3.1 需求分析的任务

3.2 与用户沟通获取需求的方法

3.3 分析建模与规格说明

3.4 实体联系图（不讲）

3.5 数据规范化（不讲）

3.6 状态转换图

3.7 其他图形工具

3.8 验证软件需求

2、面向对象的分析方法

最流行的是由 Rumbaugh、 Booch 、 Jacobson 提出的**统一建模语言UML**， Unify Modeling Language。

为面向对象的软件开发提供统一、标准、可视化的建模语言，能描述开发需要的各种视图，并以此为基础组建系统。

用例建模

用例建模用于描述系统需求，把系统当作黑盒，从用户的角度，描述系统的场景。主要的图形元素有：

参与者(Actor)：是指外部用户或外部实体在系统中扮演的角色。 可以是一个人、一个硬件设备等。

用例(Use Case)：对一组动作序列的描述，系统通过执行这一组动作序列为参与者产生一个可观察的结果。用例名往往用动宾结构命名。

表示关系的连接线。

建立用例模型的顺序

1. 确定谁会直接使用该系统。这些都是参与者(Actor)。
2. 选取其中一个参与者。
3. 定义该参与者希望系统做什么，参与者希望系统做的每件事成为一个用例。
4. 对每件事来说，何时参与者会使用系统，通常会发生什么，这就是用例的基本过程。
5. 描述该用例的基本过程。
6. 考虑一些可变情况，把他们创建为扩展用例。
7. 复审不同用例的描述，找出其中的相同点，抽出相同点作为共同的用例。
8. 重复步骤2~7找出每一个用例。

如何确定参与者?

- 识别参与者的思路
  - 谁**使用**系统的主要功能?
  - 谁**改变**系统的数据?
  - 谁从系统**获取**信息?
  - 谁**需要系统的支持**以完成其工作?
  - 谁**维护管理系统**，并保证系统正常运行?
  - 系统需要处理哪些**硬件设备**?
  - 与系统**交互的是什么系统**?
  - 谁对系统产生的**结果感兴趣**?
  - ……

确定参与者

例：饮料自动售货系统

在饮料自动售卖机中，最先想到的参与者是**顾客**。

供应商向自动贩卖机添加饮料

收银员从自动贩卖机收钱

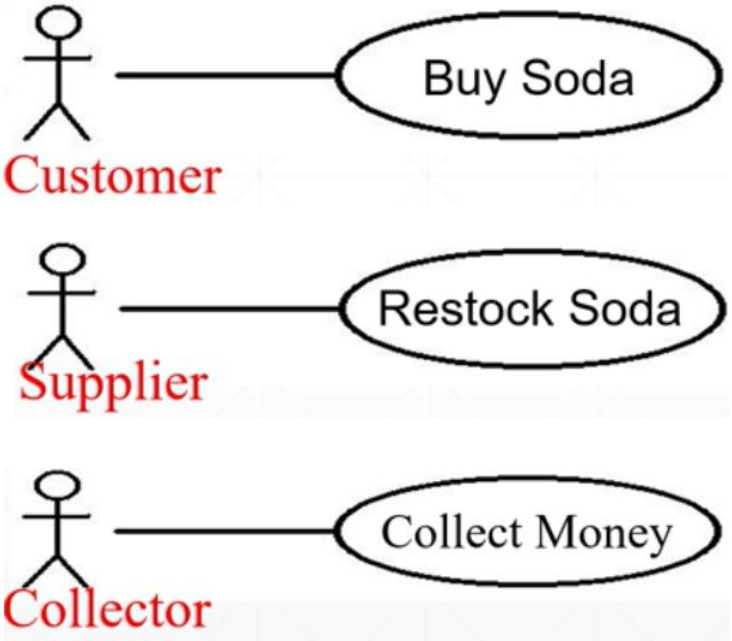
每一种参与者具有自己的 use case

如何确定用例

- 参与者希望系统执行什么任务?
- 参与者在系统中访问哪些信息(增、删、改、查等)?
- 需要将哪些外界信息提供给系统
- 需要将系统的什么事情告诉参与者
- 如何维护系统

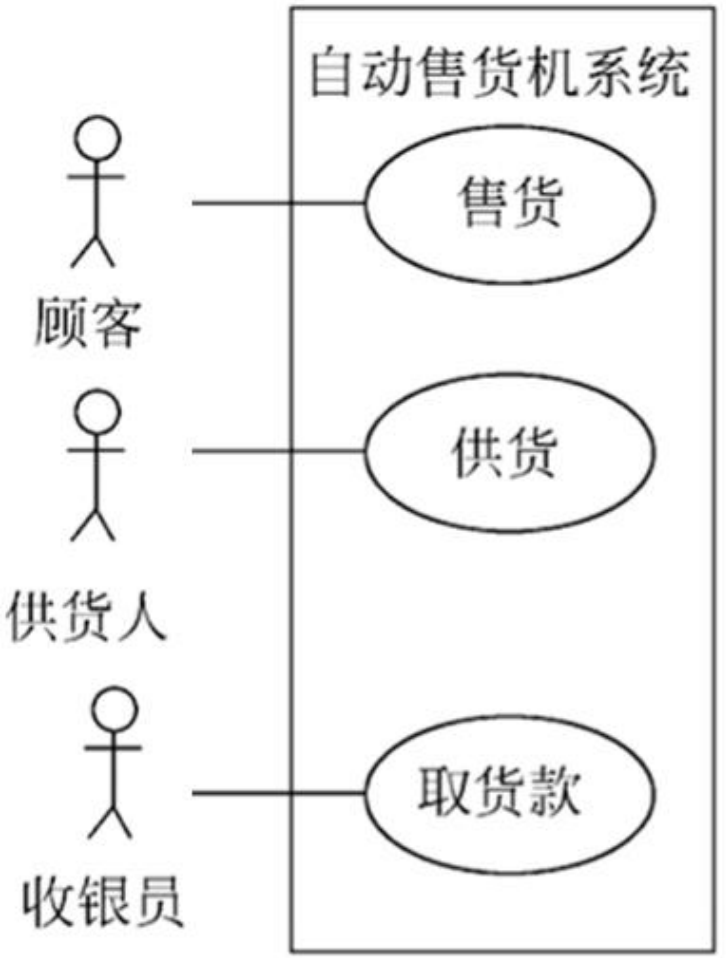
参与者希望系统做的

每件事成为一个用例



系统

系统：用于界定系统功能范围，描述该系统功能的用例都置于其中，而描述外部实体的参与者都置于其外。



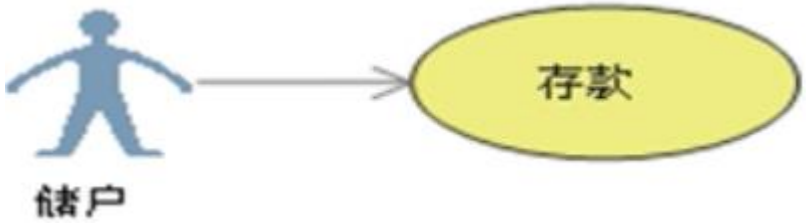
自动售货系统初步的用例图

四种关系

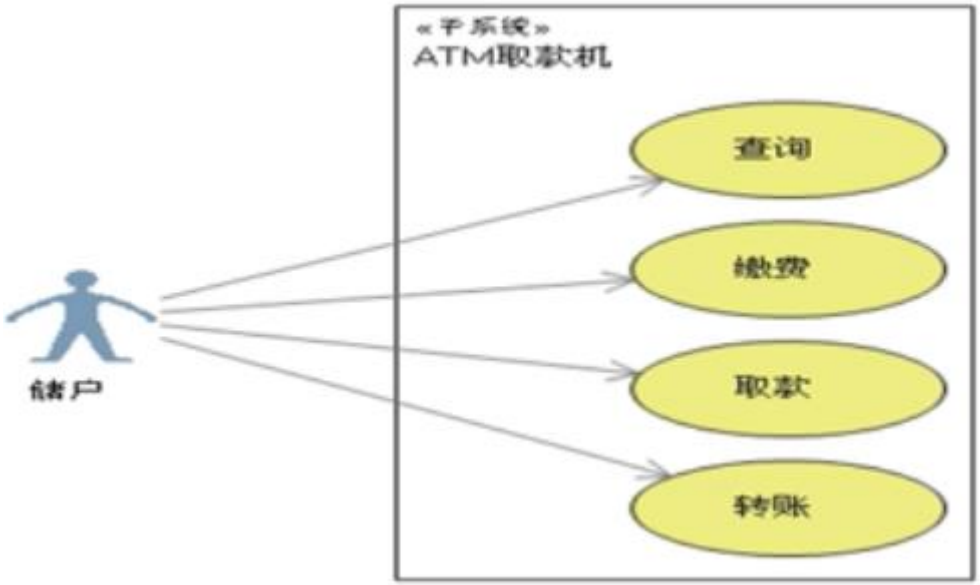
1. 关联(Association)

