绑定（bind）、导出（derive）、允许（permit）、实例（instance of）、实例化（instantiate）、幂类型（powertype）、精化（refine）、使用（use）。

3. 建立状态图的目的是为了准确认识对象的行为。只有那些行为比较复杂的对象，才需要为其建立状态图。

4. 面向对象方法：

- (a) 封装：封装原则使对象能够完整对应并描述具体的事物，体现了事物的相对独立性，使对象外部不能随意存取对象的内部数据，也不能访问对象的内部操作；封装原则降低了对象之间的耦合度，对象的内部修改对外部的影响很小，减少了由于修改而引发的对象间的“波动效应”。
- (b) 抽象：抽象是指忽略事物的非本质特征，而提取出事物共同的本质特征（即共性）的思维方式。在面向对象分析和设计中，存在过程抽象和数据抽象两种抽象类型。其中类的操作是一种过程抽象，类作为数据类型及施加在其上的操作结合体，是一种新的数据类型，所以类是一种数据抽象。
- (c) 继承：继承可以提高软件复用和可维护性。

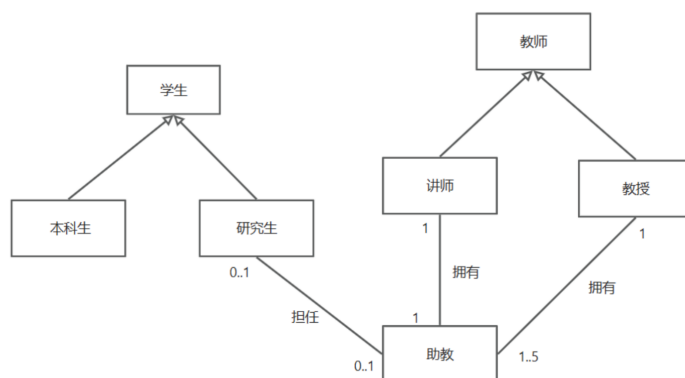
- (d) 多态性：多态性允许使用一个接口来表示多种操作。这种机制增强了系统的灵活性和可扩展性。
- (e) 包：包是模型元素的一个分组，一个包本身可以嵌套在其他包中，并且可以具有子包和其他种类的模型元素。这样对于复杂类图和其他视图，就可以按照模型元素的紧密程度分成不同的包。

结构化方法：

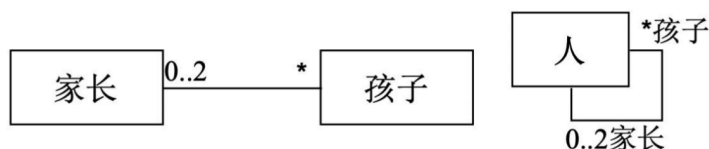
- (a) 功能/过程抽象：过程抽象是对具有明确和有限功能的功能序列的抽象。
- (b) 逐步求精：逐步求精是指把要解决问题的过程分解为多个步骤或阶段，每一步是对上一步结果的精化，以接近问题的解法。
- (c) 模块化：模块化是指按照“高内聚、低耦合”的设计原则，形成一个相互独立但又有较少联系的模块结构的过程，使每个模块具有相对独立的功能或过程。
5. 可以为复杂系统提供更完整的视图。对基本模型-类图进行辅助描述。我们可以了解系统的行为和状态变化，还可以了解系统的物理结构等。

