一、Java，C++，Python 各自的优缺点，特点，适用范围对比分析

1.Java：

（1）优点：

Java 是纯面向对象的语言；Java 是性能稳定且扩展性较强；Java 提供了很多内置的类库，通过这些类库，简化了开发人员的程序设计工作；Java 提供了对 Web 应用开发的支持；Java 具有较好的安全性和健壮性；Java 去除了 C++语言中难以理解、容易混淆的特性。

（2）缺点：

使用大量的内存。靠虚拟机运行，运行速度相对较慢；不能和底层打交道，不支持底层操作；启动时间慢；因为 Java 删除了指针，所以不如 C/C++等语言灵活。

（3）特点：

Java 语言是原生支持多线程的；Java 语言是动态的；Java 语言是体系结构中立的；支持单继承；分布式。

（4）适用范围：

Java 凭借着优秀的语言特性，成为众多企业应用青睐的对象。许多银行都使用 Java 来编写系统。在大型企业级应用中，Java 有着绝对的优势。同时 Java 还随着智能手机的普及，成为了安卓手机系统上软件开发的标准编程语言，安卓手机上的大量应用都是使用 Java 语言进行编写的。

2.C++：

（1）优点：

代码可读性好；可重用性好；可移植；C++设计成无需复杂的程序设计环境；运行效率高，高效安全；语言简洁，编写风格自由；提供了标准库 stl；支持多重继承；面向对象机制。

（2）缺点：

相对 java 来说，没有垃圾回收机制，可能引起内存设漏；内容较多较难，学起来相对困难；
语言的过度复杂和标准库的过度苍白。

（3）特点：

在 C 语言的基础上进行扩充和完善，使 C++兼容了 C 语言的面向过程特点，又成为了一种面向对象的程序设计语言；可以使用抽象数据类型进行基于对象的编程；可以使用多继承、多态进行面向对象的编程；可以担负起以模版为特征的泛型化编程。

（4）适用范围：

C++语言则应为自身的复杂性和标准问题，因此 C++语言用于一些非常复杂而又要求极高效率的领域，比如一些基础库、大型游戏等设施的开发。

3.Python：

（1）优点：

简单易学；可移植性；Python 既支持面向过程的函数编程也支持面向对象的抽象编程；可扩展性和可嵌入性；Python 标准库和第三库众多，功能强大。

（2）缺点：

运行速度慢；Python 代码加密困难。

（3）特点：

Python 采用强制缩进的方式使得代码具有极佳的可读性；开源。

（4）适用范围：

Python 语言简单易学、代码简单，它也是科研人员使用的语言，这也使得 Python 语言成为人工智能、大数据等高端技术的首选语言。

二、请对比面向对象需求分析方法和结构化需求分析方法。

- **面向对象需求分析方法：**
  强调以 **对象** 为核心，通过识别系统中的实体（对象）及其属性和行为，建立模型。
  - 基于现实世界的对象和概念，直接映射到系统设计中。
  - 对象包含数据（属性）和功能（方法）。

- **结构化需求分析方法：**
  以 **功能分解** 为核心，通过分析系统所需的功能和数据流，建立模型。
  - 强调将系统分解为一系列功能模块。
  - 使用数据流图（DFD）描述系统输入、输出及其处理过程。
- **面向对象需求分析：**
  - 通过识别 **类** 和 **对象**，建立对象模型（类图、状态图等）。
  - 关注系统的 **动态行为**，通过用例图和交互图描述对象交互。
  - 统一使用 UML（统一建模语言）进行建模。
- **结构化需求分析：**
  - 通过 **顶层分解** 和 **逐步细化**，完成功能分层。
  - 使用工具如 **数据流图（DFD）**、**实体关系图（ER 图）** 描述系统。
  - 强调功能模块之间的数据传递和依赖关系。

三．试比较瀑布模型、快速原型模型、增量模型和螺旋模型的优缺点，说明每种模型的适用范围(软件模型有什么优缺点)

软件过程模型	要点	优点	缺点	适用范围
瀑布模型	每个阶段都有**文档**产出	文档驱动的有序方法	只能通过文档了解产品，交付产品可能不符合客户的要求	**项目周期较短。需求是预知的**，软件实现方法是成熟的；
快速原型模型	不带反馈环，线性顺序进行，本质是"**快速**"	确保交付的产品符合客户的要求	**还没有证明无懈可击**	**无完整的需求说明**，只有一些基本要求
增量模型	每一个**增量均发布一个可操作产品**	增大投资的早期回报，能在较短的时间内，提供可完成部分工作的初步产品给用户；	**要求较高，要求开放的结构**，可能退化为建造-修补模型	**需求经常改变**，开发人员数量不够
螺旋模型	强调**风险分析**（快速原型+瀑布模型）+风险分析	结合上述所有模型的特性，风险驱动	**开发成本**，只能用于**大型**的内部软件产品，开发者必须精通风险分析和风险排除	适用于**庞大、复杂并具有高风险**的系统。
喷泉模型	**无间隙**	各个阶段没有明显的界限，开发人员可以同步进行开发。	**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	适用于**面向对象**的软件开发过程。
Rational统一过程（RUP）	迭代的，以**架构**为中心的，**用例驱动**的软件开发方法。四个阶段：初始阶段，精化阶段，构建阶段，移交阶段	针对**所有关键的开发活动**为每个开发成员提供了必要的准则、模版和工具指导，并确保全体成员共享相同的知识基础。简洁和清晰的过程结构，为开发过程提供较大的通用性。	缺少关于软件运行和支持等方面的内容，没有支持多项目的开发结构，这在一定程度上降低了在开发组织内大范围实现**重用**的可能性。	适用于大型的需求不断变化的复杂软件系统项目。
敏捷过程之极限编程（XP）	敏捷开发以**用户的需求进化**为核心，采**用迭代、循序渐进**的方法进行软件开发。极限编程（XP）是敏捷过程中最富盛名的一个	高适应性，以人为本，以测试为驱动	敏捷注重人员的沟通，忽略文档的重要性，若项目人员流动大太，又给维护带来不少难度，特别项目存在新手比较多时，老员工比较累。	需要项目中存在经验较强的人，要不大项目中容易遇到瓶颈问题。
微软过程	每一个生命周期发布一个递进的版本，各生命周期持续快速地迭代循环	综合了Rational统一过程和敏捷过程的优点	对方法、工具和产品等方面不够全面	适用于商业环境下具有有限资源和有限开发时间约束的项目。

四．类间的外部关系有几种类型?每种关系表达什么语义？

1. 关联（Association）

- **语义：**
  表示一个类的对象可以与另一个类的对象相关联，通常用于描述两者之间的业务逻辑关系。
- **特点：**
  - 是一种较弱的关系，通常表现为一个类持有对另一个类对象的引用。
  - 关联可以是单向或双向的。

    - 例如：在一个“学生-课程”系统中，“学生”和“课程”之间的关系是关联，表示学生选修了课程。
- **表示方法：**
  用实线连接两个类，箭头表示方向（可选）。

---

2. 聚合（Aggregation）

- **语义：**
  表示整体与部分的关系，部分可以独立于整体存在。
  例如：一辆汽车（整体）由轮胎、发动机等组成（部分），但轮胎可以脱离汽车单独存在。
- **特点：**
    - 是一种特殊的关联关系。
    - 整体和部分之间是“弱拥有”关系，部分可以属于多个整体。
    - 生命周期上，部分和整体相互独立。
- **表示方法：**
  用一个带空心菱形的实线连接两个类，菱形指向整体。

---

3. 组合（Composition）

- **语义：**
  表示整体与部分的关系，但部分不能独立于整体存在。
  例如：一本书（整体）由多个章节（部分）组成，章节依附于书的存在。
- **特点：**
    - 是一种更强的聚合关系。
    - 整体和部分之间是“强拥有”关系。
    - 生命周期上，整体销毁时，部分也随之销毁。
- **表示方法：**
  用一个带实心菱形的实线连接两个类，菱形指向整体。

---

4. 继承（Generalization/Inheritance）

- **语义：**
  表示“is-a”关系，即子类是父类的一种具体类型。
  例如：“猫”继承自“动物”，猫是一种动物。
- **特点：**
    - 子类继承父类的属性和方法。
    - 子类可以扩展父类的行为或重写父类的方法。
- **表示方法：**
  用一个带空心箭头的实线连接两个类，箭头指向父类。

---

5. 实现（Realization）

- **语义：**
  表示类实现了接口，通常用于描述接口与具体类之间的关系。
  例如：一个“飞行能力”接口可以被“鸟类”类实现。
- **特点：**
    - 是一种“can-do”关系。
    - 实现关系表示具体类实现了接口定义的行为。
- **表示方法：**
  用一个带空心箭头的虚线连接两个类，箭头指向接口。

---

6. 依赖（Dependency）

- **语义：**
  表示一个类依赖于另一个类来完成某些功能，但这种依赖是临时的、短期的。
  例如：“打印机”类需要“文档”类提供打印内容。
- **特点：**
    - 是一种较弱的关系。
    - 通常表现为方法参数、局部变量或对类的临时使用。
- **表示方法：**
  用一个带箭头的虚线连接两个类，箭头指向被依赖的类。

总结对比

关系类型	语义	强弱程度（从弱到强）	表示方法
依赖	短期的、临时的关系	弱	虚线 + 箭头
关联	长期的对象引用关系	较弱	实线
聚合	整体与部分，部分可独立存在	较强	实线 + 空心菱形
组合	整体与部分，部分依附整体	强	实线 + 实心菱形
继承	“is-a” 关系	很强	实线 + 空心箭头
实现	类实现接口的行为	很强	虚线 + 空心箭头

