

计算机导论

第四章 程序设计导引



河北师范大学软件学院
Software College of Hebei Normal University





1. 程序设计概述
2. 基本的数据结构
3. 程序设计方法
4. 软件生命周期



1. 程序设计概述
2. 基本的数据结构
3. 程序设计方法
4. 软件生命周期

编程的目的



为什么要编程？



编程是为了让计算机帮助我们解决问题。

程序的由来



程序怎么来？



算法——解决问题的方法

高级语言——Python（三、四章）

程序设计的导引



待处理数据不是单纯的数值
程序非常庞大

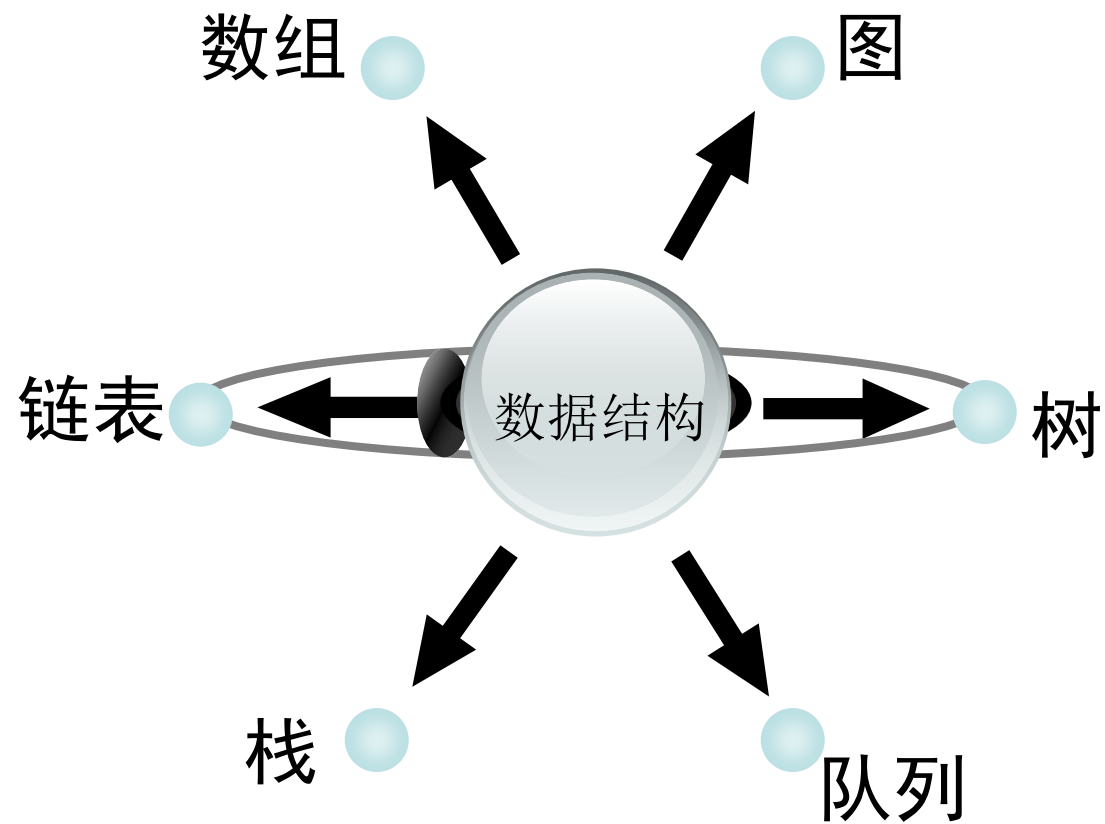


基本数据结构
程序设计方法
软件生命周期



1. 程序设计概述
2. 基本的数据结构
3. 程序设计方法
4. 软件生命周期

数据结构





- 最常用且最简单的一种数据结构。



- 特点：4个“惟一”。

线性结构的特点



数据元素的
非空有限集

- 存在唯一的一个被称作“第一个”的数据元素;
- 存在唯一的一个被称作“最后一个”的数据元素;
- 除第一个之外的数据元素均只有一个前驱;
- 除最后一个之外的数据元素均只有一个后继。

