

## 一、C++、python、Java 的优缺点、特点、适用范围：

### (1) C++

特点：支持数据封装和数据隐藏、支持继承和重用、支持多态性。

优点：1. 代码可读性好。2. 可重用性好。3. 可移植。4. C++设计成无需复杂的程序设计环境  
5. 运行效率高，高效安全 6. 语言简洁，编写风格自由。7. 提供了标准库 stl 8. 面向对象机制  
9. 很多优秀的程序框架包括 Boost、Qt、MFC、OWL、wxWidgets、WTL 就是使用的 C++。

缺点：1. 相对 java 来说，没有垃圾回收机制，可能引起内存设漏；2. 内容较多较难，学起来相对困难；

适用范围：c++更适合大项目，在大项目中 c++是其他言语不可比拟的。例如：游戏、分布式应用、操作系统、设备驱动程序等。做较小的项目，性能可能比不上 java。

### (2) java 语言

特点：简单性、面向对象、分布性、编译和解释性、稳健性、安全性、可移植性、高性能、多线程性、动态性。

优点：简单、安全、稳定、跨平台、强大的可伸缩性、多样化和功能强大的开发工具支持。

缺点：需要运行环境、不适合开发桌面应用程序、由于为了跨平台的功能，为了极度的伸缩能力，所以极大的增加了产品的复杂性。

适用范围：BS 结构的 ERP 系统、金融系统、电子商务系统、网站等，几乎所有的银行系统和银行网站都是基于 Java 平台的。

### (3) python 语言

特点：高层语言、可移植性、解释性、可扩展性、可扩充性、可嵌入性。

优点：简单易学、易读、易维护、规范的代码、免费开源、丰富的库。

缺点：单行语句和命令行输出问题、给初学者带来困惑、运行速度慢。

适用范围：Web 和 Internet 开发、科学计算和统计、人工智能、桌面界面开发、软件开发后端开发、网络接口：能方便进行系统维护和管理，Linux 下标志性语言之一，是很多系统管理员理想的编程工具。

## 二、面向对象方法和结构化方法的对比和适用范围：

结构化软件设计方法和面向对象软件设计方法是两种主流的软件设计方法。结构化软件设计方法更关注于系统的功能，采用自顶向下、逐步求精的设计过程，以模块为中心来解决问题，按照工程标准和严格的规范将目标系统划分为若干功能模块。面向数据流的方法和面向数据结构的方法是两种常用的结构化软件设计方法。面向数据流的设计方法多在概要设计阶段使用它借助于数据流图来进行设计工作，而面向数据结构的设计方法通常在详细设计阶段使用，它按输入、输出以及计算机内部存储信息的数据结构进行软件结构的设计，从而把对数据结构的描述转换为对软件结构的描述。常用的结构化软件设计工具有流程图、N-S 图和 PAD 图等。与结构化软件设计方法相比，面向对象软件设计方法的使用范围更广。与传统的软件工程方法不同的是，面向对象的方法不强调需求分析和软件设计的严格区分。从分析到设计的过渡，是一个逐渐扩充、细化和完善分析阶段所得到的各种模型的过程。面向对象的设计可以分为系统设计和对象设计两个阶段。系统设计关注于确定实现系统的策略和目标系统的高层结构，而对象设计是对需求分析阶段得到的对象模型的进一步完善、细化或扩充。

**结构化方法遵循基本原则有：**（1）抽象原则：是一切系统科学方法都必须遵循的基本原则，它注重把握系统的本质内容，而忽略与系统当前目标无关的内容，它是一种基本的认知过程和思维方式。（2）分解原则：是结构化方法中最基本的原则，它是一种先总体后局部的思想原则，在构造信息系统模型时，它采用自顶向下、分层解决的方法。

（3）模块化原则：结构化方法最基本的分解原则的具体应用，它主要出现在结构化设计阶段中，其目标是将系统分解成具有特定功能的若干模块，从而完成系统指定的各项功能。

**面向对象方法遵循基本原则有：**

(1) 抽象：抽象是指从事物中舍弃个别的、非本质的特征，而抽取共同的本质特征的做法。使用抽象的好处：一是便于访问，外部对象只需知道有限的几个操作（接口）即可访问其中的对象；二是便于维护，如果程序内部发生变化而其接口没有发生变化，则只需对其内部进行修改。

(2) 分类：分类的作用就是按照某种原则划分出事物的类别，以便有助于认识复杂世界。在面向对象中分类就是把具有相同属性和方法的对象化为一类，用类作为这些对象的抽象描述。分类实际上是把抽象原则运用于对象描述时的一种表现形式。

(3) 封装：在面向对象中封装就是用对象把属性和操纵这些属性的操作包装起来，形成一个独立的实体单元，体现了事物的相对独立性。封装的另一个含义则是信息隐蔽，即外界不能直接存取对象的内部属性以及隐藏起来的内部操作，外界也不用知道对象操作的内部实现细节。

(4) 多态性：多态性是指在具有继承关系的类层次结构中可以定义同名的操作或者属性，但是这些属性和操作具有不同的含义，即具有不同的数据类型或表现出不同的行为。

## 程序设计思想比较

### 结构化方法

结构化方法的程序设计偏重于过程，在其基本指导思想中，一个完整的程序是由算法和数据结构两部分组成的，开发人员的工作就是实现这两部分的内容，并用合适的方式使其二者建立连接，实现期望的程序功能。