五．请简述面向对象分析的原则

面向对象分析的主要原则如下。

1．抽象

从许多事物中舍弃个别的、非本质的特征，抽取共同的、本质性的特征，就叫做抽象。抽象是形成概念的必须手段。

抽象原则有两方面的意义：第一，尽管问题域中的事物是很复杂的，但是分析员并不需要了解和描述它们的一切，只需要分析研究其中与系统目标有关的事物及其本质性特征。第二，通过舍弃个体事物在细节上的差异，抽取其共同特征而得到一批事物的抽象概念。

抽象是面向对象方法中使用最为广泛的原则。抽象原则包括过程抽象和数据抽象两个方面。过程抽象是指，任何一个完成确定功能的操作序列，其使用者都可以把它看做一个单一的实体，尽管实际上它可能是由一系列更低级的操作完成的。数据抽象是指根据施加于数据之上的操作来定义数据类型，并限定数据的值只能由这些操作来修改和观察。数据抽象是面向对象分析的核心原则。它强调把数据（属性）和操作（服务）结合为一个不可分的系统单位（即对象），对象的外部只需要知道它做什么，而不必知道它如何做。

2．封装

封装就是把对象的属性和服务结合为一个不可分的系统单位，并尽可能隐蔽对象的内部细节。

3．继承

特殊类的对象拥有的其一般类的全部属性与服务，称作特殊类对一般类的继承。

在面向对象分析中运用继承原则，就是在每个由一般类和特殊类形成的一般—特殊结构中，把一般类的对象实例和所有特殊类的对象实例都共同具有的属性和服务，一次性地在一般类中进行显式定义。在特殊类中不再重复地定义一般类中已定义的东西，但是在语义上，特殊类却自动地、隐含地拥有它的一般类（以及所有更上层的一般类）中定义的全部属性和服务。继承原则的好处是：使系统模型比较简练也比较清晰。

4．分类

就是把具有相同属性和服务的对象划分为一类，用类作为这些对象的抽象描述。分类原则实际上是抽象原则运用于对象描述时的一种表现形式。

5．聚合

聚合的原则是：把一个复杂的事物看成若干比较简单的事物的组装体，从而简化对复杂事物的描述。

6．关联

关联是人类思考问题时经常运用的思想方法：通过一个事物联想到另外的事物。能使人发生联想的原因是事物之间确实存在着某些联系。

7．消息通信

这一原则要求对象之间只能通过消息进行通信，而不允许在对象之外直接地存取对象内部的属性。通过消息进行通信是由于封装原则而引起的。在 OOA 中要求用消息连接表示出对象之间的动态联系。

8．粒度控制

一般来讲，人在面对一个复杂的问题域时，不可能在同一时刻既能纵观全局，又能洞察秋毫。因此需要控制自己的视野：考虑全局时，注意其大的组成部分，暂时不详察每一部分的具体的细节；考虑某部分的细节时则暂时撇开其余的部分。这就是粒度控制原则。

9．行为分析

现实世界中事物的行为是复杂的。由大量的事物所构成的问题域中各种行为往往相互依赖、相互交织。

六．请简述面向对象分析的过程

1.寻找类和对象（UML 中的类）首先，根据对问题的调查了解编写问题的需求陈述，从该陈述中找出问题空间中存在的事物，将事物抽象成对象；

2.接着，通过寻找和确定结构进一步扩展问题空间中的对象，根据需要按照主题将问题分解为不同的子问题，确定对象属性、对象和对象之间的实例关联；

3.然后，从需求陈述中找出问题空间中存在的行为，通过分析行为和事物之间的关系确定对象的行为，通过分析对象行为之间关系确定建立对象之间的消息关联；

4.最后，对对象规格进行详细说明，按照有关规范编写软件需求规格说明书和进行复审，完成对问题的面向对象分析建模。

5.识别类与对象（找出问题中的实体）

6.识别结构（类或对象之间的关系）

7.识别主题（指导读者理解大型、复杂模型的一种机制）

8.定义属性（对象的属性）

9.建立动态模型

10.建立功能模型

11.定义服务（类所能提供的服务，即行为）

七．什么是动态模型？对象模型？功能模型？

对象模型：描述对象、类及其相互之间关系的静态数据结构，用对象图加以描述。

动态模型：描述系统内对象的相互作用，通常采用跟踪图和状态图加以表示。

功能模型：描述系统内数据数据的变化工程，通常可采用数据流图/用例图形式加以表示。

1. 动态模型（Dynamic Model）

定义

动态模型是描述系统的时间依赖行为和动态变化的模型，主要用于捕获系统中对象的状态变化及其交互行为。

特点

- **描述内容：**
  - 系统中对象的状态。
  - 状态之间的转换条件。
  - 对象的时间序列行为。
- **主要建模工具：**
  - **状态图（State Diagram）**：描述对象在生命周期中的各种状态及状态转换。
  - **序列图（Sequence Diagram）**：描述对象间消息传递的时间顺序。
  - **活动图（Activity Diagram）**：表示系统或对象中涉及的活动流程。
- **应用场景：**
  - 对时间敏感或交互复杂的系统（如实时系统、通信系统）。

2. 对象模型（Object Model）

定义

对象模型是从结构的角度描述系统中对象、类及其静态关系的模型，主要用于分析系统的组成结构。

特点

- **描述内容：**
  - 系统中的类、对象。
  - 类的属性和操作。
  - 类之间的关系（如继承、聚合、关联等）。
- **主要建模工具：**
  - **类图（Class Diagram）**：表示类、对象以及它们的属性、操作和关系。
- **应用场景：**
  - 用于分析系统的静态结构，提供系统的蓝图。

3. 功能模型（Functional Model）

定义

功能模型是描述系统的功能以及功能之间数据流的模型，主要用于捕获系统的输入、输出及功能处理。

特点

- **描述内容：**
  - 系统的功能及其分解。

- 功能之间的数据流或控制流。

- **主要建模工具：**
  - **数据流图（Data Flow Diagram, DFD）：**描述系统功能模块之间的数据流和处理关系。
  - **用例图（Use Case Diagram）：**描述系统功能及其与外部参与者的交互。
- **应用场景：**
  - 数据处理型系统（如订单管理系统、数据分析系统）。

对比总结

维度	动态模型	对象模型	功能模型
关注点	对象的时间行为与交互	系统的静态结构	系统的功能与数据流
描述对象	状态、事件、消息	类、对象、属性、操作、关系	功能模块、输入、输出
主要工具	状态图、序列图、活动图	类图	数据流图、用例图
应用场景	实时性强、行为复杂的系统	需要静态结构分析的系统	数据处理型系统

七．软件过程模型实例

1.已经充分了解了的数据处理应用程序。

瀑布模型。原因：在软件开发的过程中，需求不发生或很少发生变化，并且开发人员可以一次性获取全部需求；软件项目的风险较低；开发者对软件应用领域很熟悉。

2.通过卫星通信连接计算机的新软件产品，假设之前没有开发卫星通信软件的经验。

螺旋模型。原因：没有相应的开发经验，风险较大，螺旋模型明确包括了开发的每一阶段中的风险处理，在每一次迭代中，风险分析通过原型构建进行，允许对处理风险可用的不同选项进行权衡。

3.作为电话交换系统控制器的软件产品。

快速原型模型。原因：已有产品或产品的原型；为简单而熟悉的行业和领域。先建立一个模拟的电话交换控制器原型系统，让用户提出修改意见，快速的修改原型系统，直到用户认为模拟的电话交换系统可以达成目的，再无需返工地以线性顺序开发，便可实现需求。

4.新的图书馆自动化软件，连接城市中不同的图书馆。

基于组件的开发模型。原因：有丰富的现有组件和系统框架，充分利用软件复用的思想，降低开发成本和风险，加快产品开发。图书馆自动化软件所需要的模块较多，如数据库索引、用户的识别及连接其他图书馆等，可采用基于组件的开发模型；从组件库中筛选相关的组件，然后开发人员根据需求设计或使用现有的成熟开发框架复用这些组件，最后将所有组件集合在一起，进行系统测试。

5.超大型的软件，可以使用一套旋转的卫星，用来在订阅者中提供、监视和控制移动电话通信。

统一软件开发过程模型。原因：作为超大型软件，功能覆盖面广，开发风险大，开发时间长。采用统一软件开发过程模型，可以从初始开始不断细化迭代，可以多次地执行各个工作流程，有利于更好地理解需求，设计出合理的系统架构，降低风险。

6.一个新的文本编辑器。

增量模型。原因：文本编辑器的特点是分为基本功能和拓展功能，基本功能实现简单，但是毫无特色，新的文本剪辑器的主要优点集中在拓展功能上，为此可采用增量模型。开发人员首先实现提供基本核心功能的增量组件，创建一个具备基本功能的子系统，然后不断地增量拓展功能，以此达到需求目的。