参与者与用例之间的通信，任何一方都可发送或接受消息。

箭头指向：指向消息接收方



参与者可参与多个用例，形成子系统

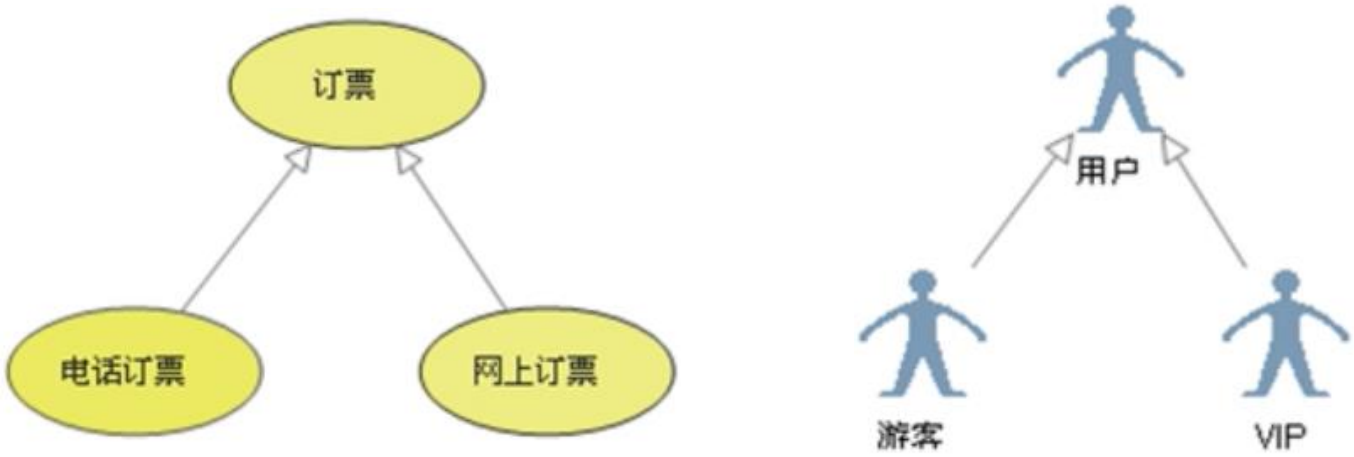


四种关系

2. 泛化(Inheritance)

就是通常理解的继承关系，子用例和父用例相似，但表现出更特别的行为；子用例可以重载父用例。父用例通常是抽象的。用例之间的is a kind of 关系，表示用例之间的场景共享；参参与者之间的 is a kind of 关系，一般描述职责共享。

箭头指向：子用例指向父用例，箭头为空箭头。

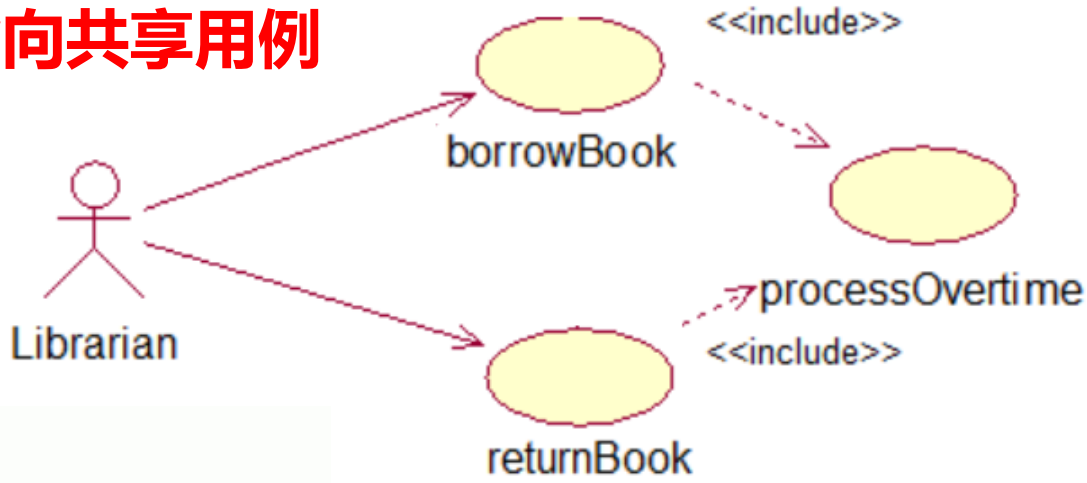


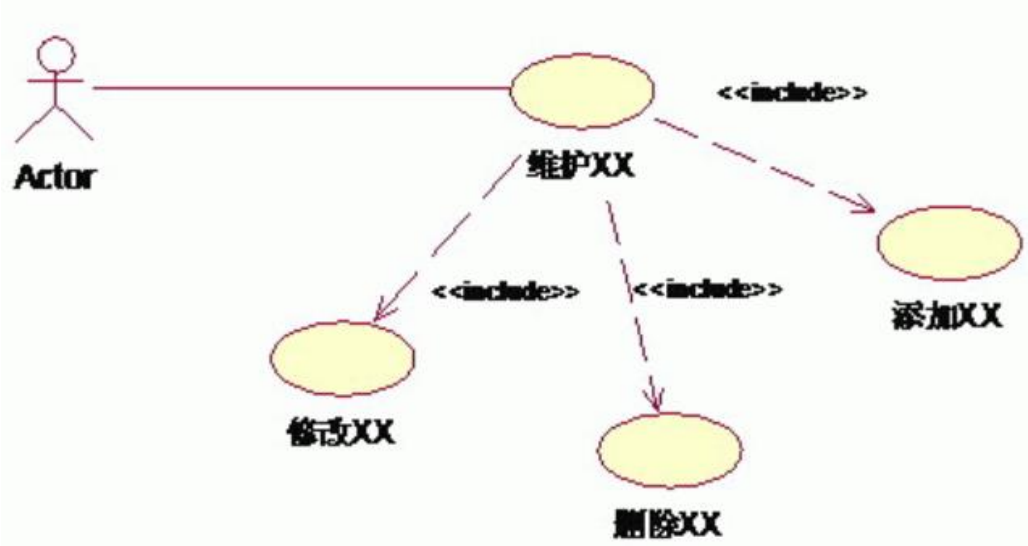
四种关系

3. 包含(Include)

当多个用例有共享行为时，应该使用包含的关系来表示它们。

箭头指向：基本用例指向共享用例





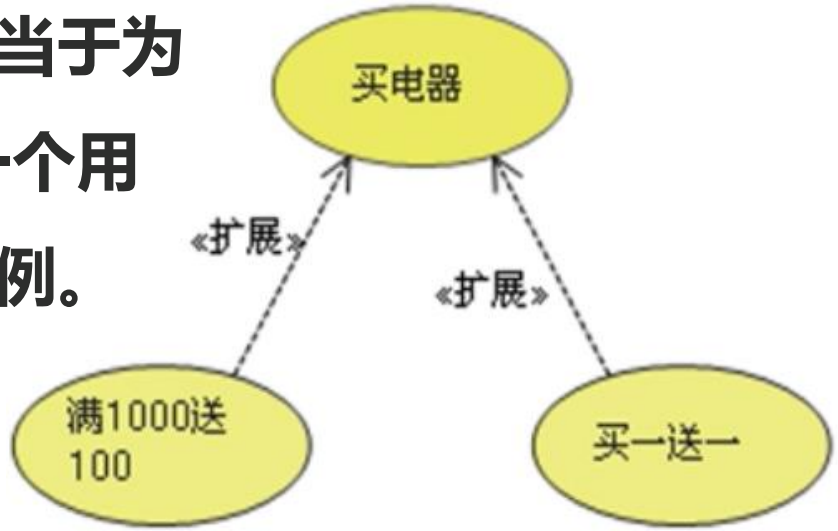
有时当某用例的事件流过于复杂时，为简化用例的描述，也可抽象出一个基用例，来包含这些颗粒的用例。

四种关系

4. 扩展(Extend)