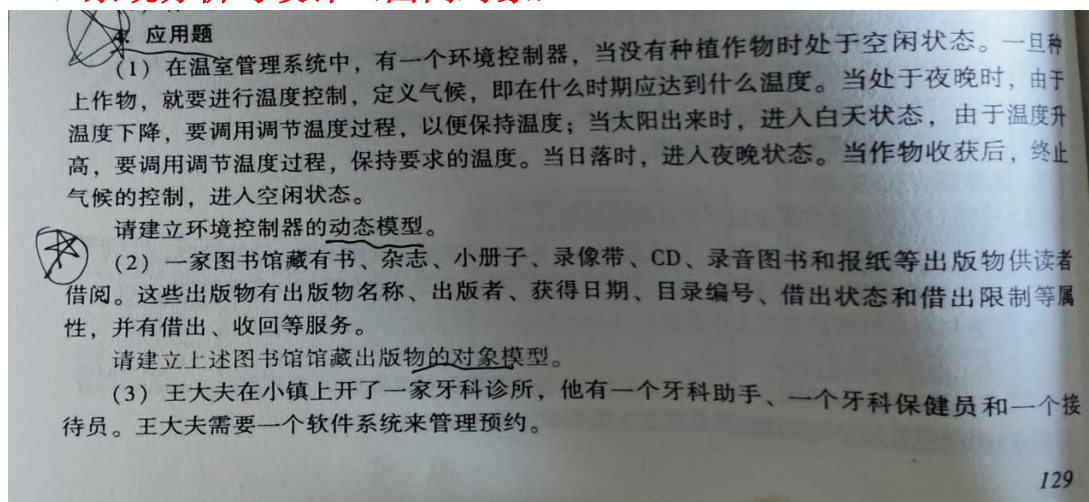
在具体设计流程上，结构化方法遵循了自底向上的设计思想，即先设计底层模块，确定模块内部算法，设计模块内部程序，并进行单元测试。模块完成之后，再进行模块之间的组装，进行模块的集成测试。随后进行确认测试和系统测试，完成设计和调试任务。

### 面向对象方法

对于面向对象方法的程序设计，其唯一组成是对象，对象之间的关系构成了程序的基本框架。由于对象是一个数据、功能统一的实体，故开发人员需要实现对象内部的数据结构与算法，不过显然实现难度比一个完整的程序要小很多。之后的关注点就是协调好对象之间的信息传递，使其能协同工作，发挥预期效果。

与结构化方法相反的是，面向对象方法采用自顶向下的设计思想，先设计父类，再设计子类，并继承父类的属性与方法，而对象则是在最后程序运行过程中动态生成。

## 三、系统分析与设计（面向对象）



**应用题**

(1) 在温室管理系统中，有一个环境控制器，当没有种植作物时处于空闲状态。一旦种上作物，就要进行温度控制，定义气候，即在什么时期应达到什么温度。当处于夜晚时，由于温度下降，要调用调节温度过程，以便保持温度；当太阳出来时，进入白天状态，由于温度升高，要调用调节温度过程，保持要求的温度。当日落时，进入夜晚状态。当作物收获后，终止气候的控制，进入空闲状态。

请建立环境控制器的动态模型。

(2) 一家图书馆藏书有书、杂志、小册子、录像带、CD、录音图书和报纸等出版物供读者借阅。这些出版物有出版物名称、出版者、获得日期、目录编号、借出状态和借出限制等属性，并有借出、收回等服务。

请建立上述图书馆馆藏出版物的对象模型。

(3) 王大夫在小镇上开了一家牙科诊所，他有一个牙科助手、一个牙科保健员和一个接待员。王大夫需要一个软件系统来管理预约。

129

当病人打电话预约时，接待员将查阅预约登记表，如果病人申请的就诊时间与已定下的预约时间冲突，则接待员建议一个就诊时间，以安排病人尽早得到诊治。如果病人同意建议的就诊时间，接待员将输入约定时间和病人的名字。系统将核实病人的名字并提供所记录的病人数据，数据包括病人的病历号等。在每次治疗或清洗后，助手或保健员需要标记相应的预约诊治已经完成，如果必要的话会安排病人下一次再来。

系统能够按病人姓名和日期进行查询，能够显示记录的病人数据和预约信息。接待员可以取消预约，可以打印出前两天预约尚未接诊的病人清单。系统可以从病人记录中获知病人的电话号码。接待员还可以打印出针对所有病人的每天和每周的工作安排。

请建立上述牙科诊所管理系统的功能模型。

三种模型都要写

- 环境控制器在被定义气候之前，处于闲置状态。在被定义气候之后，开始温度控制：当处于白天模式时，如果温度是升高，则进行调温操作；如果出现日落，则转换为夜间模式。当处于夜间模式时，如果温度降低，则进行调温操作；如果出现日出则转换为白天模式。当环境控制器被命令终止气候时，则重新处于限制状态。环境控制器的动态模型如图 C-13 所示。