7.一种新语言的编译器。

统一软件开发过程模型。原因：采用统一软件开发过程模型，可以多次执行各个工作流程，有利于更好地理解需求，设计出合理的系统架构，并最终交付一系列渐趋完善的成果。

8.面向对象的软件开发。

由于面向对象的过程模型有快速原型模型、喷泉模型和统一软件开发过程模型等，所以要根据具体的开发特性来决定模型的选择。

9.一个大型软件产品的图形用户界面。

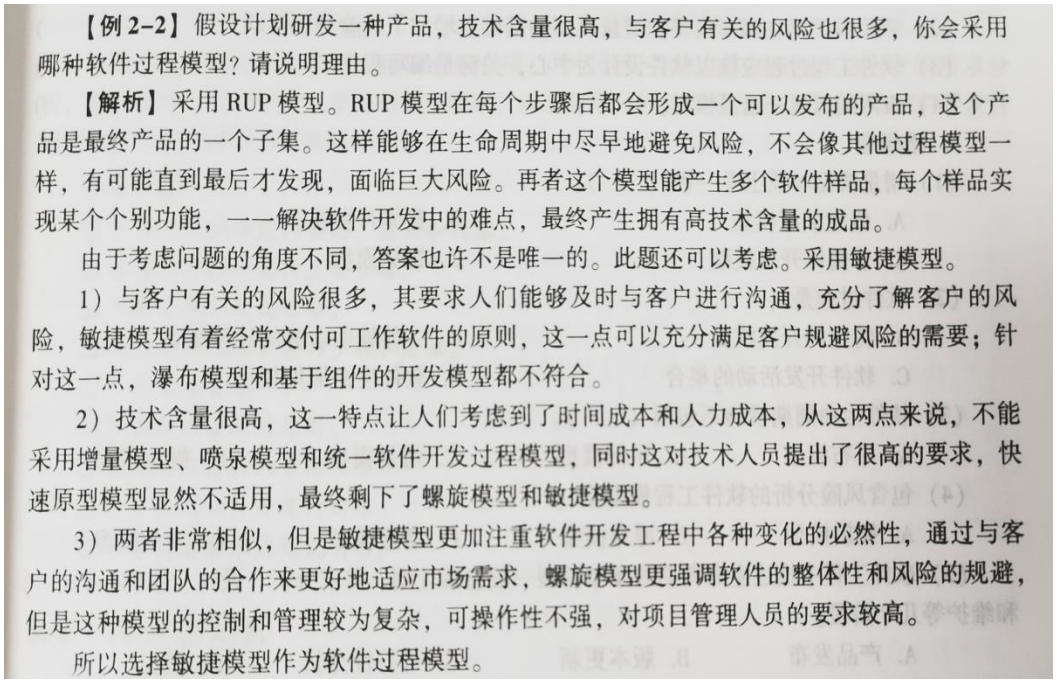
敏捷模型。原因：敏捷模型一般适用于小型项目和小型项目组，不大适用于大型项目和大型项目组。这里是对大型软件图形用户界面的开发，而不是对整个大型软件产品的开发，所以它应该算是小型项目。敏捷模型适用于软件需求常常发生变化的软件开发，大型软件的图形用户界面开发也需要用户深入交流协作，这些都是敏捷模型所擅长的。

10.当需求不能一次搞清楚，且系统需求比较复杂时。

开发模型不是孤立或排斥的，它们之间需要相互借鉴和参考。螺旋模型是一种综合性的模型，适合于较复杂的系统。

11.某大型企业计划开发一个“综合信息管理系统”，该系统涉及销售、供应、财务、生产和人力资源等多个部门的信息管理。该企业的想法是按部门优先级别逐个实现，边应用边开发。对此，需要一种比较合适的过程模型。请对这个过程模型做出符合应用需要的选择，并说明选择理由。

可以采用增量模型。它是瀑布模型与原型进化模型的综合，它对软件过程的考虑是：在整体上按照瀑布模型的流程实施项目开发，以方便对项目的管理；但在软件的实际创建中，则把软件系统功能分解为许多增量构件，并以构件为单位逐个创建与交付，直到全部增量构件创建完毕，并都被集成到系统之中交付用户使用。



【例 2-2】假设计划研发一种产品，技术含量很高，与客户有关的风险也很多，你会采用哪种软件过程模型？请说明理由。

【解析】采用 RUP 模型。RUP 模型在每个步骤后都会形成一个可以发布的产品，这个产品是最终产品的一个子集。这样能够在生命周期中尽早地避免风险，不会像其他过程模型一样，有可能直到最后才发现，面临巨大风险。再者这个模型能产生多个软件样品，每个样品实现某个个别功能，一一解决软件开发中的难点，最终产生拥有高技术含量的成品。

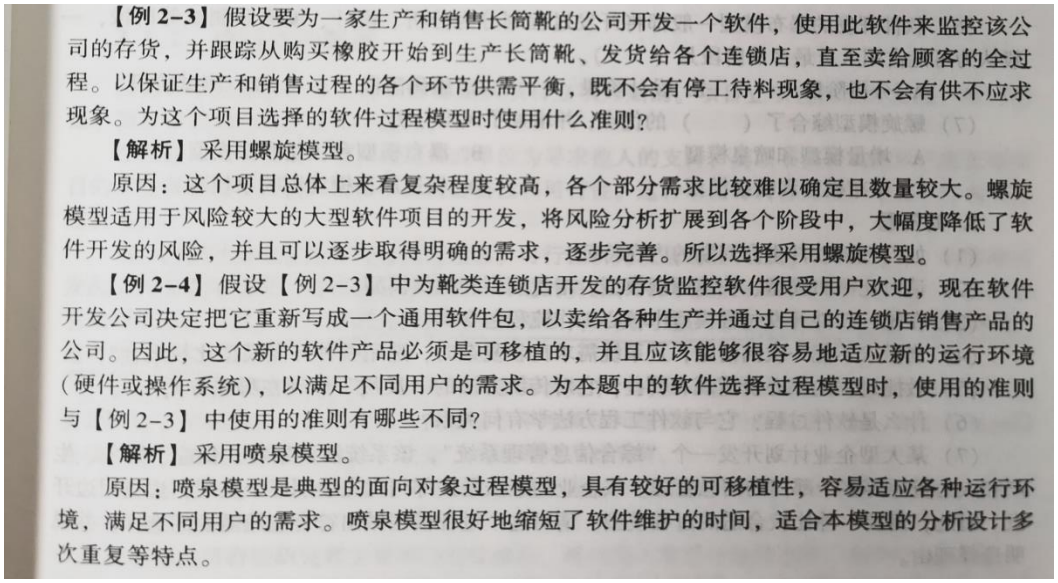
由于考虑问题的角度不同，答案也许不是唯一的。此题还可以考虑。采用敏捷模型。

1）与客户有关的风险很多，其要求人们能够及时与客户进行沟通，充分了解客户的风险，敏捷模型有着经常交付可工作软件的原则，这一点可以充分满足客户规避风险的需要；针对这一点，瀑布模型和基于组件的开发模型都不符合。

2）技术含量很高，这一特点让人们考虑到了时间成本和人力成本，从这两点来说，不能采用增量模型、喷泉模型和统一软件开发过程模型，同时这对技术人员提出了很高的要求，快速原型模型显然不适用，最终剩下了螺旋模型和敏捷模型。

3）两者非常相似，但是敏捷模型更加注重软件开发工程中各种变化的必然性，通过与客户的沟通和团队的合作来更好地适应市场需求，螺旋模型更强调软件的整体性和风险的规避，但是这种模型的控制和管理较为复杂，可操作性不强，对项目管理人员的要求较高。

所以选择敏捷模型作为软件过程模型。



【例 2-3】假设要为一家生产和销售长筒靴的公司开发一个软件，使用此软件来监控该公司的存货，并跟踪从购买橡胶开始到生产长筒靴、发货给各个连锁店，直至卖给顾客的全过程。以保证生产和销售过程的各个环节供需平衡，既不会有停工待料现象，也不会有供不应求现象。为这个项目选择的软件过程模型时使用什么准则？

【解析】采用螺旋模型。

原因：这个项目总体上来看复杂程度较高，各个部分需求比较难以确定且数量较大。螺旋模型适用于风险较大的大型软件项目的开发，将风险分析扩展到各个阶段中，大幅度降低了软件开发的风险，并且可以逐步取得明确的需求，逐步完善。所以选择采用螺旋模型。

【例 2-4】假设【例 2-3】中为靴类连锁店开发的存货监控软件很受用户欢迎，现在软件开发公司决定把它重新写成一个通用软件包，以卖给各种生产并通过自己的连锁店销售产品的公司。因此，这个新的软件产品必须是可移植的，并且应该能够很容易地适应新的运行环境（硬件或操作系统），以满足不同用户的需求。为本题中的软件选择过程模型时，使用的准则与【例 2-3】中使用的准则有哪些不同？

【解析】采用喷泉模型。

原因：喷泉模型是典型的面向对象过程模型，具有较好的可移植性，容易适应各种运行环境，满足不同用户的需求。喷泉模型很好地缩短了软件维护的时间，适合本模型的分析设计多次重复等特点。

大模型对软件工程有什么影响

1. **提升生产效率**：大模型通过智能化代码生成、代码补全和问答等能力，显著减少了手动调试和错误修复的时间，从而提高了软件开发的生产效率。

2. **改善软件质量**：大模型能够快速进行代码验证和测试，及时发现并解决潜在问题，如代码缺陷、代码异常、代码安全风险等，从而帮助开发人员编写出更高质量的代码，降低软件发布后的故障率，提升软件的稳定性和性能。

3. **加速软件创新迭代**：通过提升开发效率和改善软件产品质量，企业能够更快地推出创新产品，抢占市场先机，从而在激烈的市场竞争中脱颖而出。

4. **软件产品形态的影响**：大模型推动软件能力更加智能化，从多模态输入输出、智能识别、数据处理、决策实施等维度，全面提高软件的智能化程度。

5. **软件技术层面的变革**：传统软件技术将在大模型推动下被重新定义。从软件实现层面，大模型将以多种方式与传统软件深入融合，或通过嵌入式方式，或通过知识库及 RAG 方式，或采用单智能体或多智能体的方式等，提升软件的理解、生成、决策等维度的能力。