扩展关系是指用例功能的**延伸**，相当于为基础用例提供一个**附加功能**。 由一个用例的**扩展**点可以扩展出另外一个用例。

箭头指向：**指向基础用例**

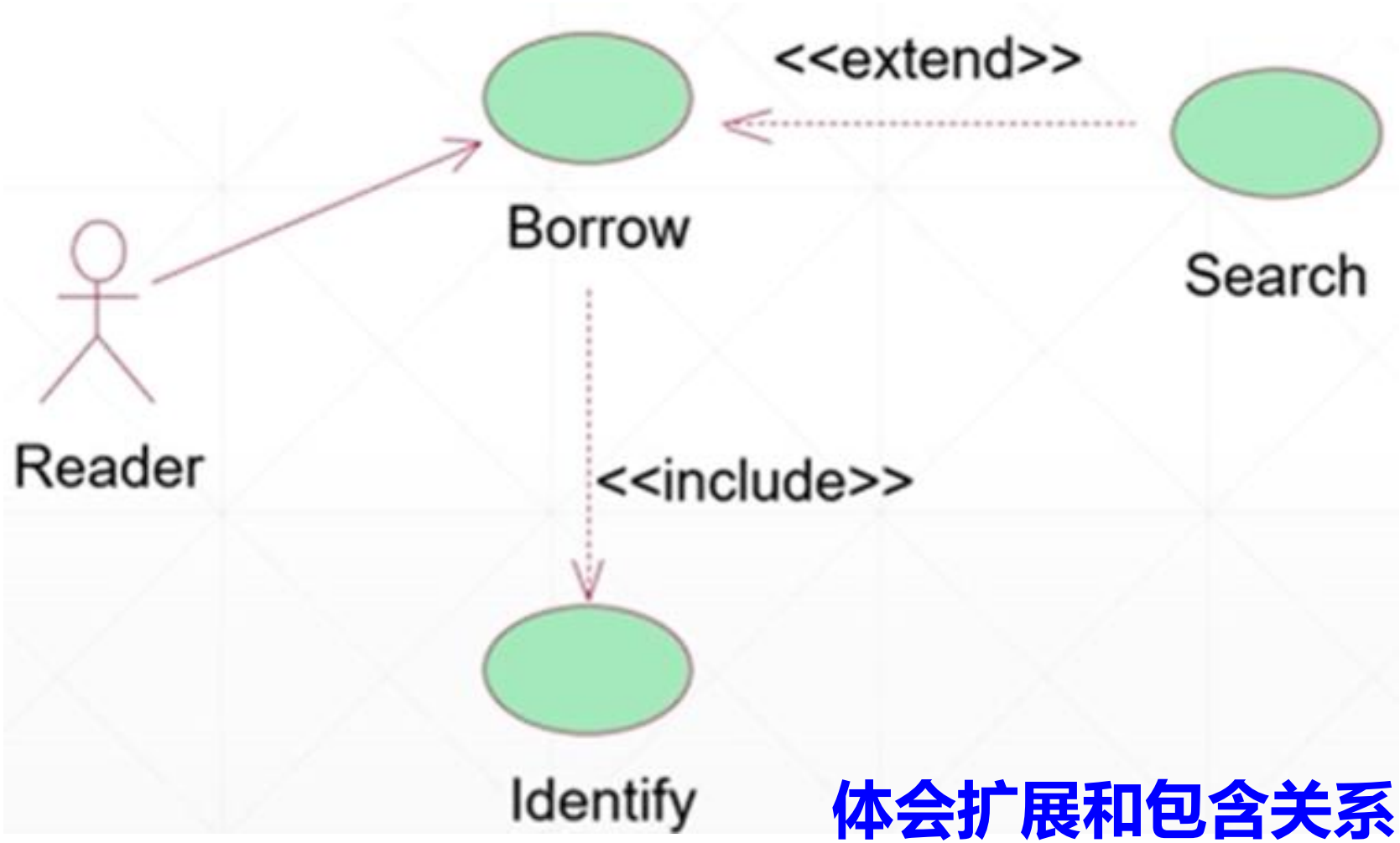


□ 包含、扩展的区别：

包含关系中，在执行基本用例时，**一定**会执行包含用例部分。

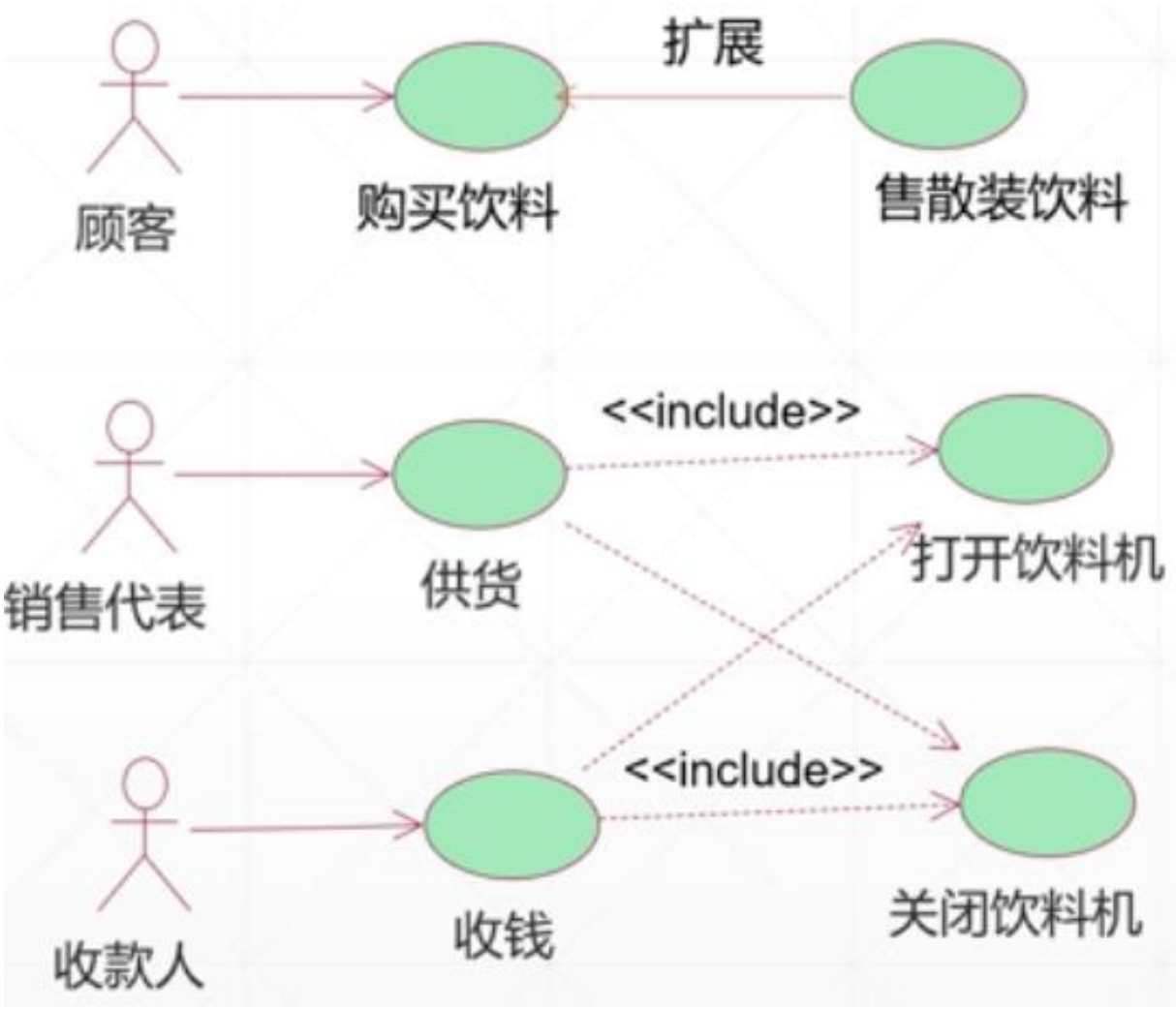
扩展关系中，一个基本用例执行时，**可以**执行、**也可以**不执行扩展用例部分。

图书借阅系统用例图



体会扩展和包含关系

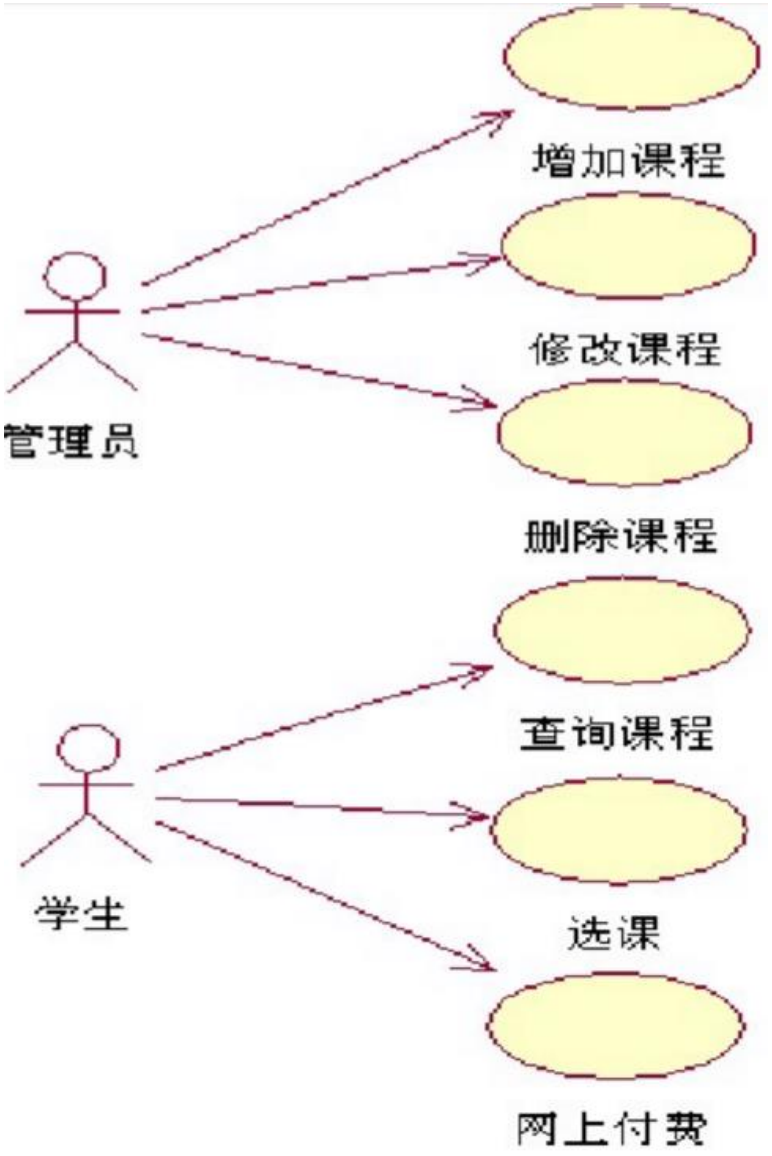
用例建模



自动售货系统用例扩展后的用例图

案例：网上选课系统用例

管理员通过系统管理界面进入系统，建立本学期要开设的各种课程，将课程信息保存到系统中并可以对课程进行改动和删除。学生通过客户机浏览器进入系统，可以查询课程，选择课程，支付课程费用。