## 2 实践题

1. 请从类之间关系的角度解释如下类图的含义。



2. 如下的两个类图描述的都是父母与子女之间的关系。请分析二者的优缺点。

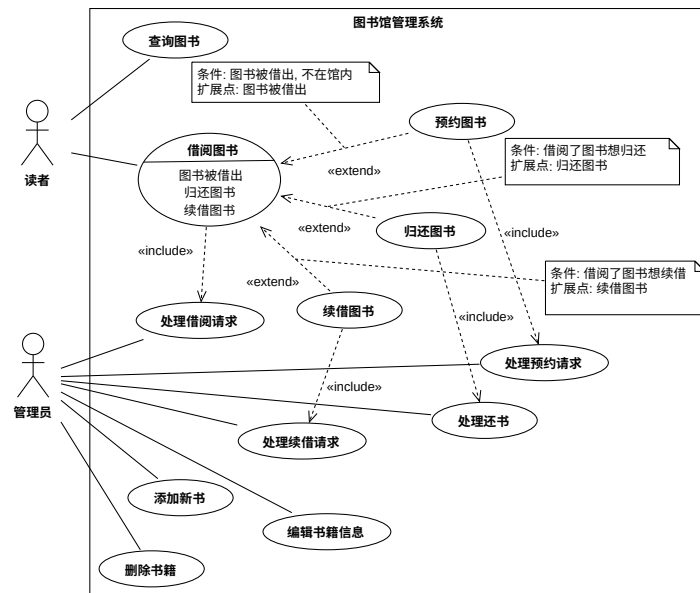


3. 假定某图书馆管理系统支持以图书查询借阅、图书管理等为中心的各项功能。在这一假定下，结合实际情况，请给出该系统的用况图，利用面向对象分析的方法建立该系统的类图，并选取类图中一个典型的类，给出它的状态图。
4. 用顺序图描述学生选课过程：

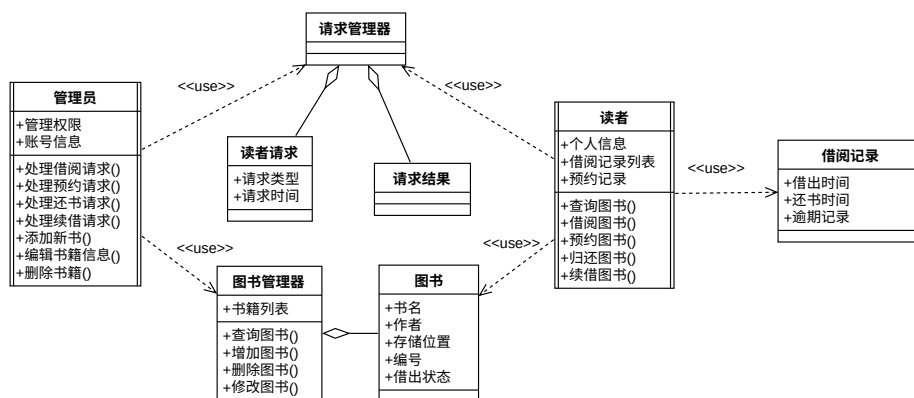
学生通过选课系统的前端页面查询课程信息并选择相应课程，前端在接收到学生的选课信息后提交给后台，后台判断该课程选课人数是否已满，若选课人数已满，则在前端页面提示学生重新选课，若人数未满，则给出选课成功信息。

解答：

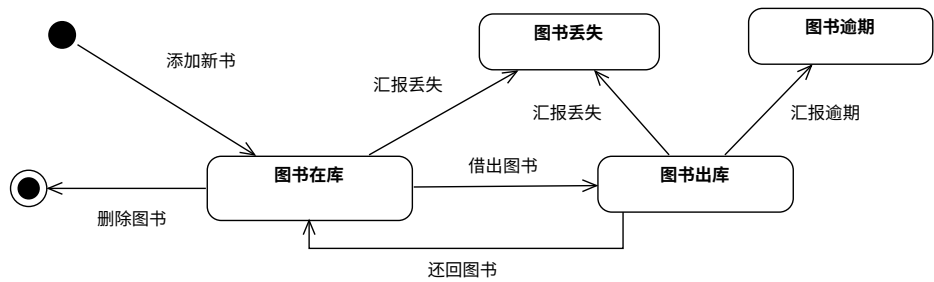
1. 本科生和研究生都是学生的子类。本科生和学生、研究生和学生都是一种泛化关系。1 个研究生既可以担任助教又可以担任助教，1 个助教既可以由 1 个研究生担任也可以不由研究生担任。1 个讲师可以拥有一个助教，1 个助教可以协助 1 个讲师。1 个教授可以拥有 1 到 5 个助教，1 个助教可以协助 1 个教授。讲师和教授都是教师的子类。
2. 左图直接表达的家长和孩子直接的关系，非常清晰；但是不具有通用性，当需要扩展其他角色或复杂的关系的时候会导致变得非常混乱。右图的通用性更好，可以应用于更广泛的场景而不至于局限于家庭结构，可以添加新的角色和关系；但是抽象度高，不够精确。
3. 用况图如下：



类图如下：



我们选择图书类，它的状态图如下：



4. 顺序图如下：