6. **软件与行业场景更深度融合**：大模型提升了软件的定制化能力，尤其是各行业大模型在行业专有数据集上进行了训练和学习，使赋能后的软件能够更有效地理解场景需求，更高效地运用行业知识，更灵活地处理场景问题，并且催生更多应用场景。

7. **对软件产业的影响**：大模型为软件从业者带来了转型的挑战，需要不断学习新的技能和工具，以适应不断变化的技术环境。同时，大模型的出现可能会减少初级程序员岗位，简单和重复性的编码任务将主要通过生成式的方式完成。

8. **新机遇**：大模型技术发展为软件工程带来了新的机遇，包括基于大模型的企业软件开发方式变革，大模型对复杂软件系统开发的支持，以及基于大模型的新型智能化软件的构造方法与运行支撑。

这些影响表明，大模型正在深刻地影响着软件开发的各个环节，从提高开发效率到改善软件质量，再到加速创新迭代，以及推动软件产业结构性变化。随着 AI 技术的不断进步，预计大模型将在未来的软件开发中发挥更加关键的作用。

一个企业的开发过程

通常包括以下几个主要阶段：需求分析、设计、编码与测试、集成与系统测试、部署与维护。这些阶段共同确保软件的质量和稳定性，同时满足客户的需求。

需求分析：在软件开发的第一步，开发团队与客户沟通，详细了解客户的需求和期望。通过交流，开发团队明确软件的功能和特性，并制定相应的计划。

设计阶段：在设计阶段，团队根据客户需求设计软件的架构和界面，确保软件的可扩展性和用户体验。

编码与测试：开发团队根据设计文档编写代码，并进行单元测试，确保代码的正确性和稳定性。

集成与系统测试：在编码阶段完成后，开发团队将各个模块的代码整合进行集成测试，并对整个软件系统进行系统测试，以确保各个模块之间的协同工作和系统的稳定性。

部署与维护：通过集成和系统测试后，软件被部署到生产环境中。开发团队确保软件的稳定性和安全性，并在上线后继续监控和维护，以确保软件的正常运行和及时修复可能出现的问题。

具体流程细节

1.**立项阶段**：包括市场调研、用户调研和竞争对手分析，明确产品的定位和目标。

2.需求阶段：产品经理与设计师沟通，制定产品设计方案，包括核心功能、界面设计和用户体验。

3.技术选型：根据需求选择合适的技术、组件和框架。

4.写 demo：制作测试用例，验证功能。

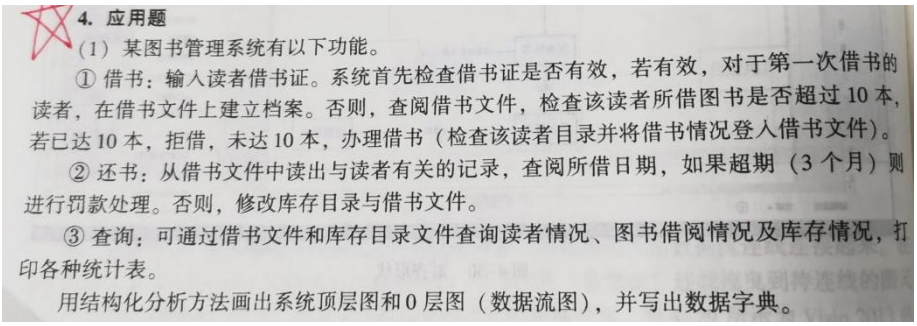
5.代码提交/代码评审：将代码提交到远程仓库，同事之间进行项目检查。

6.服务器部署：将项目放到服务器或容器环境上。

7.发布：选择性地发布不同机器上的容器或服务。

八．结构化分析

1.



4. 应用题

（1）某图书管理系统有以下功能。

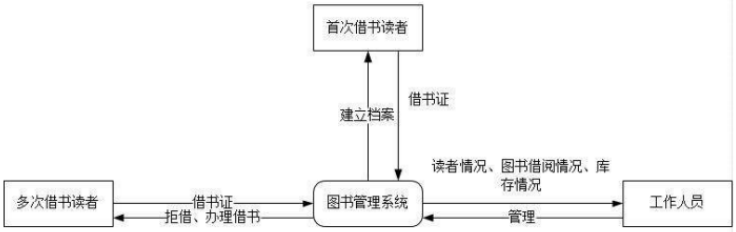
① 借书：输入读者借书证。系统首先检查借书证是否有效，若有效，对于第一次借书的读者，在借书文件上建立档案。否则，查阅借书文件，检查该读者所借图书是否超过 10 本，若已达 10 本，拒借，未达 10 本，办理借书（检查该读者目录并将借书情况登入借书文件）。