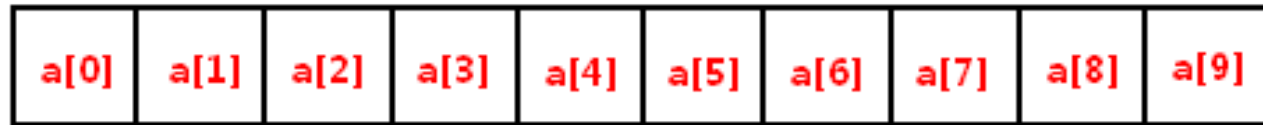
例:

法学系	8523101	→ “第一个” 数据元素
国贸系	8522105	
工商系	8523150	↩ 直接前驱
计算机系	8521088	↪ 直接后继
会计系	8525789	
统计系	8528136	
...	...	
外语系	8523026	→ “最后一个” 数据元素

数组

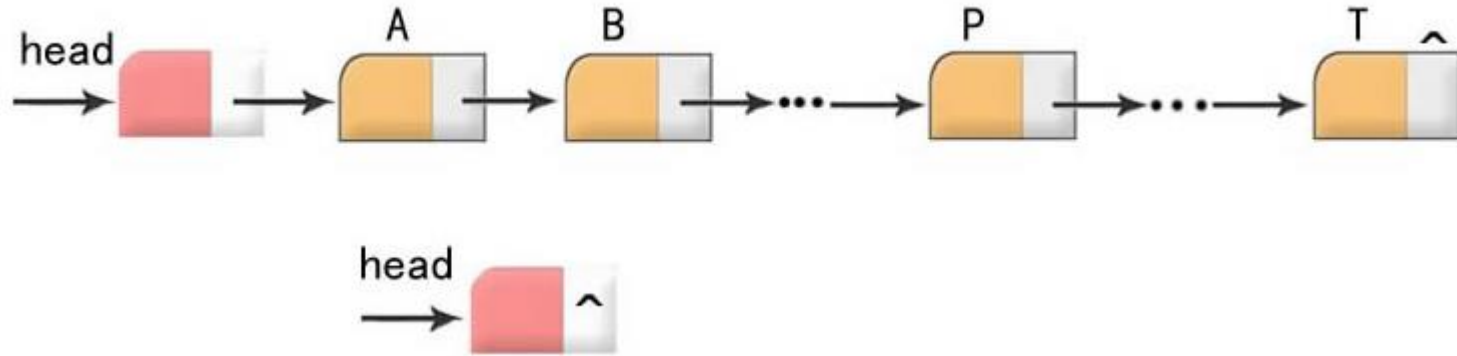


数组a, int型



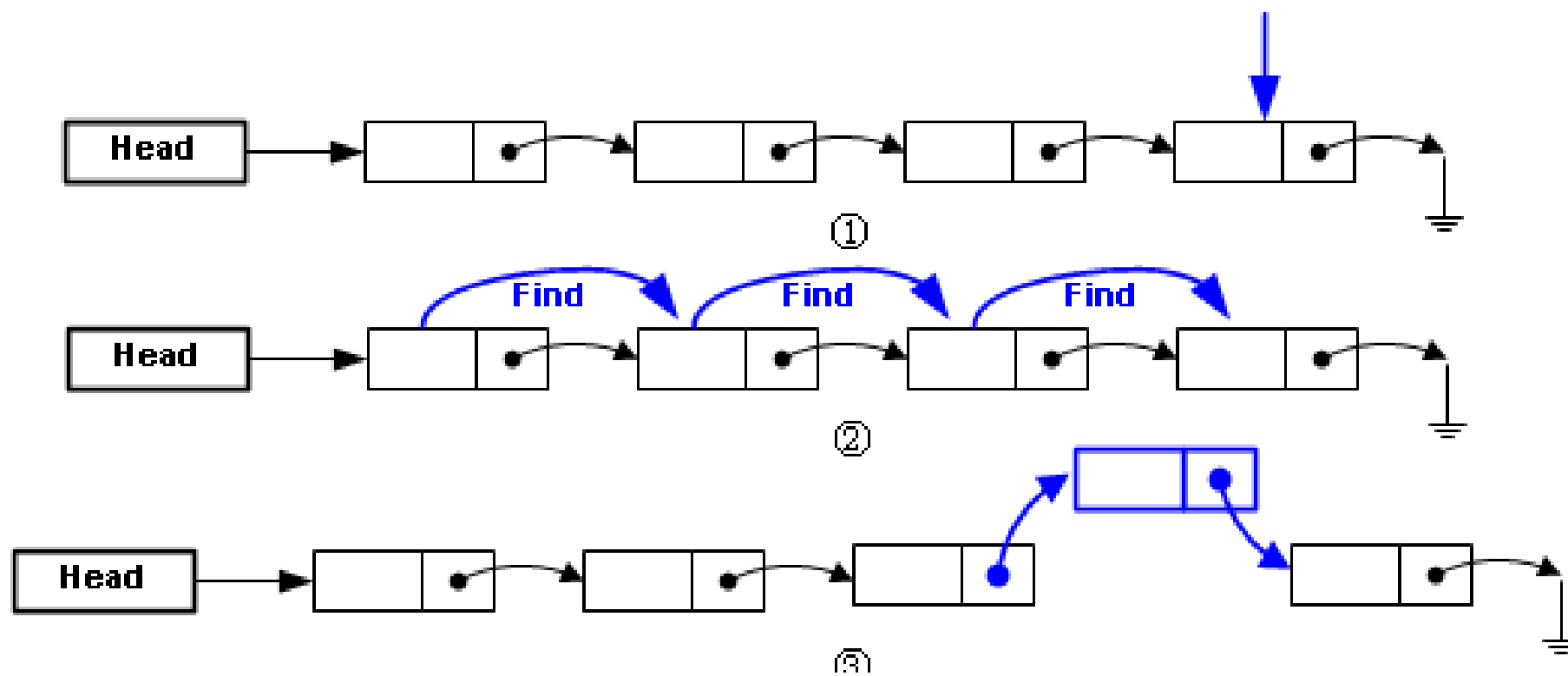
在内存中的地址: 1000 1004 1008 1012 1016 1020 1024 1028 1032 1036 (十进制表示)