编制用例说明

用例:增加课程

参与者:管理员

操作流:

①**管理员**选择进入管理界面，用例开始。

②**系统**提示输入管理员密码。

③**管理员**输入密码。

④**系统**检验密码。**A1:密码出错**

⑤进入管理界面，**系统**显示当前所建立的全部课程信息。

⑥**管理员**选择增加课程，**管理员**输入新课程信息。

⑦**系统**验证具否与已有课程冲突。**A2：有冲突**

⑧**系统**添加新课程，并提示添加成功。

⑨**系统**回到管理主界面，显示所有课程，用例结束。

主观题 10分

设置

有一家教管理系统，其要求的功能如下：

普通用户可以浏览家教信息、搜索家教信息；

家教老师和家教学生还可以注册本人信息、修改本人资料。

请根据需求，画出该系统的用例图。

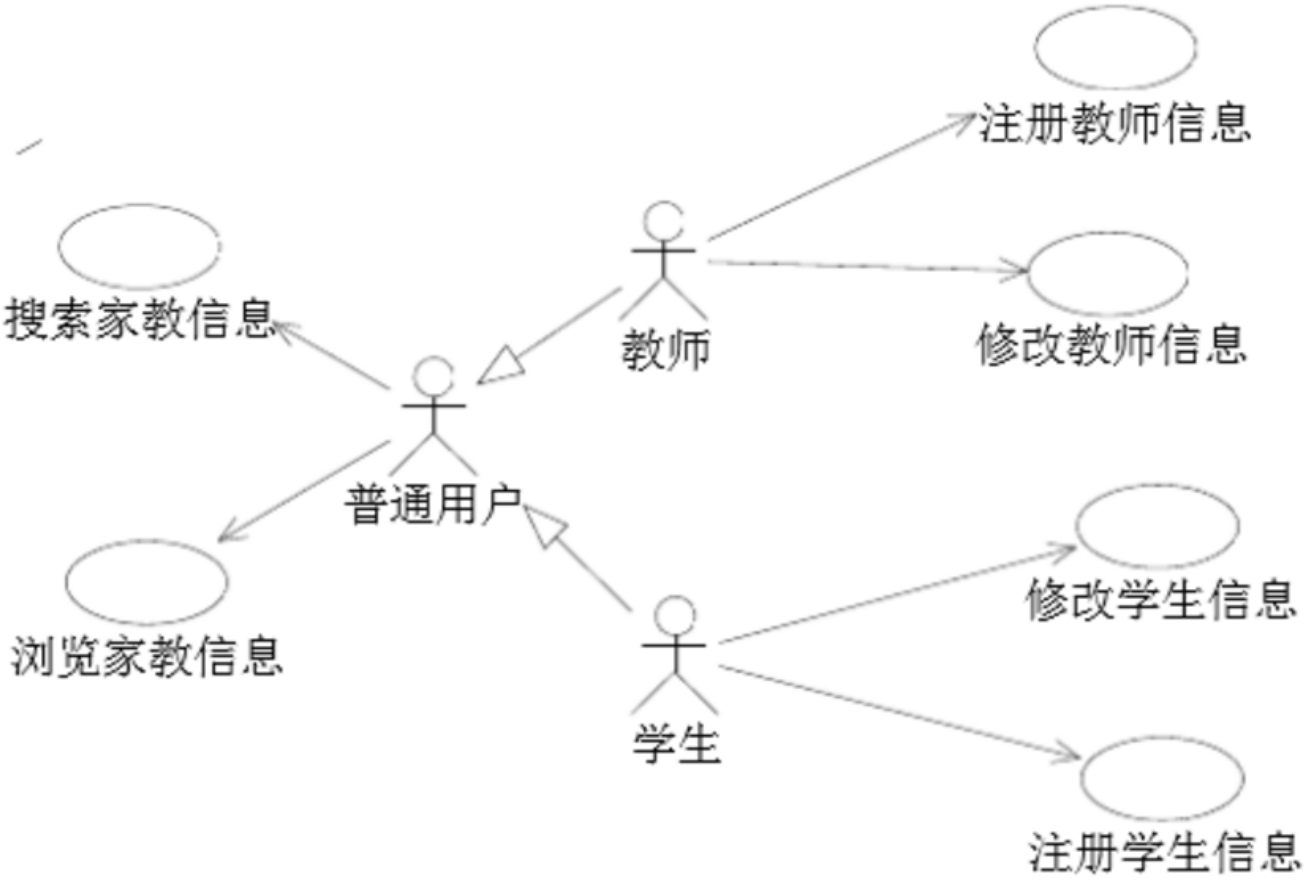
作答

正常使用主观题需2.0以上版本雨课堂

课堂作业

普通用户可以浏览家教信息、搜索家教信息；

家教老师和家教学生还可以注册本人信息、修改本人资料。



面向对象的数据模型——类图

类图是由若干类关联在一起，反映系统或者子系统组成结构的静态图。

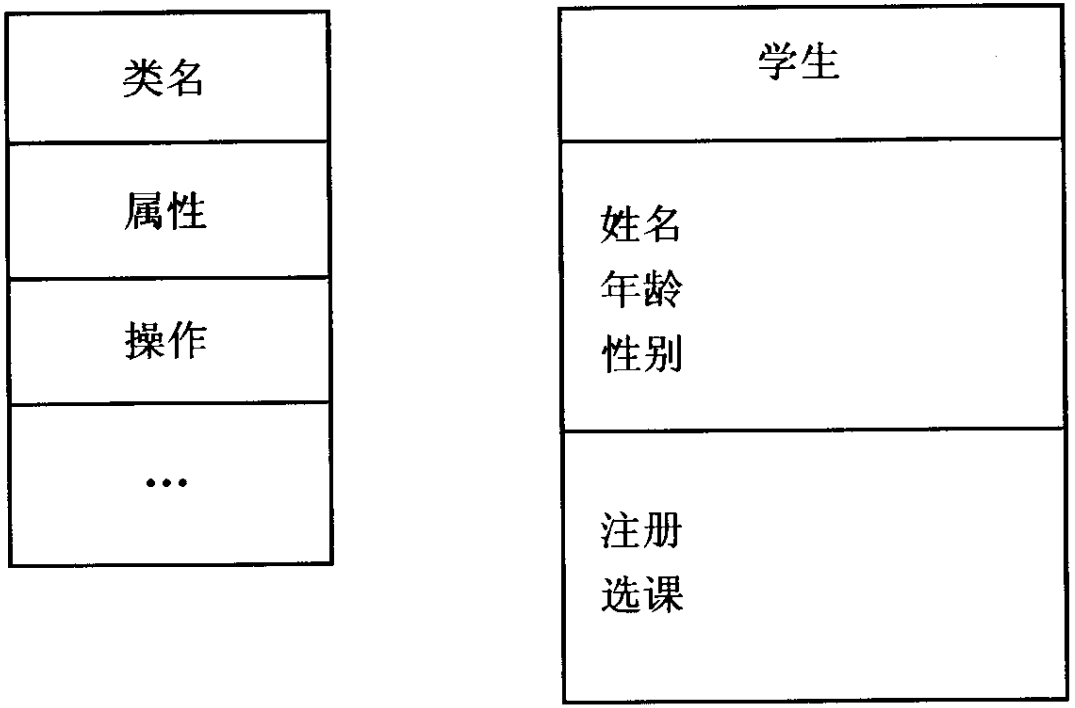
一、类图的组成

- 类(Class)：是具有共同结构特征、行为特征、联系和语义的对象集合的抽象形式。
- 关联（Association）：表示类与类之间的关系。

类图

(一) 类

在UML中通常以实线矩形框表示，矩形框中含有若干分隔框，分别包含类的名字、属性、操作、约束以及其他成分等。



类的图形表示和示例

类图

1. 属性

在UML类图标的矩形框中用文字串说明。

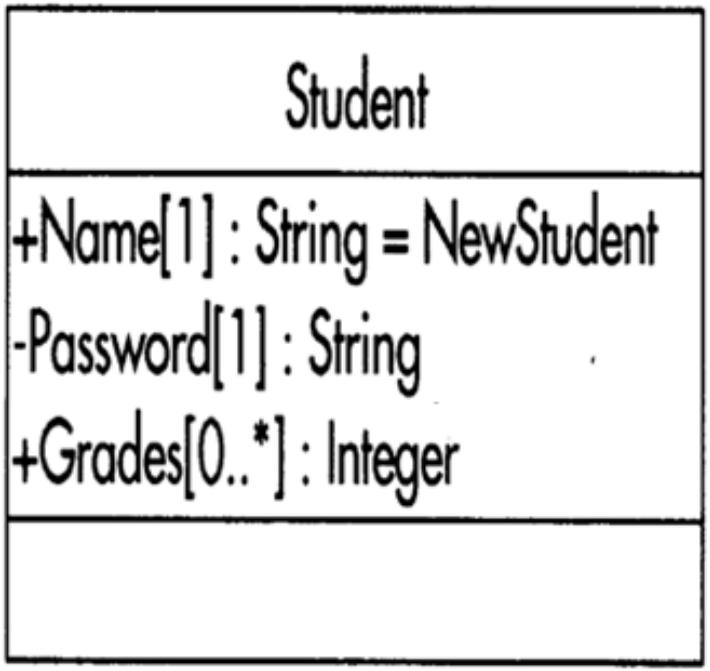
可视性　　属性名　　[多重性]：类型 = 初始值

- 可视性（Visibility）标记表示：
  - + 或 public　　公有
  - - 或 private　　私有
  - # 或 protected　　保护