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。

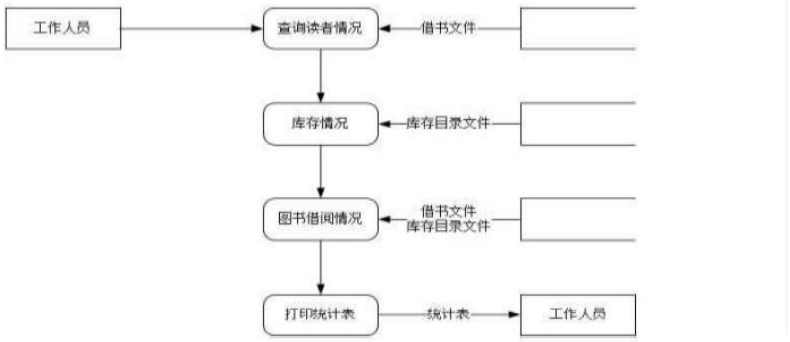
用结构化分析方法画出系统顶层图和 0 层图（数据流图），并写出数据字典。

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。



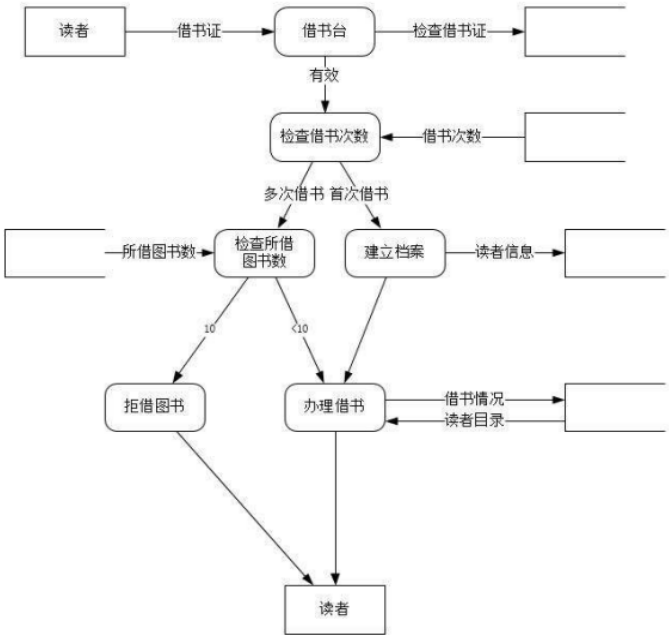
图C-1 系统顶层图

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。



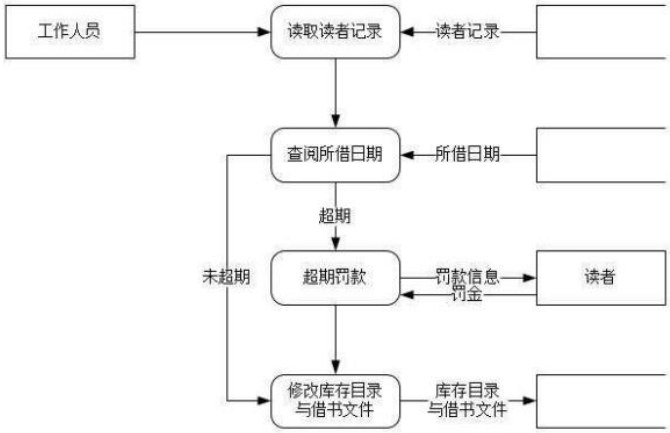
图C-2 查询0层图

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

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

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}

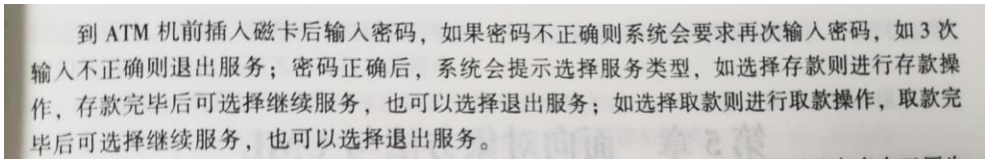
图书编号: 6{数字}6

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

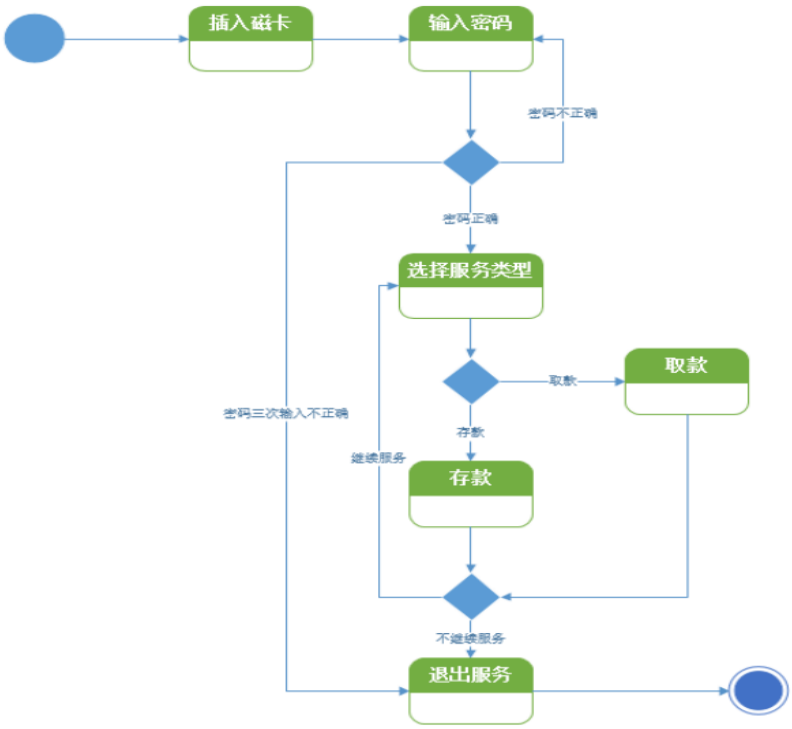
C．还书 0 层图：
工作人员 ={姓名+工作人员代号}
读者记录 ={读者+图书名+图书编号+所借日期}
读者 ={姓名+学号+班号}
罚款信息 ={读者+图书名+图书编号+罚金数额}
库存目录 ={图书名+图书编号+库存数量}
借书文件 ={读者+图书名+图书编号+所借日期}
姓名: 2{汉字}4 学号: 8{数字}8
班号: 4{数字}4
工作人员代号: 4{数字}4
图书名: {汉字}
图书编号: 6{数字}6
所借日期: 8{数字}8
罚金数额: 1{数字}2

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.根据以下描述画出相应的状态转换图


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。

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。

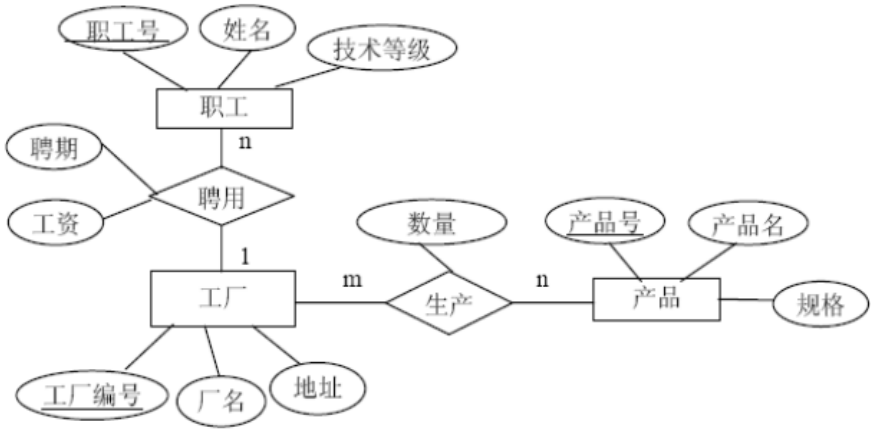


图C-6 状态转换图

3.

（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、厂名和地址，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。请画出E－R图。

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

九．面向对象分析

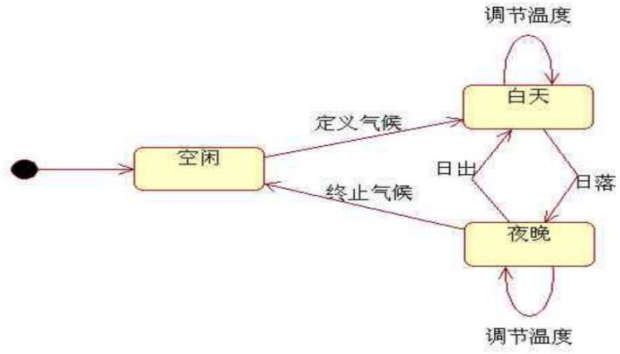
1.

4. 应用题 考试题型 AIM

（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。

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。

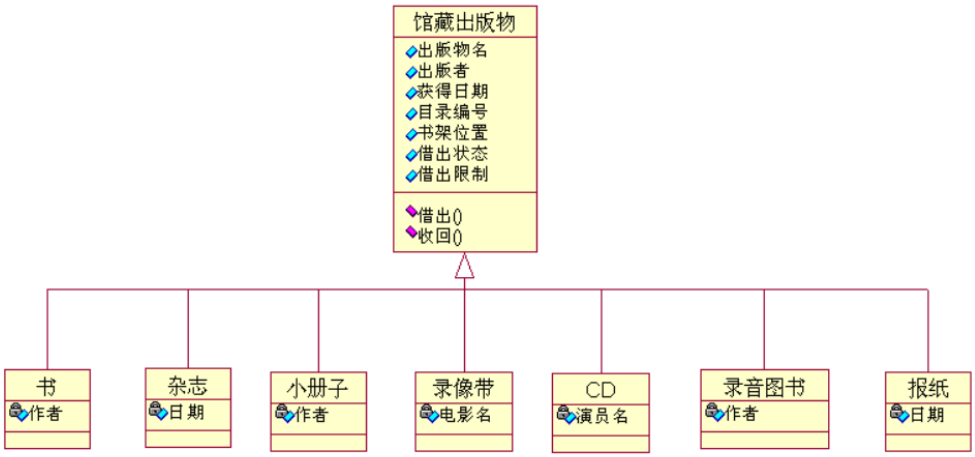


2.

（2）一家图书馆藏有书、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。

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。



3.

（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。

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。

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：

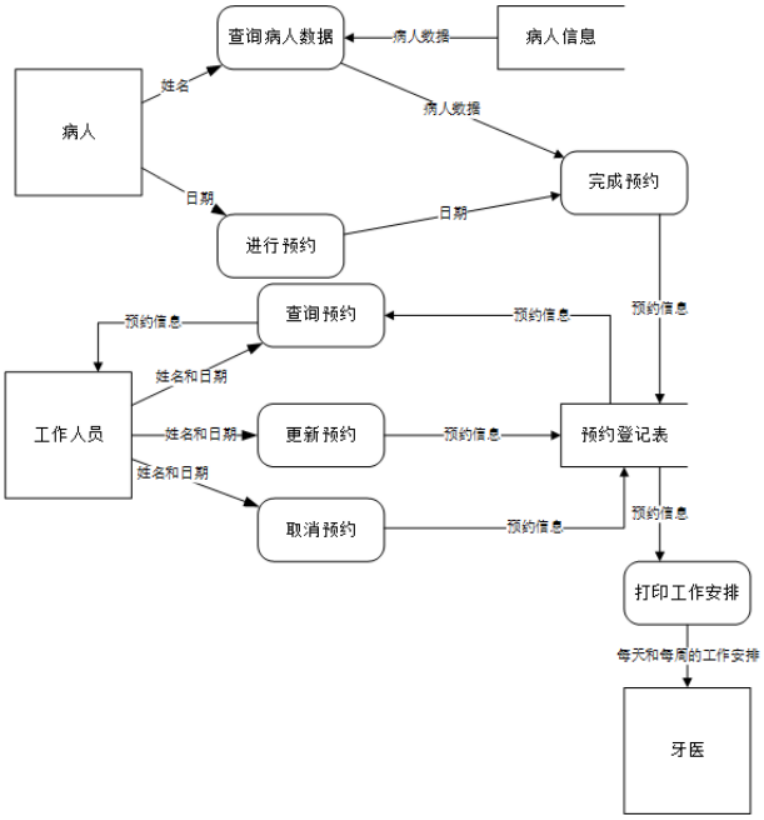
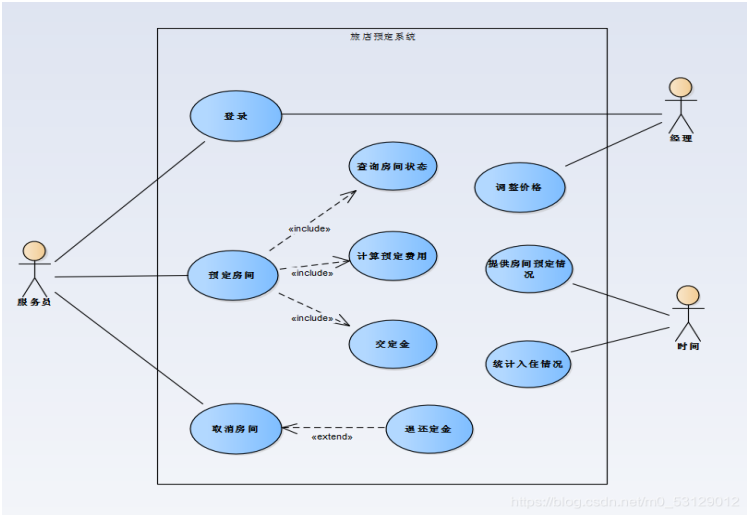