- 遍历
- 插入
- 删除



- 遍历
- 插入
- 删除

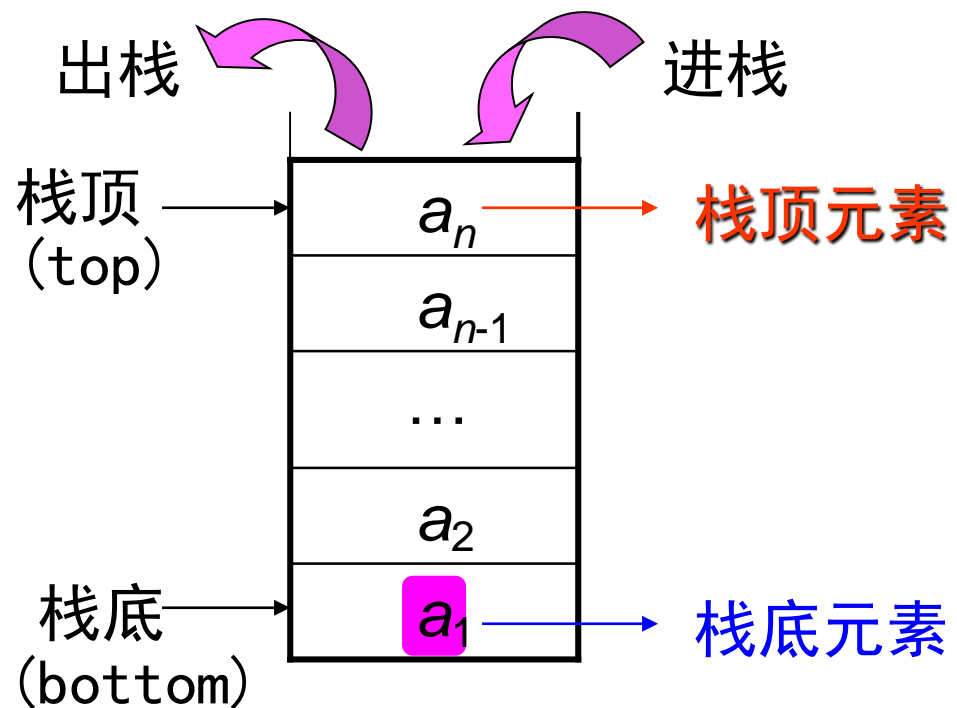
链表插入



栈



栈：线性表 { 限定仅在表尾进行插入或删除操作。
后进先出（LIFO结构）



队列



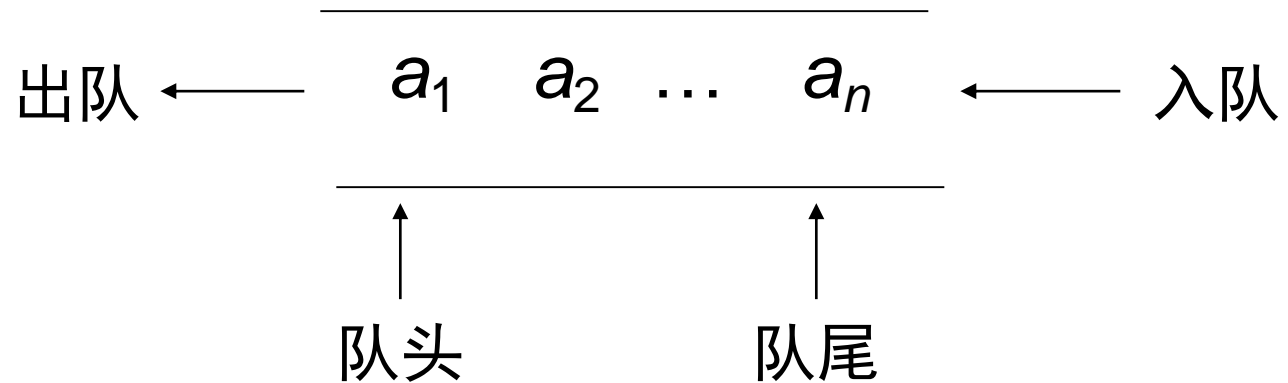
队列：线性表
(queue)