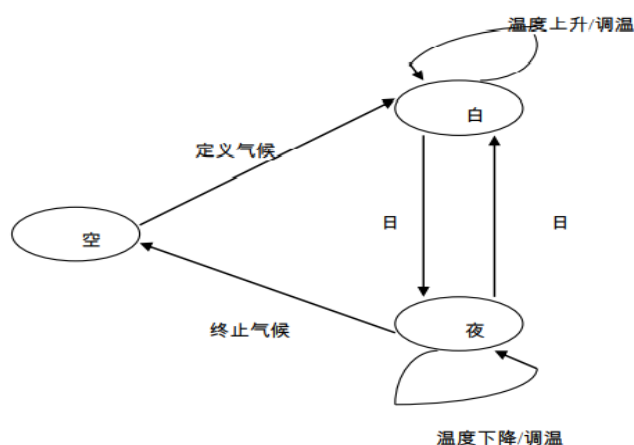
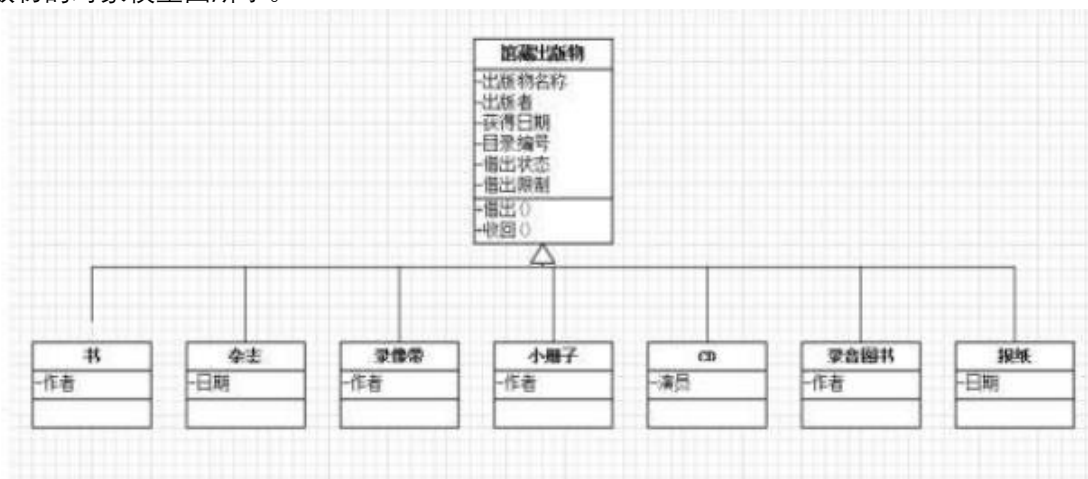


图 C-13 环境控制器的动态模型

- 图书馆所藏出版物拥有共同的基类：馆藏出版物类。馆藏出版物类的成员变量包括出版物名称、出版者、获得日期、目录编号、借出状态、借出限制等，并且包括借出和收回两项操作。各种类型的出版物类继承于馆藏出版物类，并且定义了各自的属性。图书馆馆藏出版物的对象模型图所示。



- (2) 从需求中可以看出，当病人进行预约时，需要提供姓名，预约日期。然后系统查询预约登记表，查看该日期是否有效，若预约成功，则记录入预约登记表。然后病人提供名字，系统核实，病人提供病人的病历号。在每次预约完成后，更新预约登记表，标记其已经完成，必要时也可以预约下次就诊日期。诊所的职员可以查询预约登记表，删除预约。系统可以提取每天的预约打印出来给牙医。
- 根据上述功能，可以建立该系统功能模型，如图所示：