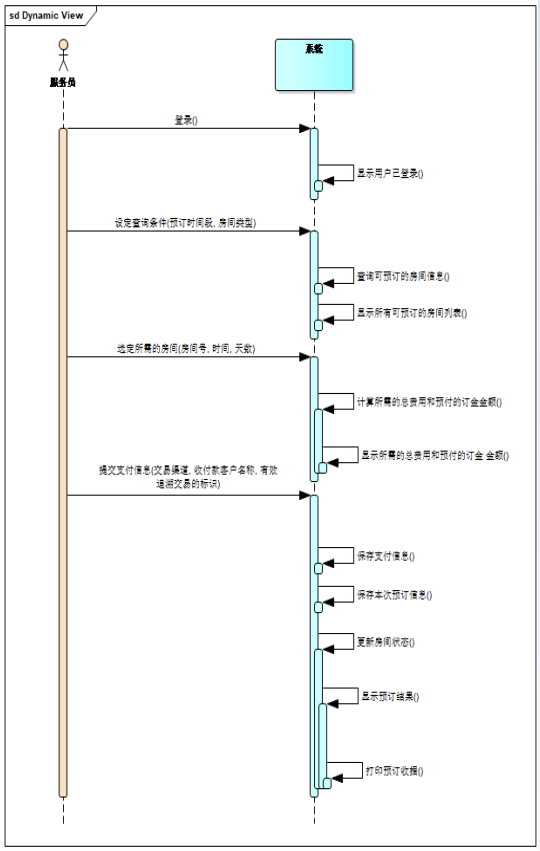
图 C-15 牙科诊所管理系统的功能模型

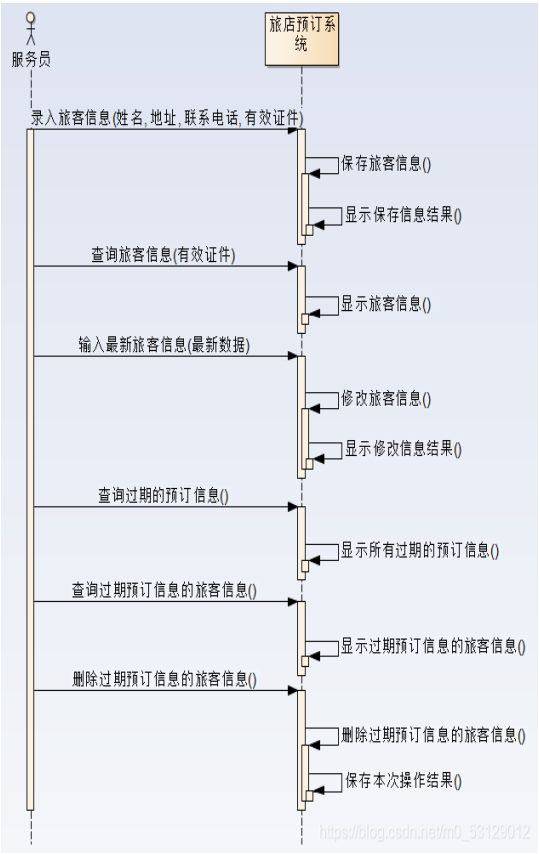
4. 某旅店可对外开放 50 个双人间和 2 0 个单人间，房间费用视情况按季节调整，但周一到周五提供半价（周末全价）折扣。旅客可以直接入住房间(如果有空房)，也可提前预订；入住和预订都需要登记个人信息。旅客提前预订房间时，需提交一定的订金；入住时间 24 小时之外的旅客可以取消预订，并退回所有订金，24 小时以内则不退还订金。



用例图：

顺序图：预订房间用例描述/管理旅客信息





十．结构化设计

1.

9.8 实验：利用 Visio 绘制“‘墨韵’读书会书籍共享平台”的结构图

在 4.5 节中，绘制出了书籍共享平台的 3 层数据流图，本节将在底层数据流图的基础上，运用面向数据流的方法对该系统进行设计，得到书籍共享平台的结构图。设计的过程可以分为以下 3 步。

1）区分数据流中的变换型数据流和事务型数据流。

2）对于变换型数据流，识别变换型数据流中的输入流、变换流和输出流；对于事物型数据流，识别事务中心和数据接收通路。

3）设计软件系统结构，绘制系统结构图。

在本章的理论部分，已了解了变换型数据流和事务型数据流的相关概念。变换型数据流强调对输入信息流的加工与处理，而事务型数据流强调根据输入信息流的特征，在若干处理方式中选择一个合适的进行加工。以书籍共享平台的发表日志数据流图和查看图书数据流图为例，可以分辨出，发表日志数据流图中的数据流为变换型数据流，而书籍信息查看中的数据流为事务型数据流，其中“查询书籍”为其事务中心。

下面分别以“发表日志”和“查看图书”的设计为例，介绍利用数据流图绘制系统结构

图的方法。

对于变换型数据流，要重点识别变换型数据流中的输入流、变换流和输出流。输入流对应的输入控制模块完成数据输入及与数据输入相关的操作。变换流对应的中心控制模块包括所有对数据的核心操作。输出流对应的输出控制模块协调输出信息的产生过程。经分析，“发表日志”有边界的数据流图如图 9-31 所示。

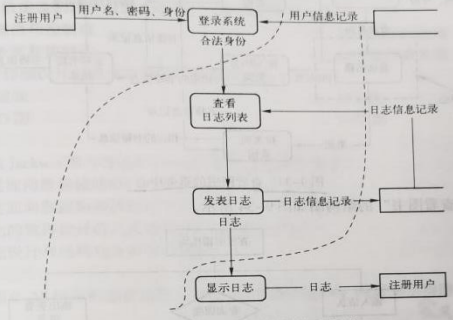


图 9-31　发表日志具有边界的数据流图

图 9-31 中用虚线标识了 3 个区域。自左向右，这 3 个区域分别为输入部分、数据处理部分和输出部分。可见，“登录系统”流程属于输入部分。模块内部对数据的变换处理包括“查看日志列表”和“发表日志”。“显示日志”为输出部分。

根据“发表日志”具有边界的数据流图，可以将其映射为结构图，如图 9-32 所示。

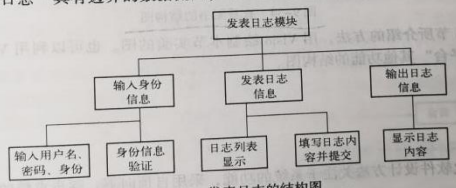


图 9-32　发表日志的结构图

对于事务型数据流图，可以通过事务分析得到对应的结构图。在事务分析的过程中，重点在于对事务中心的识别。在“查看图书”的数据流图中，“查询书籍”为其事务中心，如图 9-33 所示。

对于每一个事务中心，外界必须输入相应的信息来帮助系统判断选择何种事务处理方式。例如，在“查看图书”中，必须输入有关查询类型选择的信息，来使系统判断在 3 种查询方式（按书名查询、按 ISBN 查询和按类别查询）中选择哪一种。这样，输入部分不仅包括与登录信息相关的数据，还包括查询类型信息。

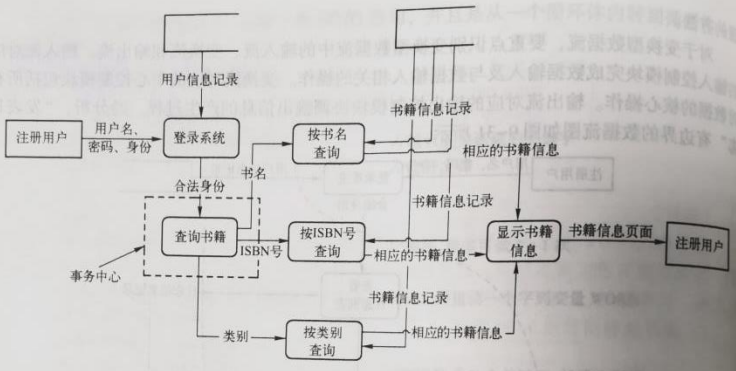


图 9-33　查看图书的事务中心

经分析，“查看图书”的结构图如图 9-34 所示。

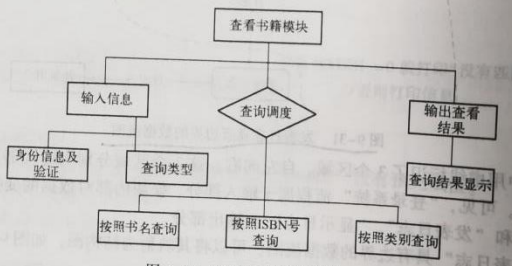


图 9-34　查看图书的结构图

可参照 4.5 节所介绍的方法，用 Visio 绘制本节实验的图。也可以利用 Visio 绘制“‘墨韵’书籍共享平台”其他功能的结构图。

2.