限定在表的一端插入、另一端删除。
先进先出 (FIFO结构)。

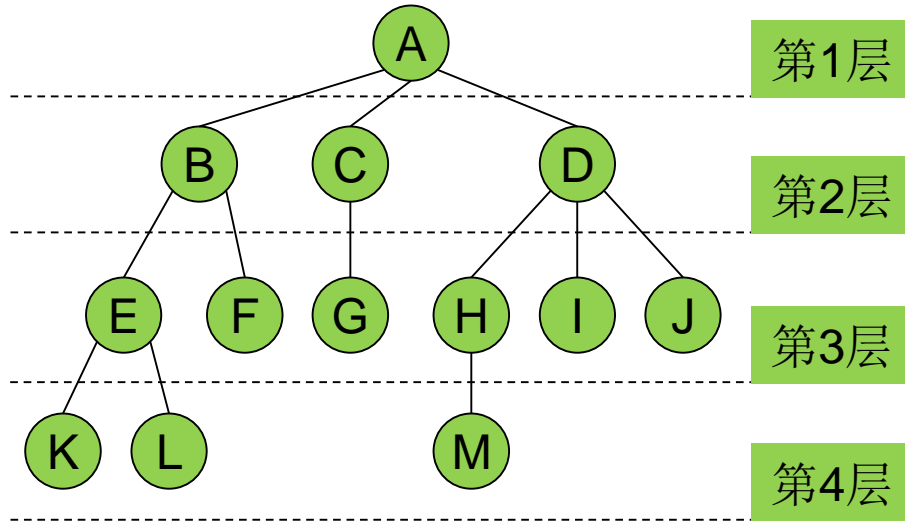
队尾

队头

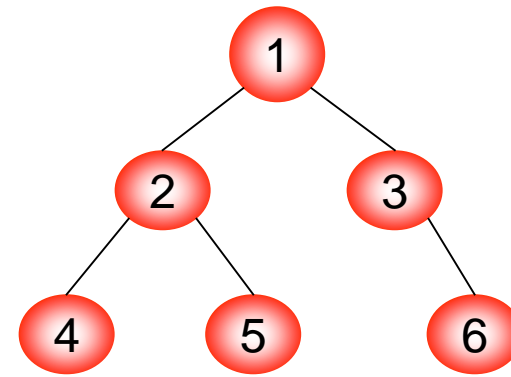
下图是队列的示意图：



树

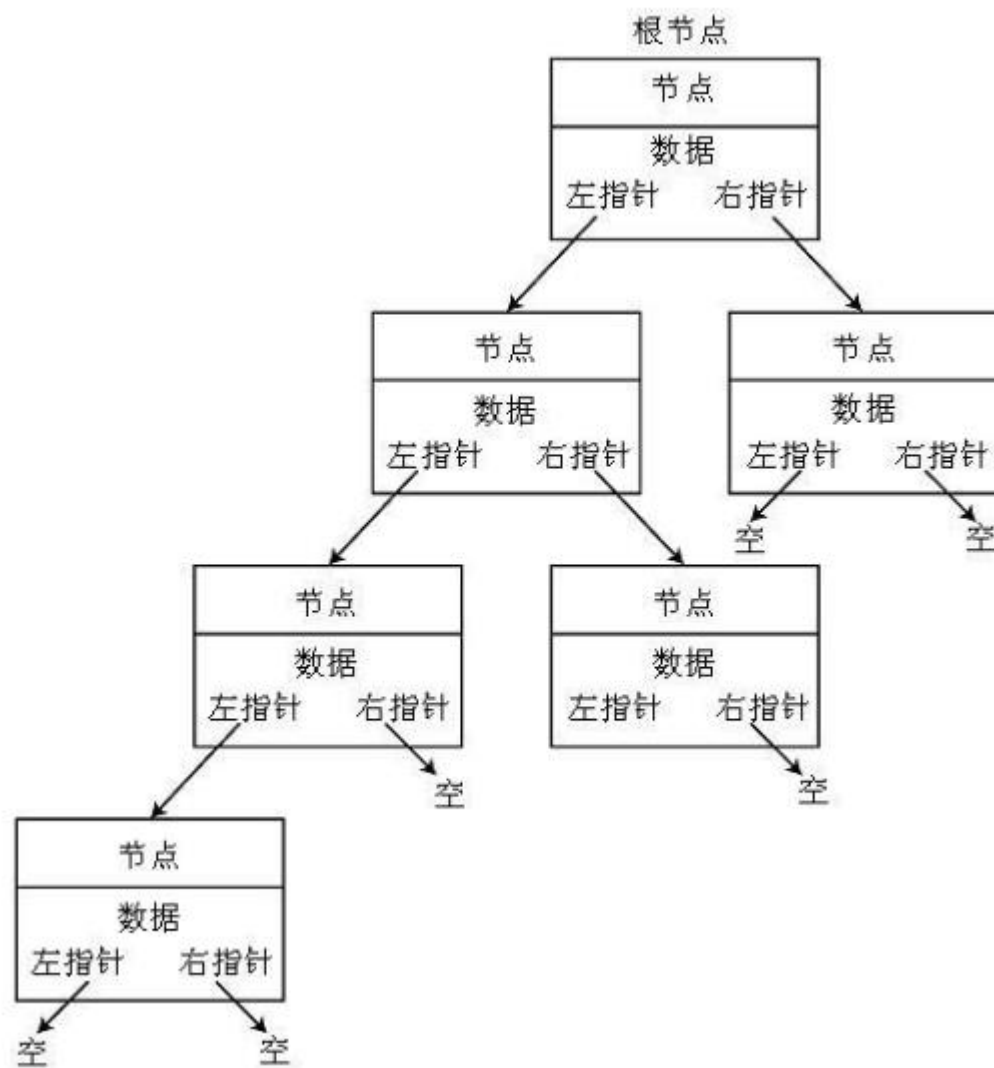