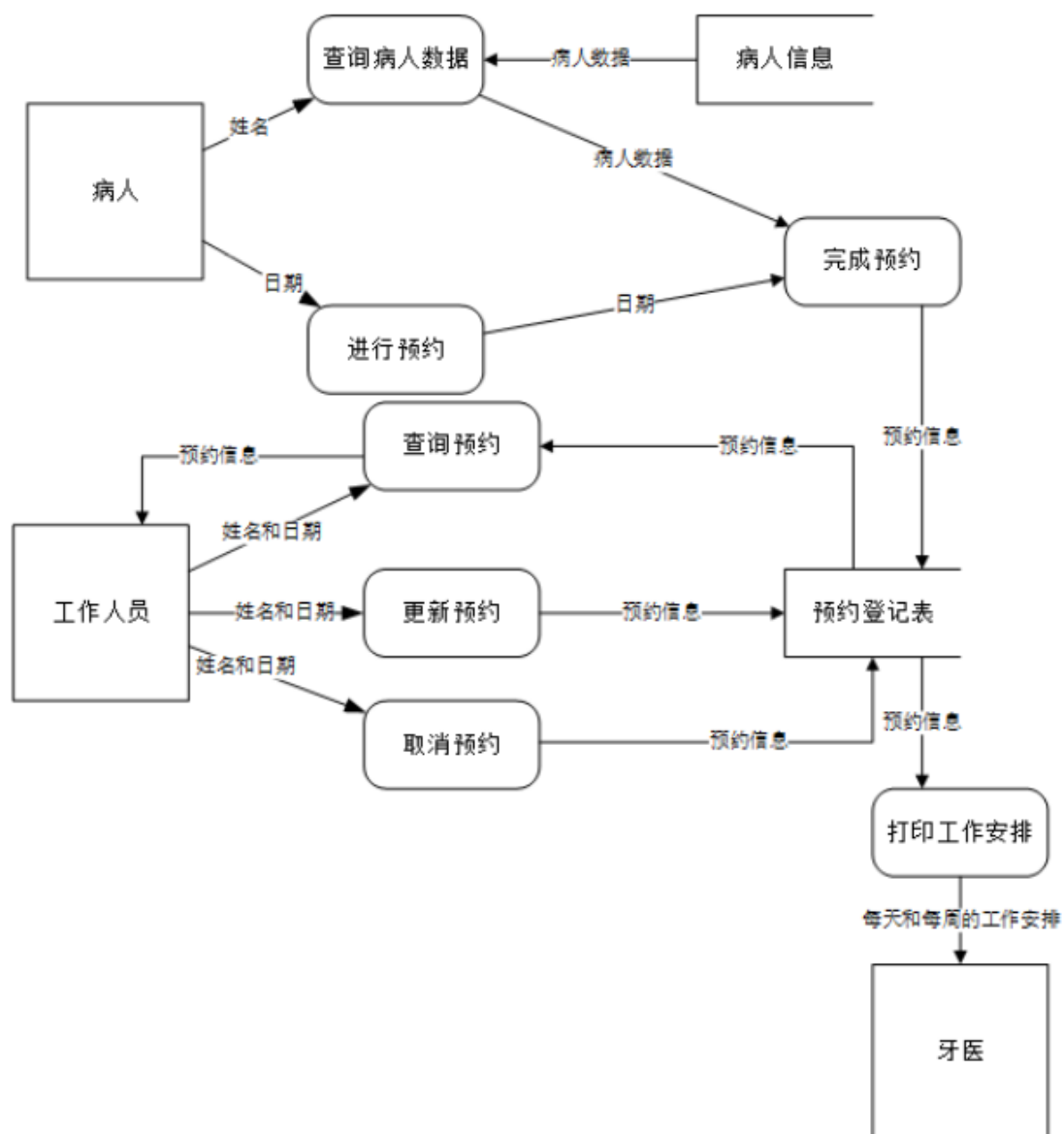


图 C-15 牙科诊所管理系统的功能模型

## 四、系统分析与设计（面向过程）



(1) 某图书管理系统有以下功能。

① 借书：输入读者借书证。系统首先检查借书证是否有效，若有效，对于第一次借书的读者，在借书文件上建立档案。否则，查阅借书文件，检查该读者所借图书是否超过 10 本，若已达 10 本，拒借，未达 10 本，办理借书（检查该读者目录并将借书情况登入借书文件）。

② 还书：从借书文件中读出与读者有关的记录，查阅所借日期，如果超期（3 个月）则进行罚款处理。否则，修改库存目录与借书文件。

③ 查询：可通过借书文件和库存目录文件查询读者情况、图书借阅情况及库存情况，打印各种统计表。

用结构化分析方法画出系统顶层图和 0 层图（数据流图），并写出数据字典。

(2) 根据以下描述画出相应的状态转换图。

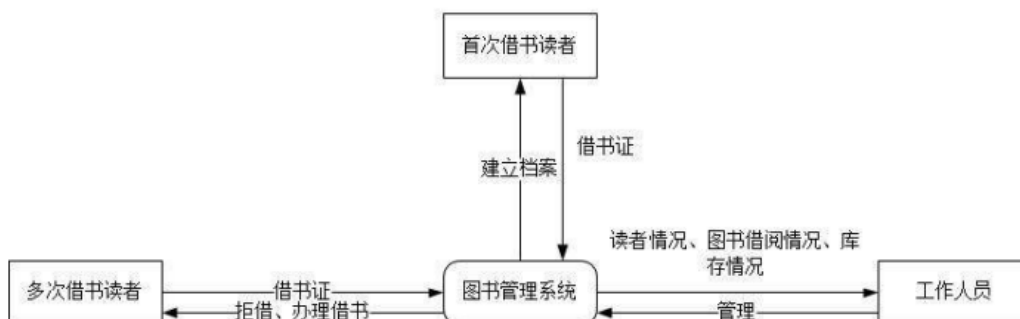
70

到 ATM 机前插入磁卡后输入密码，如果密码不正确则系统会要求再次输入密码，如 3 次输入不正确则退出服务；密码正确后，系统会提示选择服务类型，如选择存款则进行存款操作，存款完毕后可选择继续服务，也可以选择退出服务；如选择取款则进行取款操作，取款完毕后可选择继续服务，也可以选择退出服务。

(3) 某企业集团有若干工厂，每个工厂生产多种产品，且每一种产品可以在多个工厂生产，每个工厂按照固定的计划数量生产产品，计划数量不低于 300；每个工厂聘用多名职工，且每名职工只能在一个工厂工作，工厂聘用职工有聘期和工资。工厂的属性有工厂编号、厂址和地址，产品的属性有产品编号、产品名和规格，职工的属性有职工号、姓名和技术等级。画出 E-R 图。