类图

口属性类型表示：

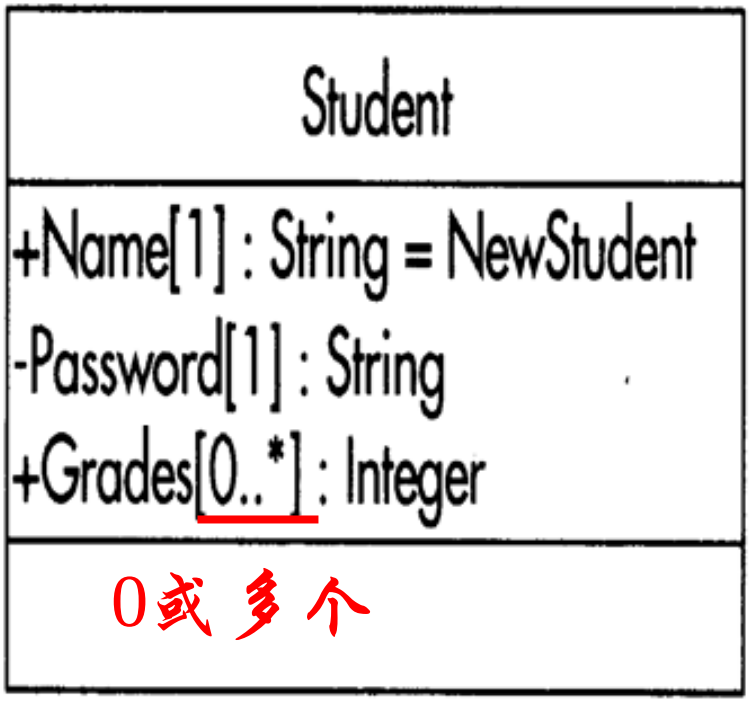
- “：”后跟属性值的数据类型。
- 数据类型的表示依赖于实现语言。
- 可通过在属性名和数据类型后添加 = 为属性指定初始值

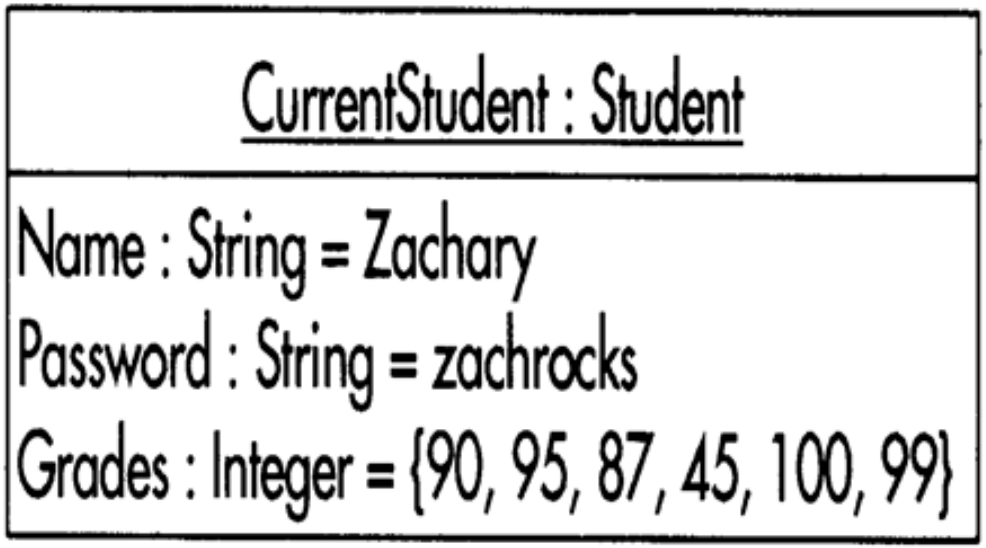


类图

口属性多重性：

- 可选项，表达属性值个数。如Student类的属性Grades包含该学生的所有成绩，可以是任意多个。



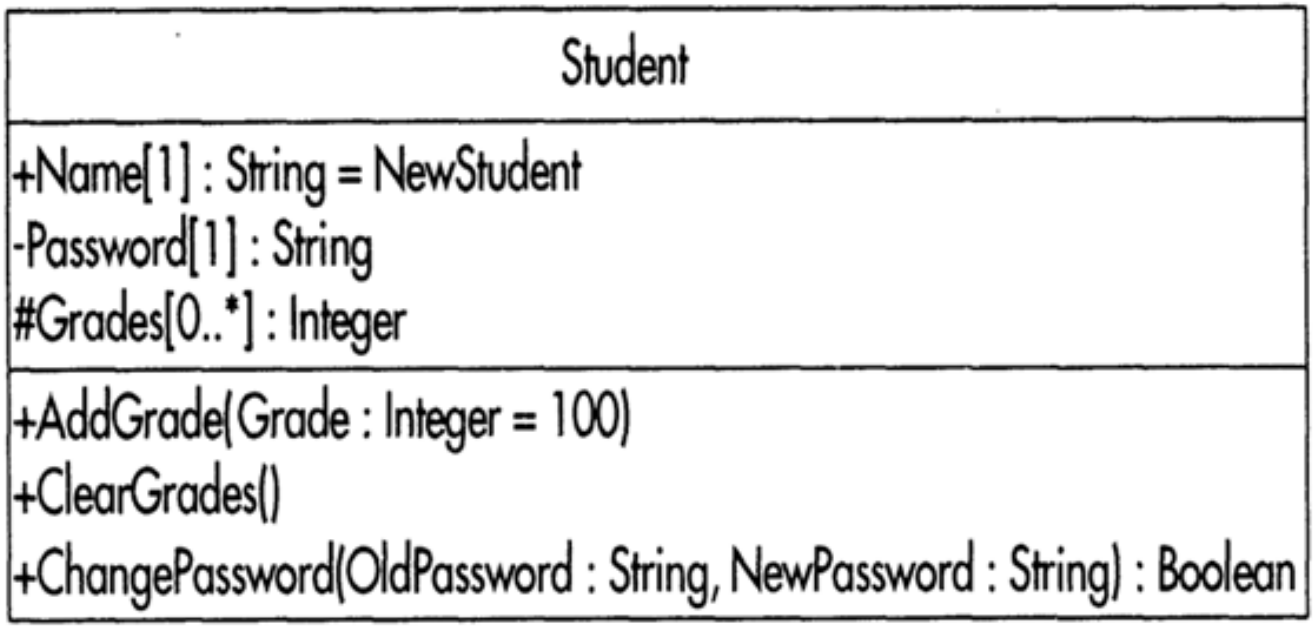


2. 操作(方法)

类提供的功能服务，在类矩形框中用文字串说明。

可视性 操作名(参数列表)：返回值类型

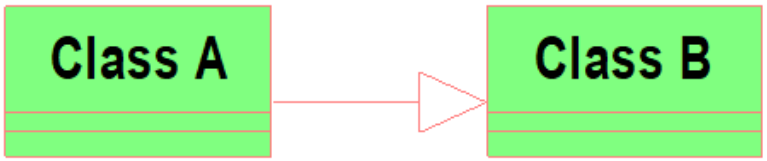
- **参数列表是可选项，格式为参数名：类型=缺省值，…**



（二）类的关系

关联关系

依赖关系



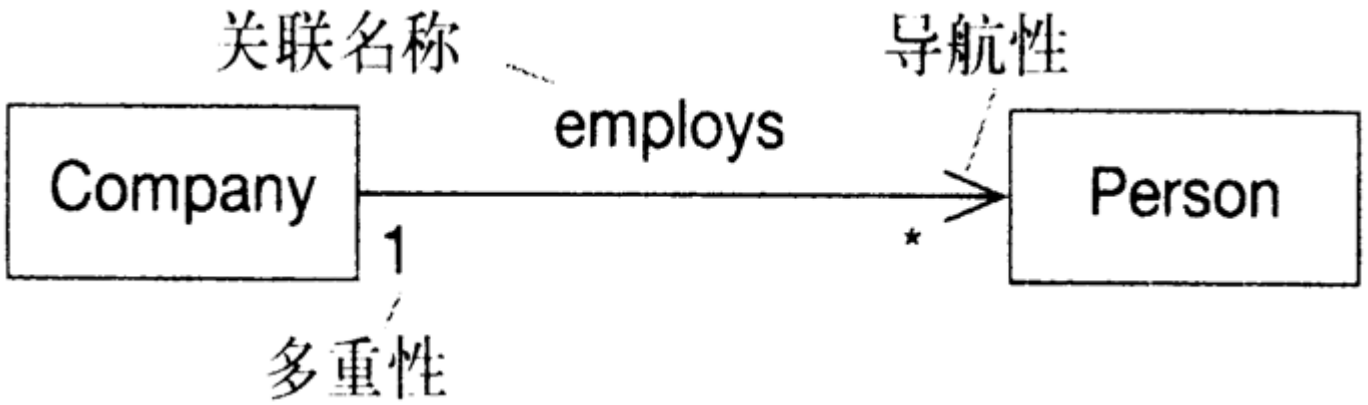
泛化关系

类图

1、关联关系

指类之间的语义联系。关联可以具有如下特性：

- **关联名称**(明确表达关联的含义)
- **多重性**(有多少个对象参与该关联)
- **导航性**(带箭头的实线表示单向关联，一个类可以访问另一个，反过来却不行)