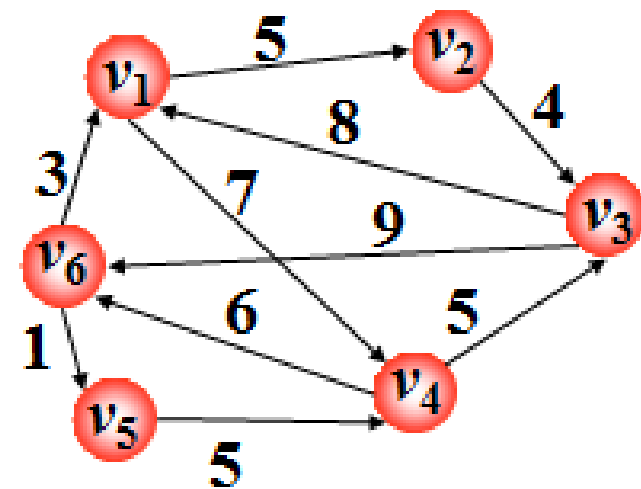
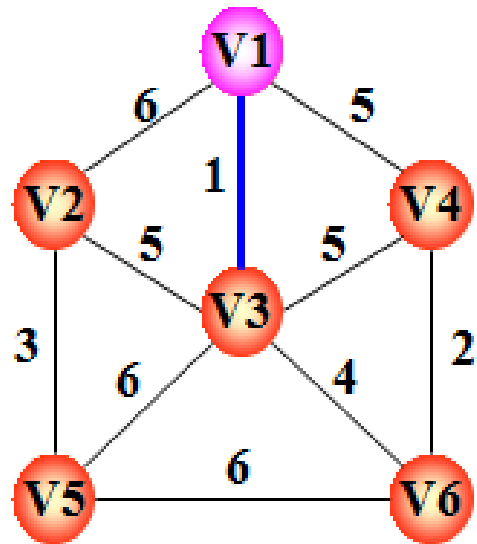
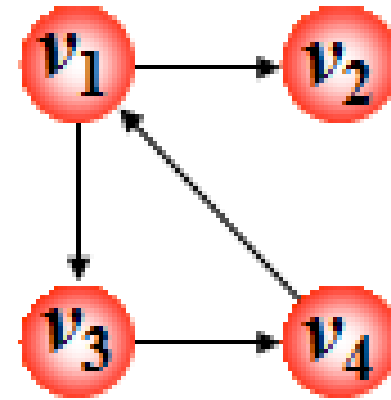
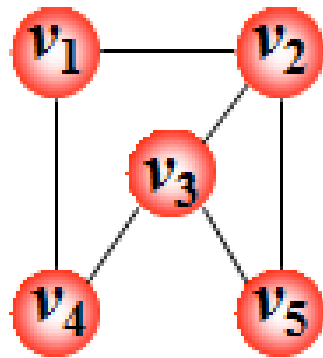
树