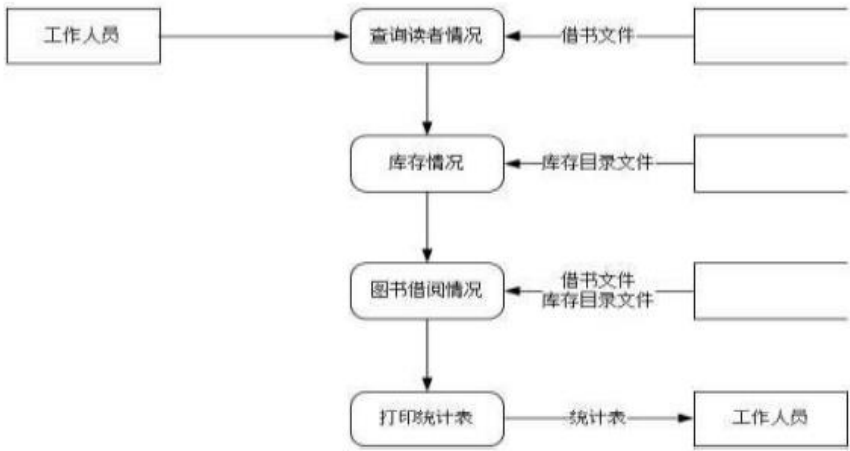
答案：

- 1) 顶层图：对于图书管理系统，外部用户有读者和管理工作人员。读者分为第一次借书的读者和多次借书的读者，第一次借书的读者需要在借书文件上建立档案才能借书。工作人员需要对借书文件、库存数目文件进行修改，也能查阅读者情况、图书借阅情况、库存情况。系统顶层图如图 C-1 所示。



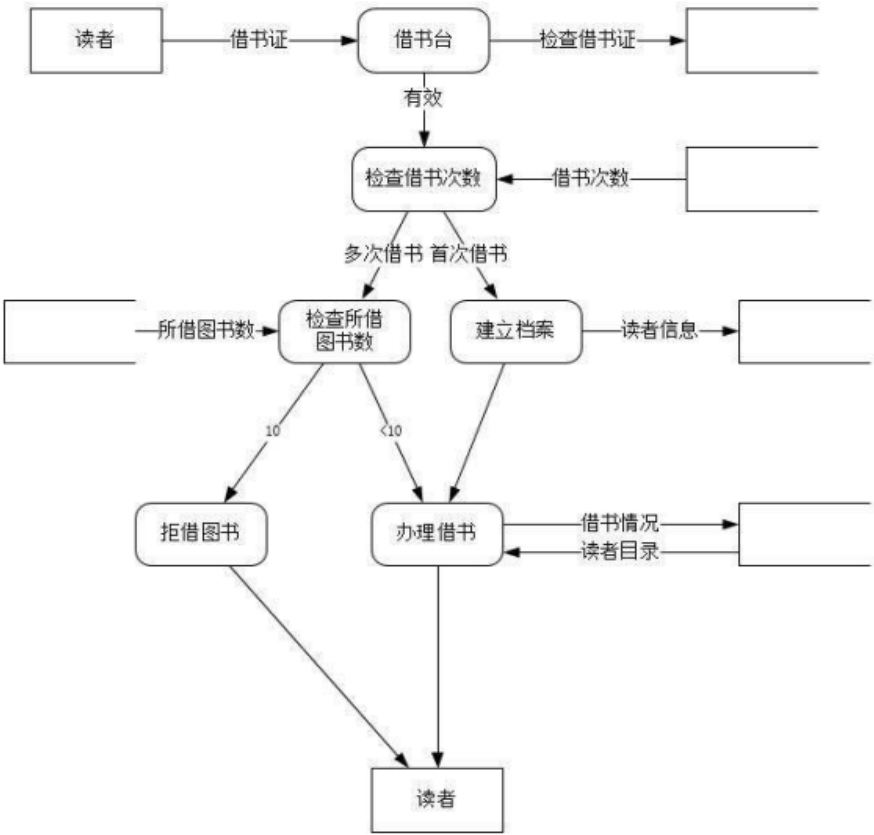
图C-1 系统顶层图

2) 查询 0 层图: 工作人员通过借书文件查询读者情况。通过库存目录文件查询库存情况。通过查询借书文件和库存目录文件查询图书借阅情况。最后打印统计表。查询 0 层图如图 C-2 所示。



图C-2 查询0层图

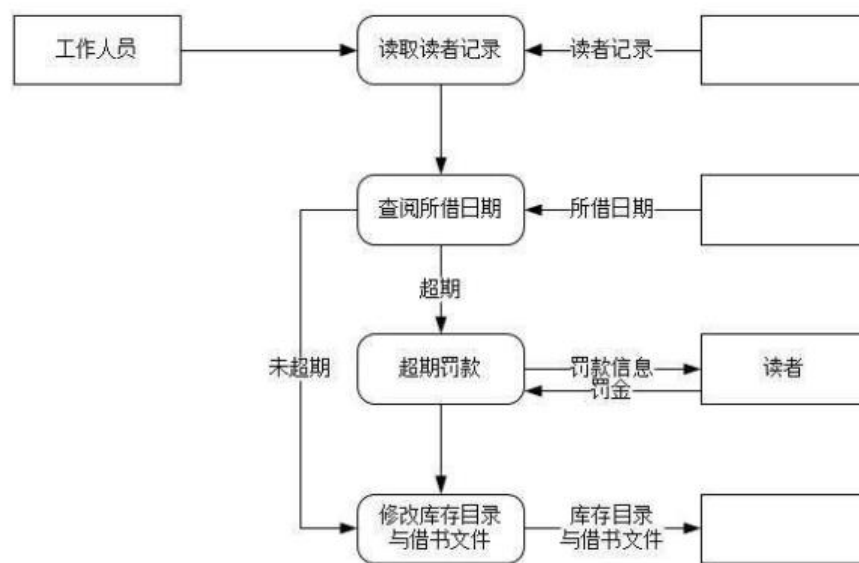
3) 借书 0 层图: 读者将借书证输入借书台, 借书台在系统中查询借书证是否有效, 若有效, 通过借书文件查看借书次数。若第一次借书, 则在借书文件中建立档案, 办理借书, 并将借书信息登入借书文件, 检查读者目录; 若为多次借书, 则从借书文件中检查所借图书是否超过 10 本, 若超过 10 本, 拒借, 并将信息反馈给读者, 否则办理借书, 并将借书信息登入借书文件, 检查读者目录。借书 0 层图如图 C-3 所示