4. 应用题

(1) 请将图 9-35 所示的数据流图（具有边界的数据流图）映射成结构图。

定义⇒画出数据流图
⇒功能结构图
(设计十分析)

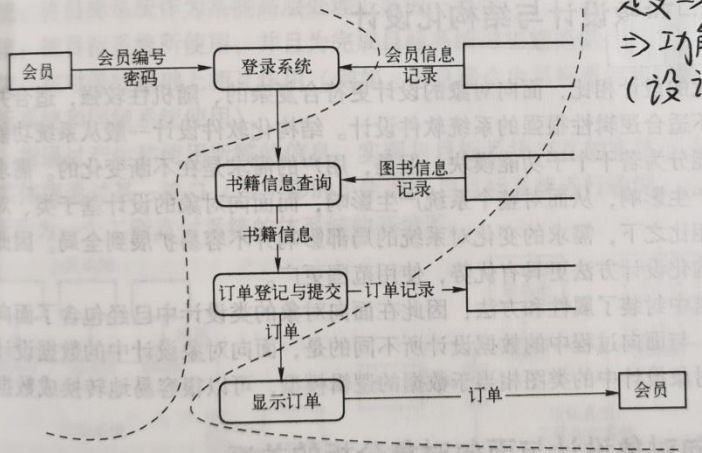
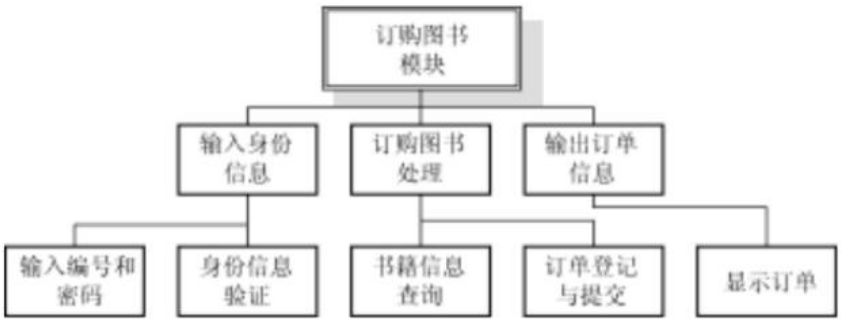
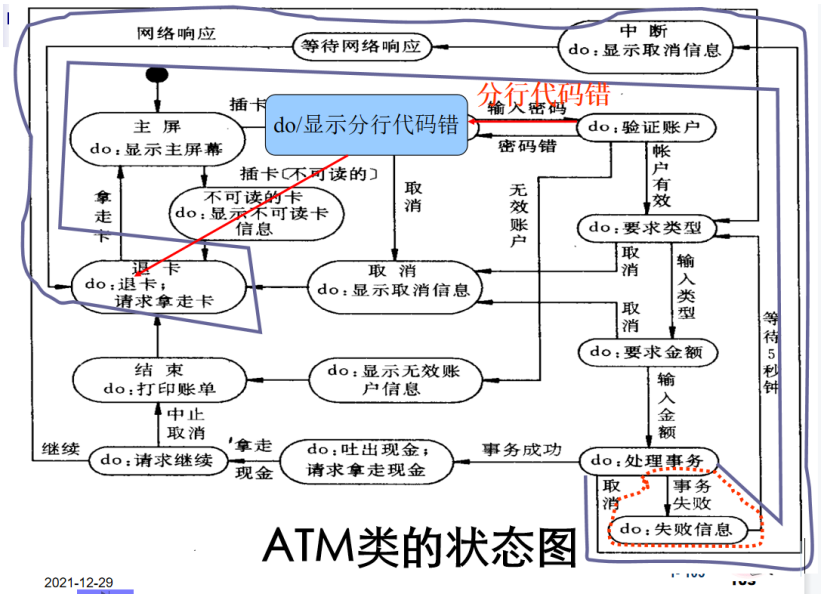
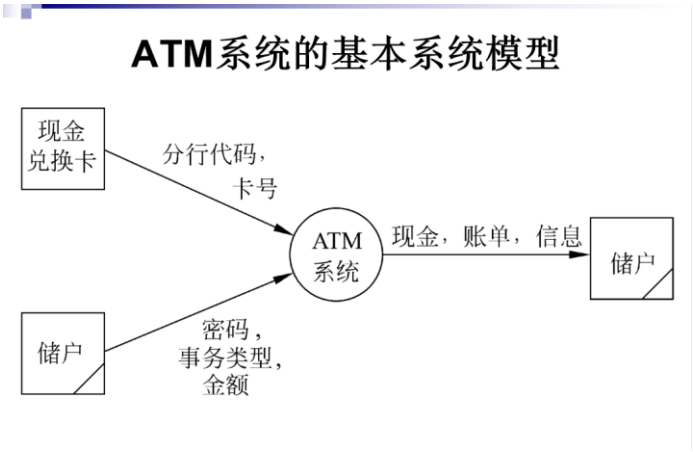


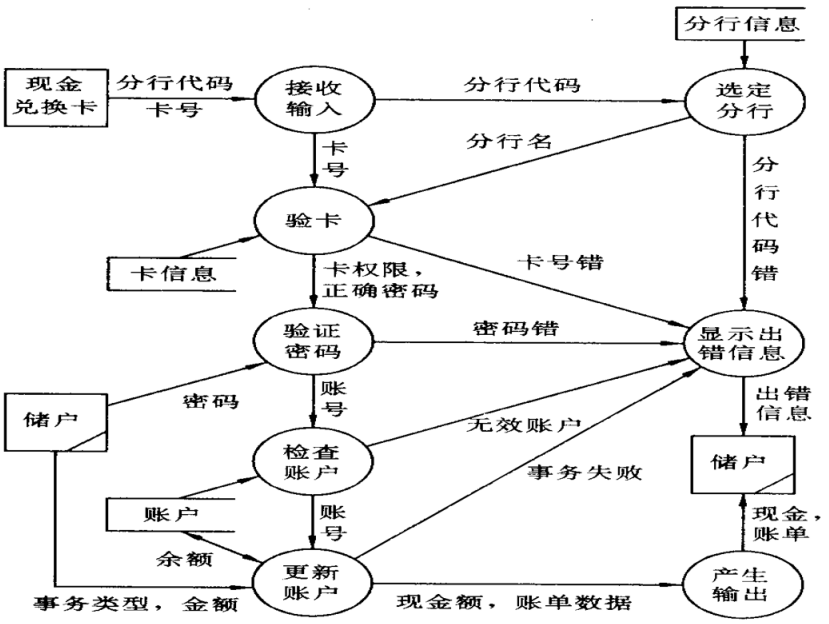
图 9-35　“订购图书”具有边界的数据流图



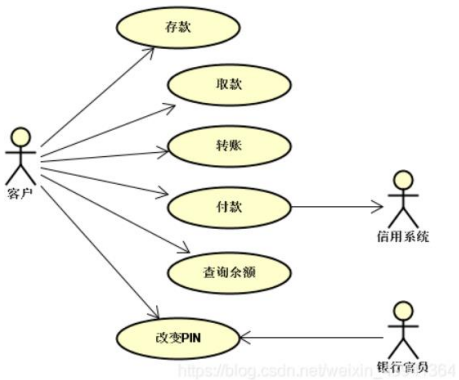
ATM


网络响应
等待网络响应
中 断
do：显示取消信息
分行代码错
插卡
主 屏
do：显示主屏幕
do/显示分行代码错
输入密码
do：验证账户
密码错
帐户有效
插卡〔不可读的〕
不可读的卡
do：显示不可读卡信息
拿走卡
取消
无效账户
do：要求类型
退 卡
do：退卡；请求拿走卡
取 消
do：显示取消信息
取消
输入类型
等待5秒钟
do：要求金额
结 束
do：打印账单
do：显示无效账户信息
输入金额
中止取消
继续
do：请求继续
拿走现金
do：吐出现金；请求拿走现金
事务成功
do：处理事务
取消
事务失败
do：失败信息
ATM类的状态图
2021-12-29


ATM系统的基本系统模型
现金兑换卡
分行代码，
卡号
ATM系统
现金，账单，信息
储户
储户
密码，
事务类型，
金额


分行信息
现金兑换卡
分行代码
卡号
接收输入
分行代码
选定分行
卡号
分行名
验卡
分行代码错
卡信息
卡权限，正确密码
卡号错
验证密码
密码错
显示出错信息
储户
密码
账号
出错信息
检查账户
无效账户
储户
事务失败
账户
账号
现金，账单
余额
更新账户
产生输出
事务类型，金额
现金额，账单数据

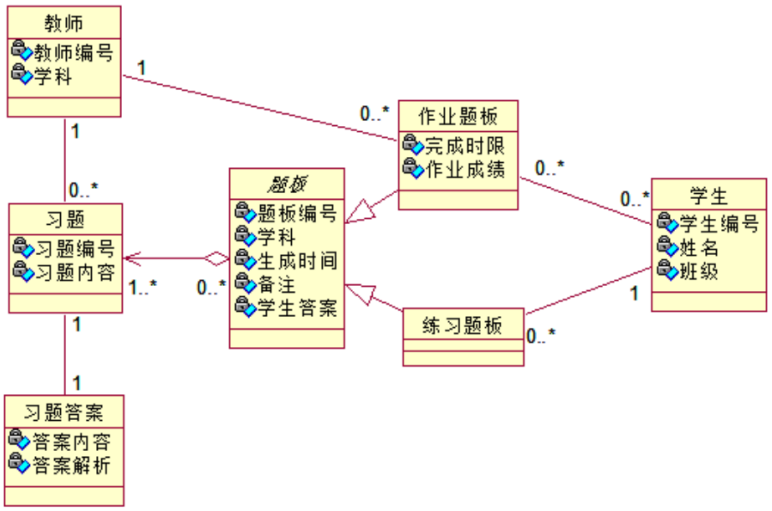
用户用例图



4. 应用题

（1）在一个习题库下，各科教师可以在系统中编写习题及标准答案，并将编写的习题和答案加入题库中，或者从题库中选取一组习题组成向学生布置的作业，并在适当的时间公布答案。学生可以在系统中完成作业，也可以从题库中选择更多的习题练习。教师可以通过系统检查学生的作业，学生可以在教师公布答案后对自己的回答进行核对。阅读这一情境，分析出该系统所包括的实体类并适当添加属性，绘制出类图。

（1）根据题意可以抽象出系统的实体类包括教师、习题、习题答案、作业题板、练习题板以及学生。作业题板与练习题板可以泛化为题板抽象类。根据描述对各个类添加适当的属性。在各个类之间按照文字描述建立关联关系。教师与习题之间构成 1 对多关系，教师与作业题板构成 1 对多关系，习题与习题答案之间构成 1 对 1 关系，习题与题板之间构成多对多关系且这一关联关系可以描述为聚合关系，学生与作业题板构成多对多关系，学生与练习题板构成 1 对多关系。类图可参考下图。类图如 C-21 图所示。