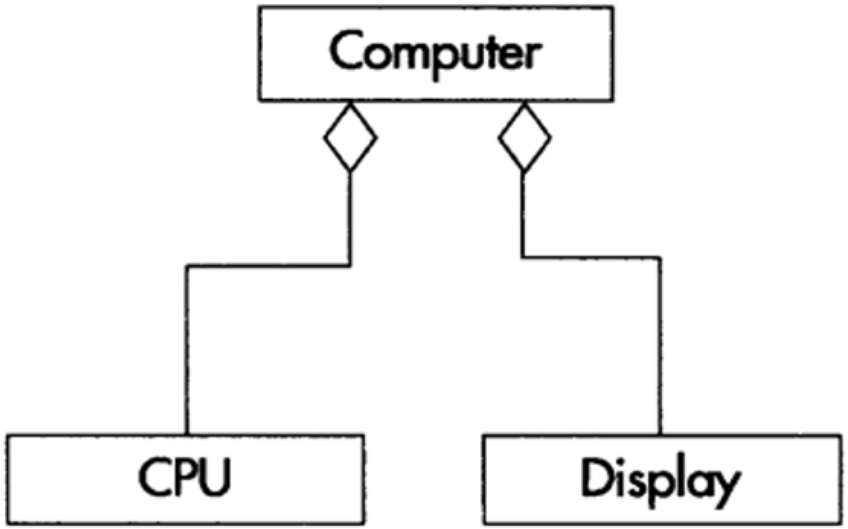


类图

特殊的关联：类之间的整体——部分组成关系

1）聚合

描述整体和部分的关系，其中一个类为整体，它由一个或多个部分类组成。**在聚合中，部分类可以没有整体类而存在。**

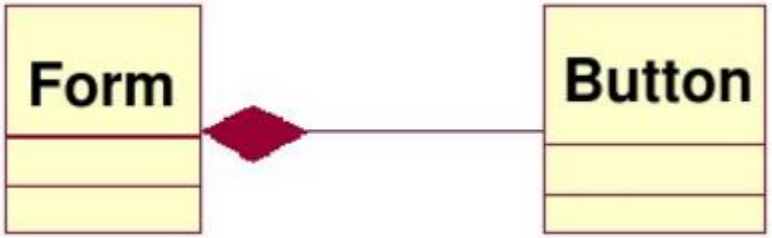


CPU和显示器都可以独立类的形式存在，但是当它们组成Computer类时，它们就变为整个计算机的组成部分。

类图

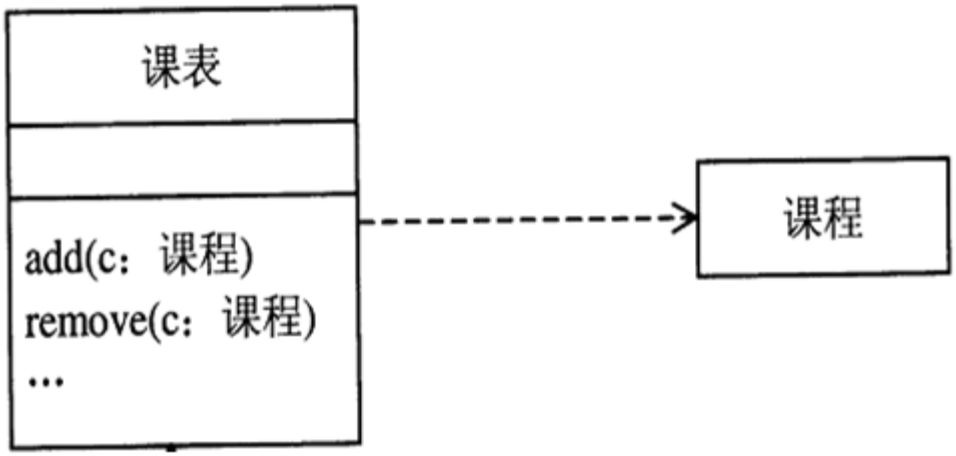
2）组合

特殊的聚合关联。组合关联中组成整体类的**部分类不能独立存在**，需要整体类才能存在。**组合关系中的“整体”控制着“部分”的生存期**，销毁整体类将会同时销毁部分类。使用带有实心菱形的实线连接。



2、依赖关系

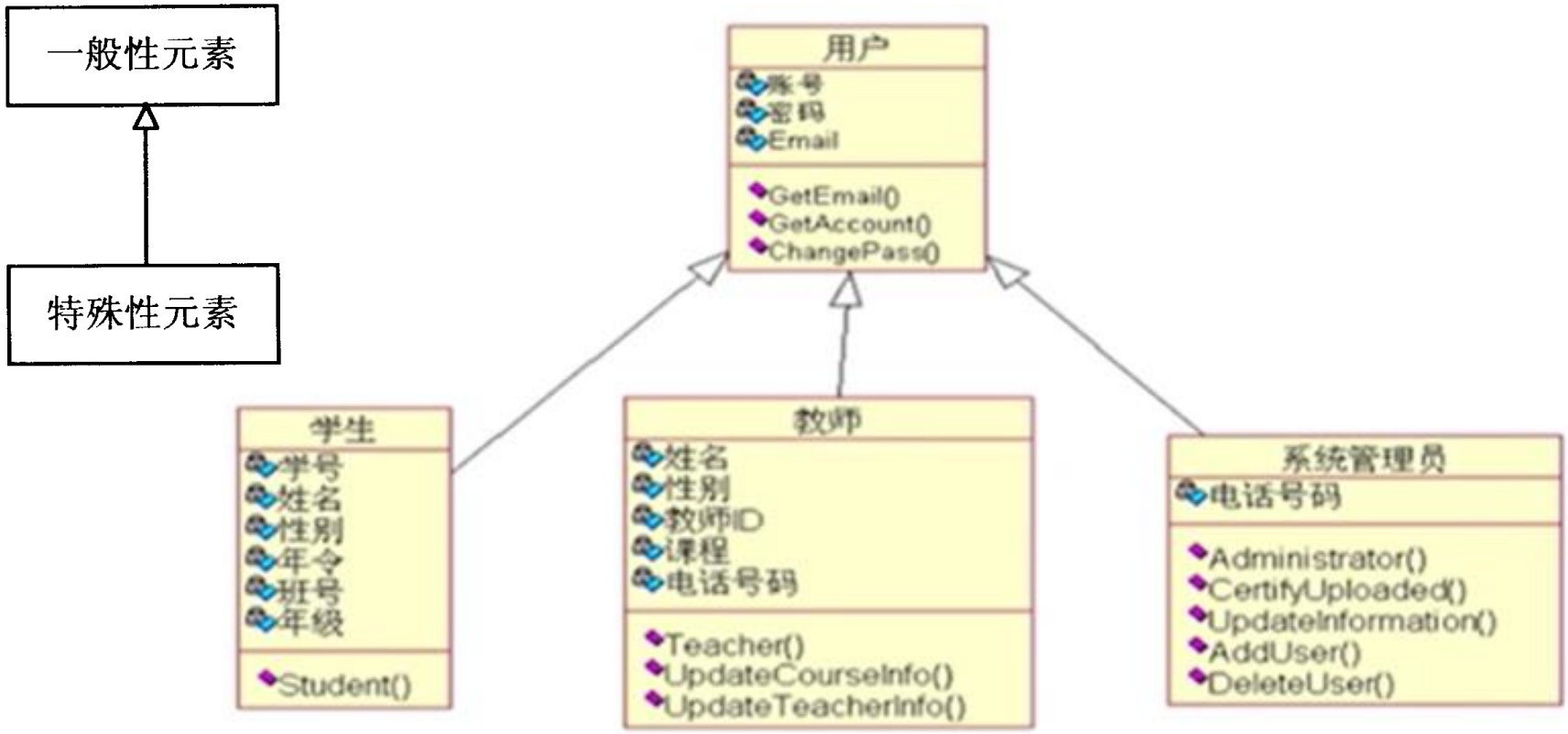
指一个类的元素使用了另一个类。依赖关系描述类之间的引用关系。



类图

3、泛化关系

描述类之间的继承关系。利用泛化来表达类之间的相似性，子类共享父类的属性和操作。



类图建模实例

名词/动词法为例

类或属性　　操作

小张是一个爱书之人，家里各类书籍已过千册，时常被外借，需要一个图书管理系统。该系统能将书籍的基本信息按计算机类、非计算机类分别建档，实现按书名、作者、类别、出版社等关健字的组合查询功能。使用该系统录入新书籍时系统会自动按规则生成书号，可以修改信息。该系统还需记录书籍的外借情况，打印外借情况列表。还能按特定时间周期统计书籍的购买金额、册数。