图C-3 借书0层图

4) 还书 0 层图: 工作人员通过借书文件读取读者记录。通过读者记录查询所借日期。如果

借书超期，反馈读者罚款信息并收取罚款，并修改库存目录与借书文件。还书 0 层图如图 C-4 所示。



图C-4 还书0层图

##### 5) 数据字典

###### A . 顶层图数据字典:

首次借书读者 = {读者+借书证}

多次借书读者 = {读者+借书证}

借书证 = {姓名+学号}

读者 = {姓名+学号+班号}

工作人员 = {姓名+工作人员代号}

姓名: 2{汉字}4 学号: 8{数字}8

班号: 4{数字}4

工作人员代号: 4{数字}4

读者情况 = {姓名+学号}

图书借阅情况 = {图书名+图书编号+读者+库存数量}

库存情况 = {图书名+图书编号+库存数量}

图书名: {汉字}

图书编号: 6{数字}6

###### B . 借书 0 层图: 借书证 = {姓名+学号}

读者 = {姓名+学号+班号}

借书次数: 0..\*

读者信息 = {姓名+学号}

借书情况 = {读者+图书名+图书编号+所借日期}

读者目录 = {读者+图书名+图书编号+所借日期}

姓名: 2{汉字}4

学号: 8{数字}8

班号: 4{数字}4

图书名: {汉字}

图书编号: 6{数字}6

所借日期: 8{数字}8

C . 还书 0 层图:

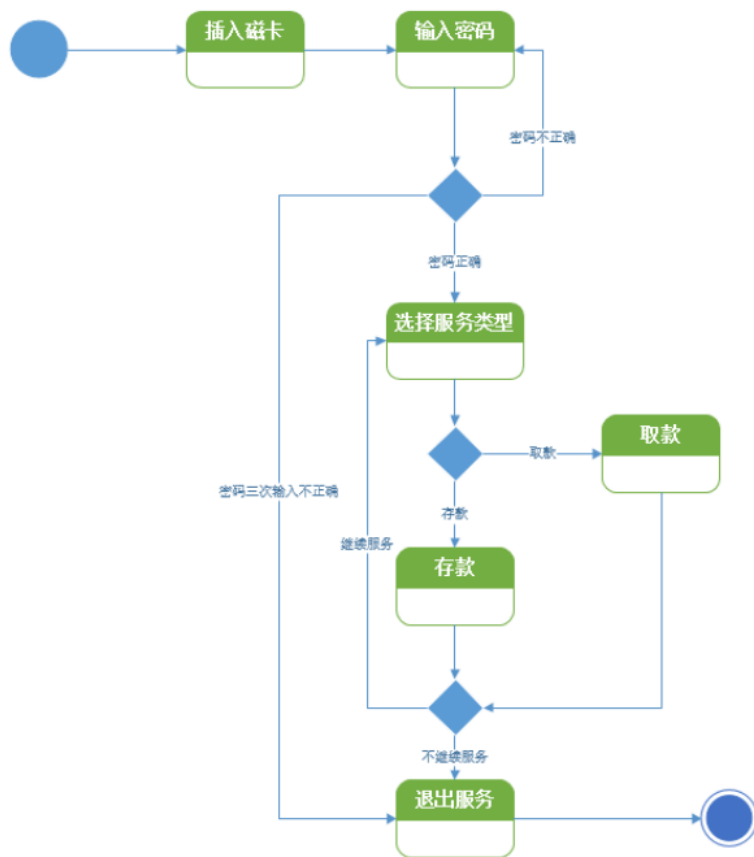
工作人员 = {姓名+工作人员代号}  
读者记录 = {读者+图书名+图书编号+所借日期}  
读者 = {姓名+学号+班号}  
罚款信息 = {读者+图书名+图书编号+罚金数额}  
库存目录 = {图书名+图书编号+库存数量}  
借书文件 = {读者+图书名+图书编号+所借日期}  
姓名: 2{汉字}4 学号: 8{数字}8  
班号: 4{数字}4  
工作人员代号: 4{数字}4  
图书名: {汉字}  
图书编号: 6{数字}6  
所借日期: 8{数字}8  
罚金数额: 1{数字}2

D . 查询 0 层图:

工作人员 = {姓名+工作人员代号}  
库存目录 = {图书名+图书编号+库存数量}  
借书文件 = {读者+图书名+图书编号+所借日期}  
统计表 = {库存目录+读者记录}  
读者记录 = {读者+图书名+图书编号+所借日期}  
姓名: 2{汉字}4  
学号: 8{数字}8  
班号: 4{数字}4  
工作人员代号: 4{数字}4  
图书名: {汉字}  
图书编号: 6{数字}6  
所借日期: 8{数字}8

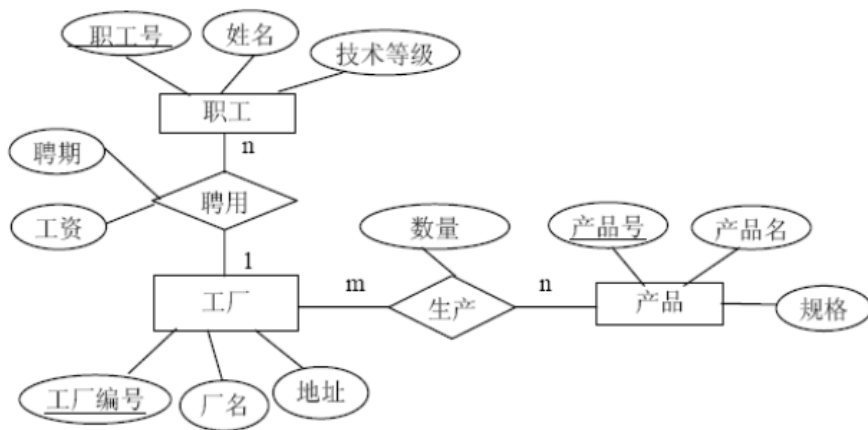
(2) 根据题目描述, 活动由插入磁卡开始, 输入密码后需对密码进行判断——若密码不正确, 则返回输入密码状态; 若三次输入密码都不正确, 则进入退出服务状态而结束; 若密码正确, 进入服务类型选择状态。在服务类型选择状态中, 需要再次判断, 若用户选择存款, 则进入存款状态; 用户选择取款, 则进入取款状态。存款或取款状态结束后, 继续进行判断, 若用户选择继续服务, 则再次进入服务类型选择状态, 否则进入退出服务状态而结束活动。其状态转换图如图 C-6 所示。





图C-6 状态转换图

(3) 根据题目中给出的一对多和多对多的关系，画出此 E-R 图，如图 C-7 所示。



图C-7 某企业集团工厂的E-R图

# 练习题

某图书公司欲开发一个基于Web的书籍销售系统，为顾客提供在线购买书籍的功能，同时对公司书籍的库存及销售情况进行管理。系统的主要功能描述如下：

（1）首次使用系统时，顾客需要在系统中注册。顾客填写注册信息表要求的信息，包括姓名、收货地址、电子邮箱等，系统将为其生成一个注册码；

（2）注册成功的顾客可以登录系统在线购买书籍。购买时可以浏览书籍信息，包括书名、作者、内容简介等。如果某种书籍的库存量为0，那么顾客无法查询到该书籍的信息。顾客选择所需购买的书籍及购买数量，若购买数量超过库存量，提示库存不足；若购买数量小于库存量，系统将显示验证界面，要求顾客输入注册码。注册码验证正确后，自动生成订单；否则，提示验证错误。如果顾客需要，可以选择打印订单。

（3）派送人员每天早晨从系统中获取当日的派送列表信息，按照收货地址派送顾客订购的书籍。

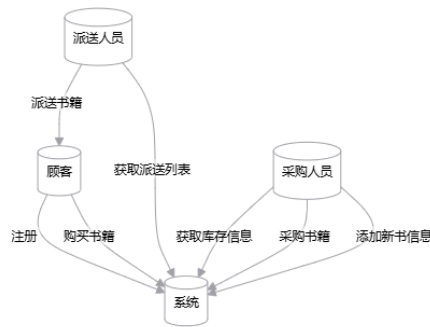
（4）用于销售的书籍由公司采购人员进行采购。采购人员每天从系统中获取库存量低于再次订购量的书籍信息，对这些书籍进行再次购买，以保证充足的库存量。新书籍到货时，采购人员向在线销售目录中添加新的书籍信息。

现采用面向对象软件工程方法，使用UML建模技术，完成该软件系统分析与建模，请分别给出：

- （1）业务用例和业务序列图；
- （2）系统用例图，并给出顾客“购买书籍”用例的用例规约；
- （3）系统类图。

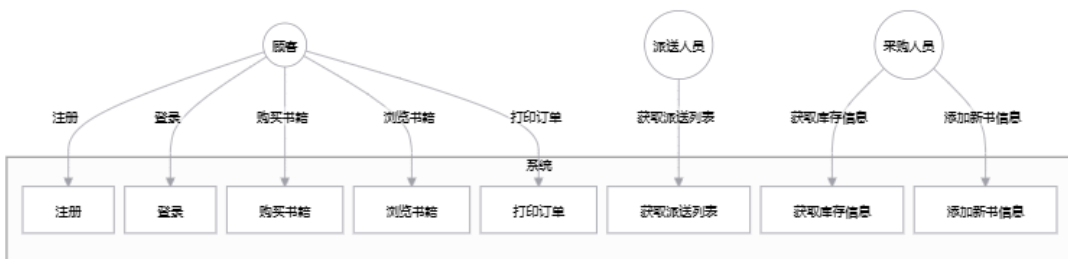
（1）业务用例图和业务序列图

业务用例图：



业务序列图（以购买书籍为例）：

（2）系统用例图



"购买书籍"用例规约:

1. 用例名称: 购买书籍
2. 参与者: 顾客
3. 前置条件: 顾客已注册并登录系统
4. 后置条件: 生成订单, 更新库存
5. 主事件流:
  - a. 顾客浏览可用书籍信息
  - b. 顾客选择要购买的书籍和数量
  - c. 系统检查库存
  - d. 系统显示验证界面
  - e. 顾客输入注册码
  - f. 系统验证注册码
  - g. 系统生成订单
  - h. 系统显示订单信息
  - i. 系统更新库存
6. 分支事件流:
  - a. 如果库存不足, 系统提示库存不足
  - b. 如果注册码验证失败, 系统提示验证错误
  - c. 如果顾客要求打印订单, 系统打印订单
7. 异常事件流:
  - a. 如果系统出现故障, 提示系统错误, 终止购买过程

## 五. 软件过程模型

常用的软件开发过程模型: 编码—修正模型、瀑布模型、快速原型模型、增量模型、螺旋模型、喷泉模型

### 1. 编码—修正模型

含义: 这是一种类似作坊的开发方式, 对编写几百行的小程序来说还不错, 但这种方法对任何规模的开发来说都是不能令人满意的。

缺点: 低估分析和设计, 代码结构差, 难修改, 不重视需求, 开发风险大, 用户不满意, 难修改、难维护。

### 2. 瀑布模型

含义: 各项活动按自上而下, 相互衔接的固定次序, 如同瀑布逐级下落, 每项活动均处于一个质量环(输入-处理-输出-评审)中, 基本思想: “分而治之”。

适用范围: 结构化方法, 面向过程的软件开发方法、需求变化少、开发人员熟领域、低风险项目、使用环境稳定。

优点: 迫使开发人员采用规范方法、严格规定了每阶段必须提交的文档、要求各阶段的产品必须质量验证。

缺点：需求难开始就完全确定、开发期长，一旦修改，则损失惨重、不支持软件复用和集成技术、缺乏灵活性。

### 3. 原型模型

含义：针对软件开发初期需求难以确定，基本思想是快速建立原型、完善用户需求，原型可作为单独的过程模型使用，它也常被作为一种方法或实现技术应用于其它的过程模型中。

适应范围：已有产品（原型）、简单而熟悉的领域、有快速原型开发工具、进行产品移植或升级。

优点：处理模糊需求、用户参与、快速。

缺点：快速（弱功能）、资源规划和管理较为困难、对开发环境要求高。

### 4. 增量模型

含义：是一种渐进地开发逐步完善的软件版本的模型，该模型一般首先开发产品的基本部分，然后再逐步开发产品的附加部分，如字处理软件的开发。

适用范围：在整个项目开发过程中，需求都可能发生变化，客户接受分阶段交付、分析设计人员对应用领域不熟悉，难以一步到位、中等或高风险项目、用户可参与到整个软件开发过程中、使用面向对象语言、软件公司自己有较好的类库、构件库。

优点：在较短时间内向用户提交可完成部分工作的产品、开发人员可以一个构件一个构件地逐步开发、逐步增加产品功能可以使用户有较充裕的时间学习和适应新产品、采用增量模型比采用瀑布模型和快速原型模型需要更精心的设计。

缺点：软件体系结构必须是开放的、既要把软件系统看作整体。又要看成可独立的构件、多个构件并行开发，具有集成的风险。

### 5. 螺旋模型

含义：对于复杂的大型软件，开发一个原型往往达不到要求。螺旋模型将瀑布模型和增量模型结合起来，加入了风险分析。在该模型中，软件开发是一系列的增量发布，早期的迭代中，发布的增量可能是一个纸上的模型或原型，在以后的迭代中，逐步产生系统更加完善的版本。

适用范围：庞大、复杂、高风险的系统，内部开发的大规模软件项目

优点：风险驱动、质量保证、利于维护。

缺点：对开发人员要求高、要求用户参与阶段评价，对用户来说比较困难。

### 6. 喷泉模型

含义：喷泉模型是一种支持面向对象开发的模型，喷泉模型各阶段均采用了“对象”这一统一范式，整个过程看起来像喷泉从喷出到落下再喷出的周而复始过程产生的光滑水柱。

适用范围：适用于面向对象的软件开发过程。

优点：各个阶段没有明显的界限，开发人员可以同步进行开发。多次反复地增加或明确目标系统，而不是本质性的改动，降低错误的可能性。

缺点：由于喷泉模型在各个开发阶段是重叠的，因此在开发过程中需要大量的开发人员，不利于项目的管理。要求严格管理文档，使得审核的难度加大，尤其是面对可能随时加入各种信息、需求与资料的情况。

在选择软件过程模型时需要考虑以下几点。

- 符合软件自身的特性，如规模、成本和复杂性等

- 满足软件开发进度的要求
- 对软件开发的进行进行预防和控制
- 具有计算机辅助工具的支持
- 与用户和软件开发人员的知识和技能相匹配
- 有利于软件开发的和管理和控制