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。

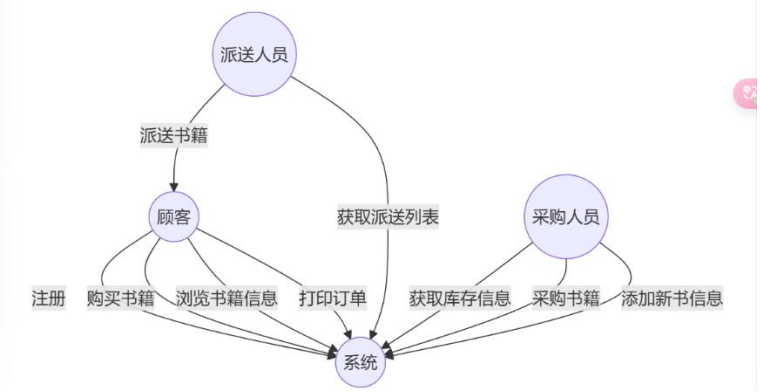
现采用面向对象软件工程方法，使用UML建模技术，完成该软件系统分析与建模，请分别给出：

（1）业务用例和业务序列图；

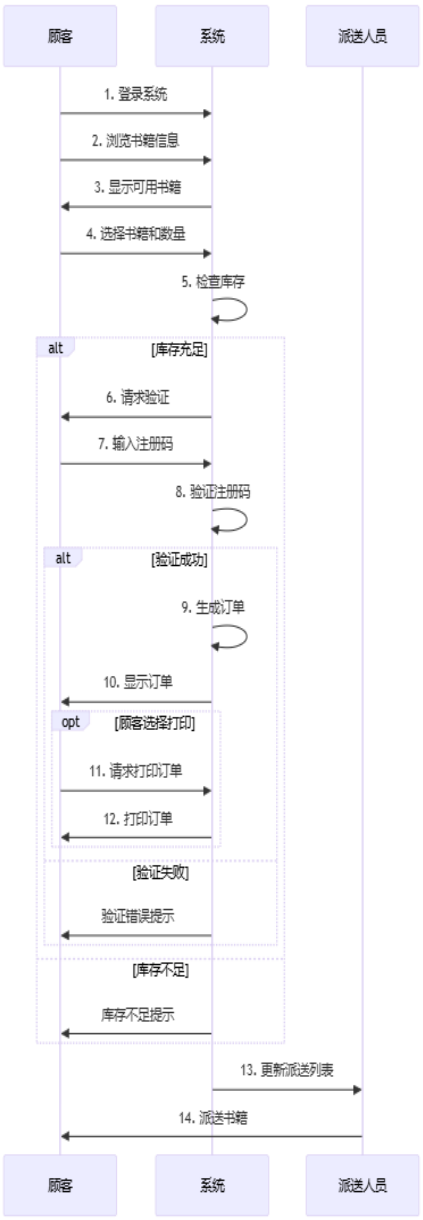
（2）系统用例图，并给出顾客“购买书籍”用例的用例规约；

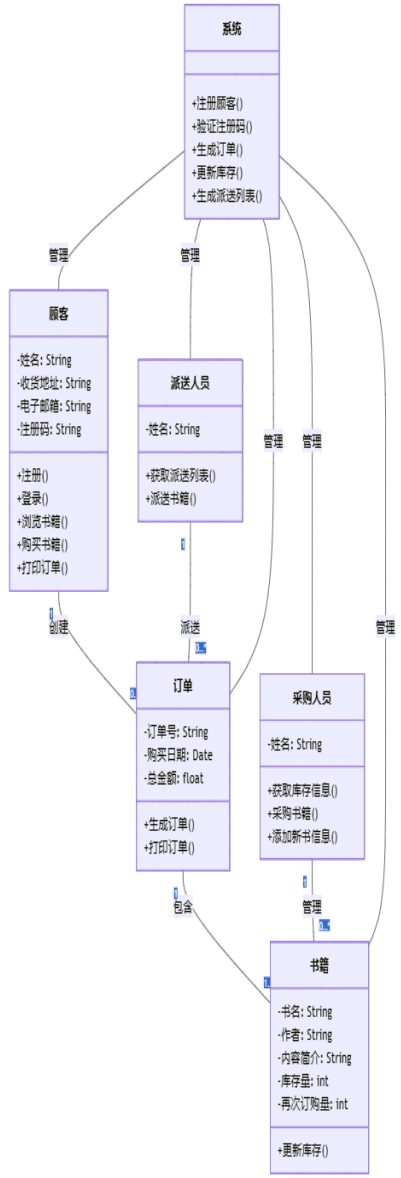
（3）系统类图。

业务用例图：

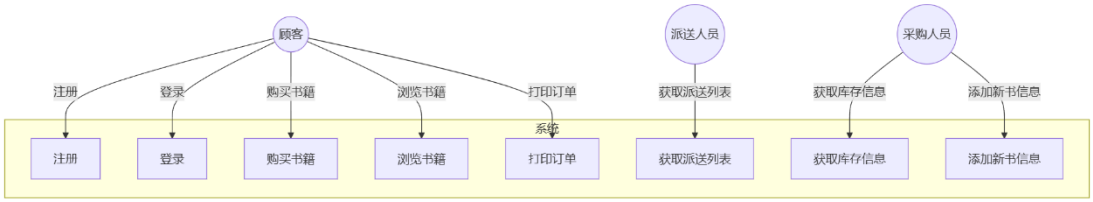


业务序列图（以购买书籍为例）与系统类图





（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

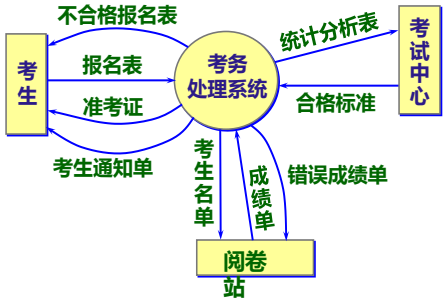
b. 顾客选择要购买的书籍和数量

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

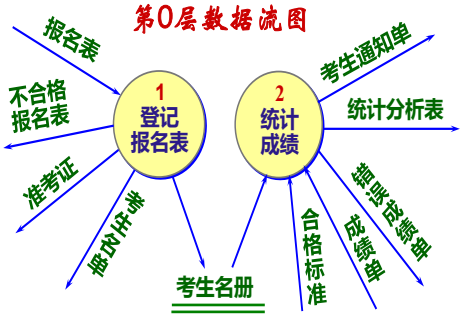
实例考务处理系统的功能

(1) 对考生送来的报名单进行检查;
(2) 对合格的报名单编好准考证号后将准考证送给考生，并将汇总后的考生名单送给阅卷站;
(3) 对阅卷站送来的成绩单进行检查，并根据考试中心制定的合格标准审定合格者;
(4) 制作考生通知单（含成绩及合格/不合格标志）送给考生;
(5) 按地区进行成绩分类统计和试题难度分析，产生统计分析表。

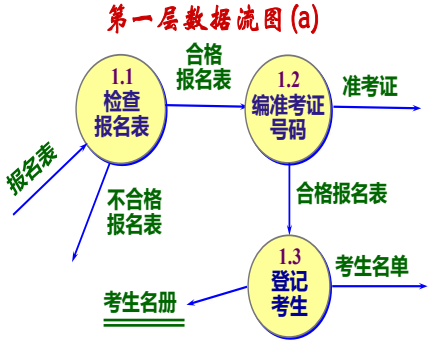
顶层数据流图



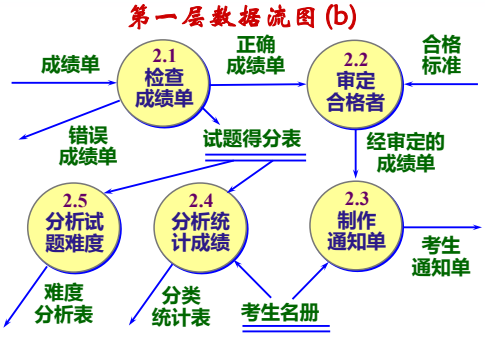
第0层数据流图



第一层数据流图 (a)



第一层数据流图 (b)



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，喷泉模型的各阶段均采用了“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

●对软件开发的风险进行预防和控制

●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

●有利于软件开发的管理和控制

*包含：如果 B 是 A 的某项子功能，并且建模者确切地知道在 A 所对应的动作序列中何时将调用 B，则称用例 A 包含用例 B。

*扩展：如果 A 与 B 相似，但 A 的功能较 B 多，A 的动作序列是通过 B 的动作序列中的某些执行点上插入附加动作序列而构成的，则称用例 A 扩展用例 B。

*继承（泛化）：如果 A 与 B 相似，但 A 的动作序列是通过改写 B 的部分动作或者扩展 B 的部分动作而获得的，则称用例 A 继承用例 B。

用例图

关系类型	说明	表示符号
关联	参与者与用例之间的关系	——
泛化	参与者之间或用例之间的关系	——▷
包含	用例之间的关系	- - «includes» ->
扩展	用例之间的关系	- - «extends» ->

扩展关系中基本用例的基本流执行时，扩展用例不一定执行，即扩展用例只有在基本用例满足某种条件的时候才会执行。箭头从子用例指向基用例