类图建模步骤

- 第一步：发现类(名词或名词短语)
- 第二步：筛选类
  - “小张”、“家里”明显是系统外的概念，无须建模;
  - “图书管理系统” 是指系统本身，也不必建模;
  - “书籍”显然是重要的类，“书名”“作者”“类别”“出版社”“书号”等用于描述书籍的应作为类的属性。“规则”可以作为编写“书籍”类构造函数的指南。
  - “基本信息”“关键字”是指书籍信息，无须建模
  - “功能”“新书籍”“信息”“记录”都是在描述需求时使用到的一些相关词语，可以淘汰。

类图建模步骤

第二步：筛选类

- “计算机类” “非计算机类”是图书的两大类，应建模。并改名为“计算机类书籍” “非计算机类书籍” 减少歧义
- “外借情况” 用来表示一次借阅行为，应该成为一个候选类“借阅记录”，多个外借情况组成“借阅记录列表”。“购买金额” “册数”都是统计结果，是数字，不必建模，而“特定时限”则是统计范围，不必建模。
- 可发现一个隐藏的关键类——书籍列表，也就是执行统计的主体

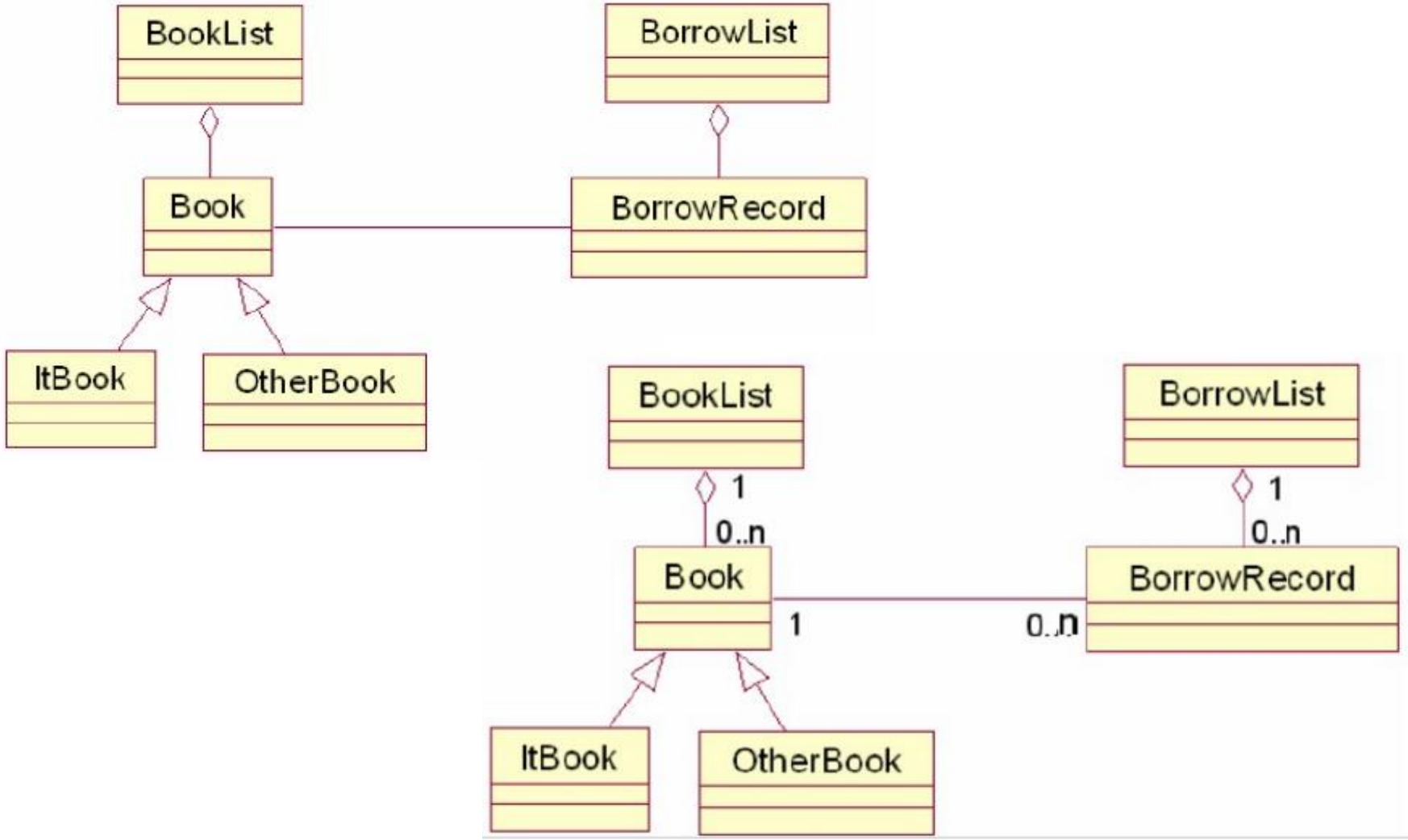
类图建模步骤

- 第三步：得到候选类
  - 书籍、计算机类书籍、非计算机类书籍，借阅记录、借阅记录列表、书籍列表

在使用名词动词法寻找类时，很多团队会耗费大量时间，特别是对于大中型项目，建议无须咬文嚼字，抓住本质，关键是对问题领域建立概要的了解。

类图建模步骤

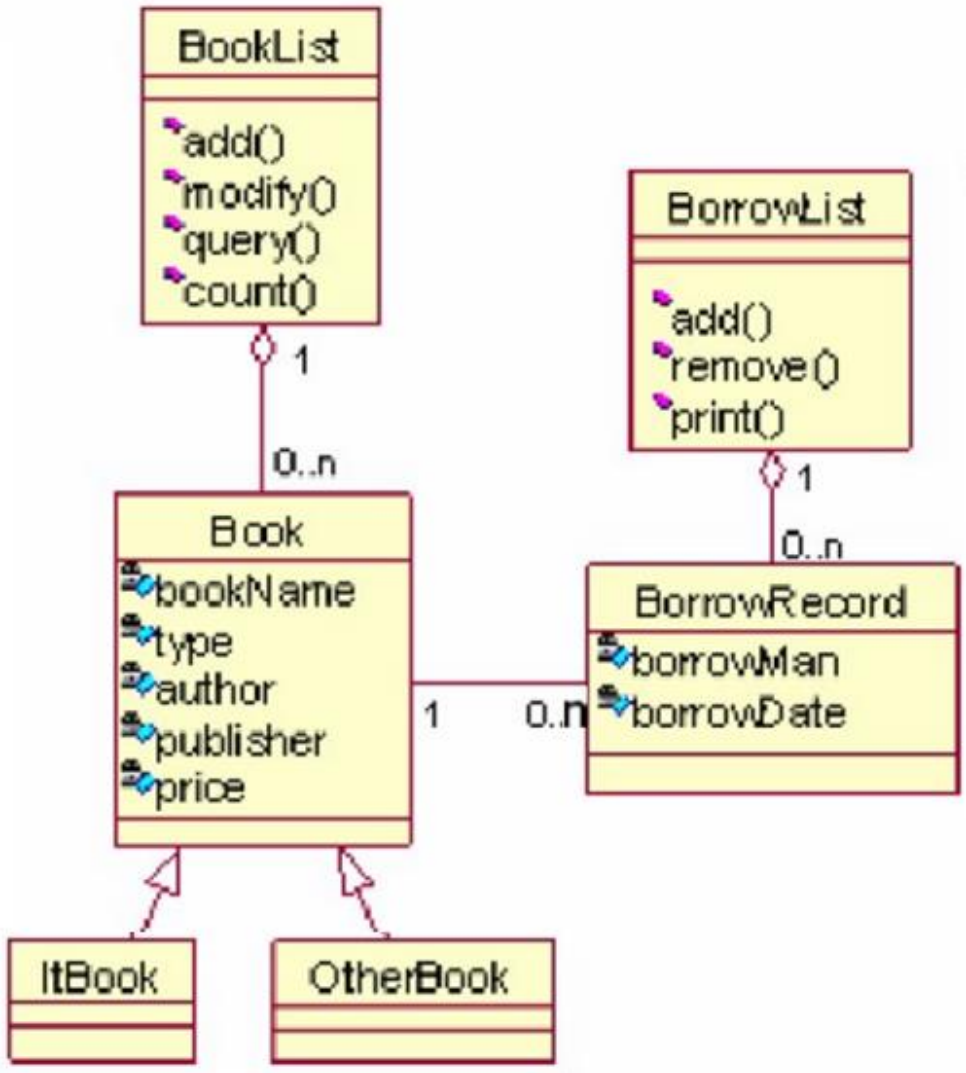
- 第四步：关联分析，建模，多重性分析，再建模



类图建模步骤

第五步：标识属性、定义操作

- 书籍类：从需求描述中可找到书名、类别、作者、出版社，从统计需要中可知“定价”也是一个关键属性。
- 书籍列表类：全部的藏书列表，其主要的成员方法是新增、修改、查询(按关键字)、统计(按特定时限统计金额与册数)。
- 借阅记录类：借阅人、借阅时间。
- 借阅记录列表类：主要职责就是添加记录(借出)、删除记录(归还)以及打印。



二、软件需求规格说明

- 是需求分析阶段得出的最主要的文档，对待开发系统的完整描述。
- 需求分析工作完成的一个基本标志是形成一份完整的、规范的需求规格说明书，准确地表达用户的需求。

需求规格说明书的编制是为了使用户和软件开发者双方对该软件的初始规定有共同的理解，使之成为整个开发工作的基础。

3.3 分析建模与规格说明

- 需求规格说明书从用户的角度描述软件产品的功能、输入、输出、界面、功能的边界问题、异常情况等，不涉及软件内部的实现细节。
  - **谁来写？** 通常是项目的PM，或者是有一定经验的开发或测试人员
  - **谁来实现？** 开发人员、UI设计人员
  - **谁来验证是否全部实现了？** 一般是质量保障人员(QA)

怎么才能写好规格说明书（Specification）

3.3 分析建模与规格说明

例：写一份“系鞋带”的说明书给外星人描述地球人是如何系鞋带的，你会怎么写?

提示：定义好相关的概念

规范好一些假设

界定一些边界条件(避免误解)

描述主要的用户交互步骤

可能的副作用

服务质量的说明

主观题 10分

设置

写一份“系鞋带”的说明书给外星人描述地球人是如何系鞋带的?

提示：定义好相关的概念

规范好一些假设

界定一些边界条件(避免误解)

描述主要的用户交互步骤

可能的副作用

服务质量的说明

作答

正常使用主观题需2.0以上版本雨课堂

3.3 分析建模与规格说明

第一，定义好相关的概念

- 鞋、鞋带、系鞋带、解鞋带的概念?
- 系好鞋带的好处是什么?
- 系鞋带的目标是什么?
- “系好了”如何定义?

第二，规范好一些假设

- 鞋带是否已经穿在鞋上？要对什么样的鞋(拖鞋、凉鞋、运动鞋、溜冰鞋、靴子)系鞋带?

3.3 分析建模与规格说明

第三，避免一些误解，界定一些边界条件

- 什么是“鞋带绑紧了”，打死结算“系好了”吗？要打多少个结才算“系好了”？打好的鞋带能拖在地上吗？可列出一些有二义性的情况，让大家讨论，形成共识。

第四，描述主流的用户/软件交互步骤

- 用明确的步骤说明从“没系好”到“系好”的系鞋带过程

第五，一些好的功能还会有副作用

- 鞋带系的过松或过紧有什么副作用？明确说明副作用

第六，服务质量的说明

- 要明确服务质量是什么等级，意味着什么。

如何写好规格说明？——实践

Spec的两大敌人

乏味

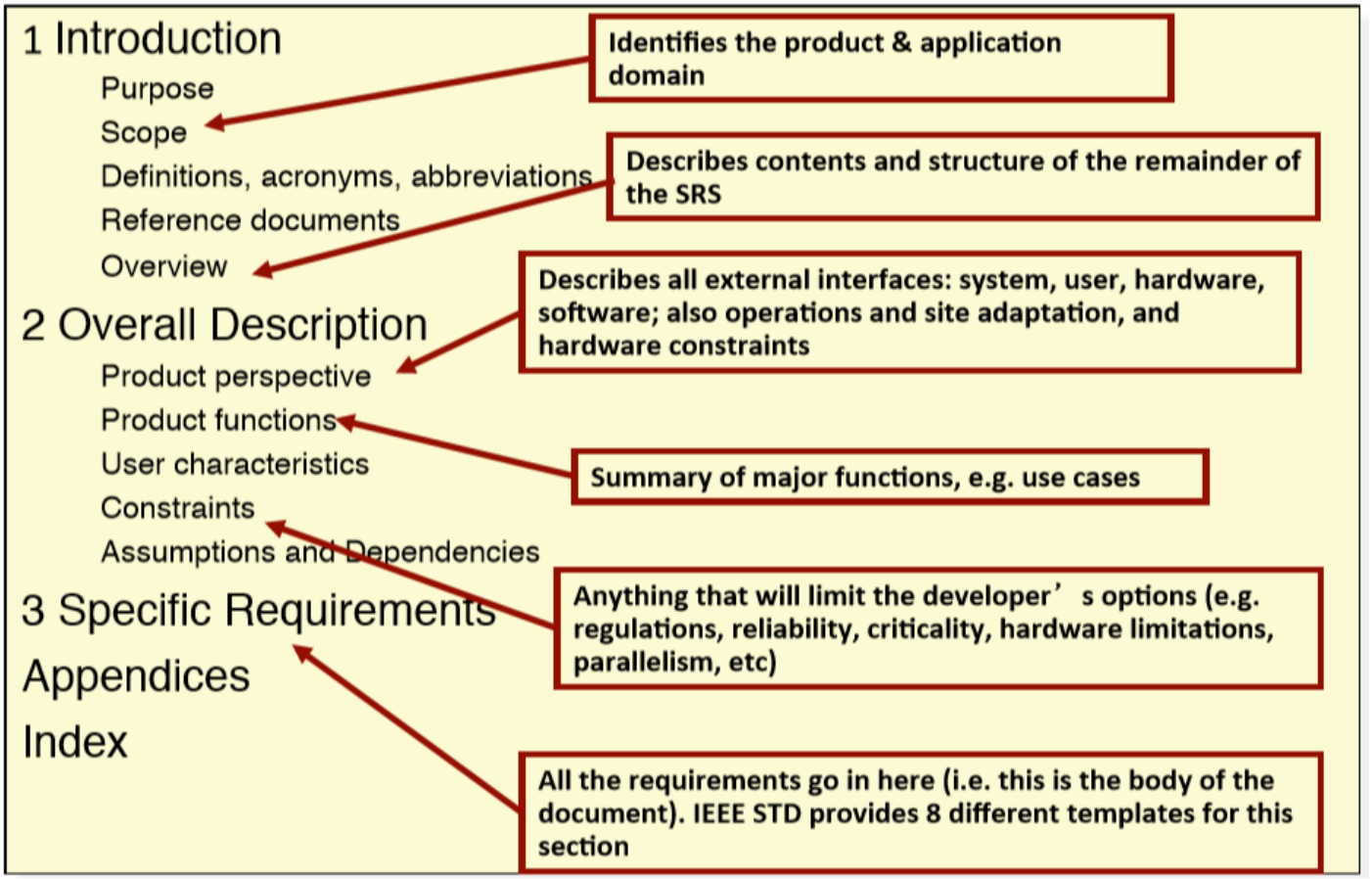
- 可以用人物和故事描述用户如何使用软件。
- 保持简单、直接的描述。涉及UI的部分可以直接上图，也可以画表格，不要写长篇累牍的文字。
- 规定清楚边界条件。……

时间

- 记录版本修订的时间和负责人——出了问题好找人。
- 说明哪些部分是容易发生变化的，提前做好预案。说明哪些部分如果发生改变，会有何种连锁反应。
- 做任何改动时，要事先参考Spec，事后更新Spec。……

规格说明书IEEE-830 SRS模板大纲

- 介绍
- 术语表
- 用户需求规格说明
- 系统结构
- 系统需求规格说明
- 系统模型
- 系统的演化
- 附录
- 索引



IEEE-830 SRS模板第三部分

3.1 外部接口

3.1.1 用户接口

3.1.2 硬件接口

3.1.3 软件接口

3.1.4 通信接口

3.2 功能需求描述

按系统的操作模式、用户分类、特征分类来定义:

3.2.1 模式 1

3.2.1.1 功能需求 1.1

…

3.2.2 模式 2

3.2.2.1 功能需求 2.1

…

…

3.2.n 模式 n

3.3 性能需求

要用可度量的方式来表达!

3.4 设计约束

3.4.1 标准依从性

3.4.2 硬件限制

etc.

3.5 软件质量属性

3.5.1 可靠性

3.5.2 可用性

3.5.3 安全性

3.5.4 可维护性

3.5.5 可移植性

3.6 其他需求

SRS模板的优缺点

优点:

- 模板提高效率
- 在有模板的情况下，面对一个完整的大纲，不容易遗漏重要的信息

缺点

- 并非对于所有的系统，模板的章节设 计都是类似的
- 如果仅仅为了满足标准，而填写模板 的所有章节，在不相关的章节，会加 入一些没有意义的内容
- 读者很难将这些无意义的文字和真正 的需求分开

规格说明书

不要盲目套用最全面的模板

- 目标是什么？目标不包括什么？
- 用户和典型场景是什么？
- 用到了哪些术语，他们的定义是什么？
- 用户是如何使用软件的功能的？
- 各种边界条件是什么、软件功能应该怎样随之变化？
- 功能有什么副作用，对其他功能有什么依赖关系？
- 什么叫“好”，什么叫“这个功能测试完了，可以交付了”
- 有哪些相关数据可以收集，如何准备数据收集的工作？
- ……

课程设计任务一

- 确定题目，然后开展需求分析工作，可采用面向过程或面向对象的分析方法，也可将两者结合使用。
- 初步完成软件需求规格说明书，需求分析文档需在GitHub上提交，**第一版提交时间为11月8日之前**
- 需简要说明采用了何种需求获取方式，如有材料证明的，请给出相关证明材料，如调查问卷、会议座谈的照片等。