二叉树

树的存储

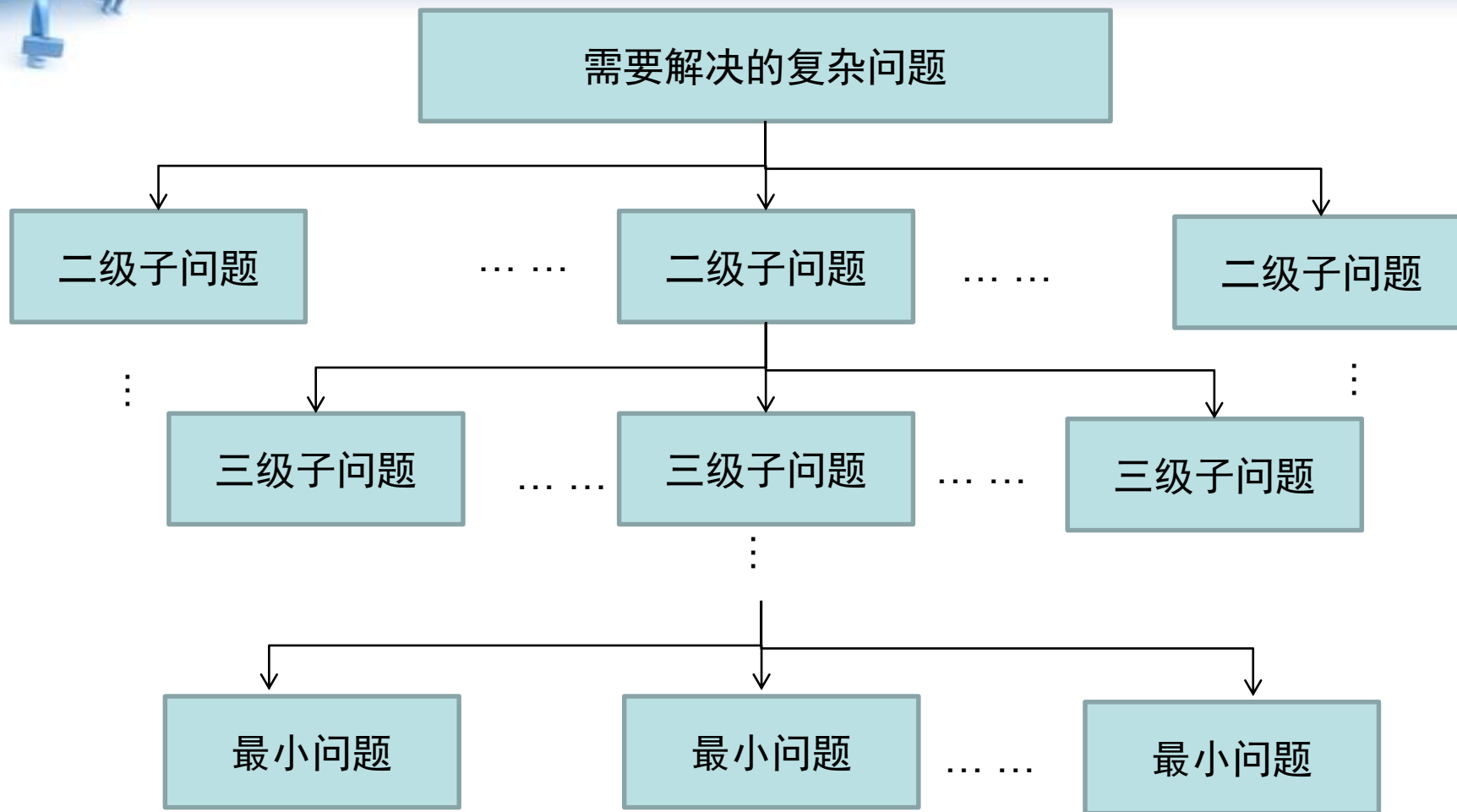






1. 程序设计概述
2. 基本的数据结构
3. 程序设计方法
4. 软件生命周期

研究问题的思路



自上而下的结构示意图

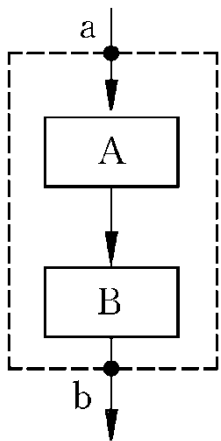


- 模块化方法是一种传统的软件开发方法。该方法通常将待开发软件划分为一些功能相对独立的模块，模块与模块之间定义相应的接口，各个模块可以分别单独开发、调试、运行、测试，最后再将多个模块组合起来，进行整体测试，从而完成整个软件的开发。

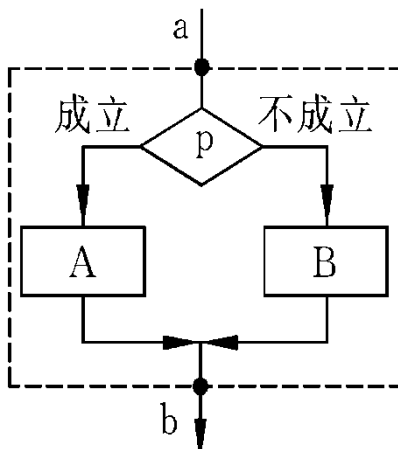


- 采用自上而下、逐步求精的设计方法和单入口、

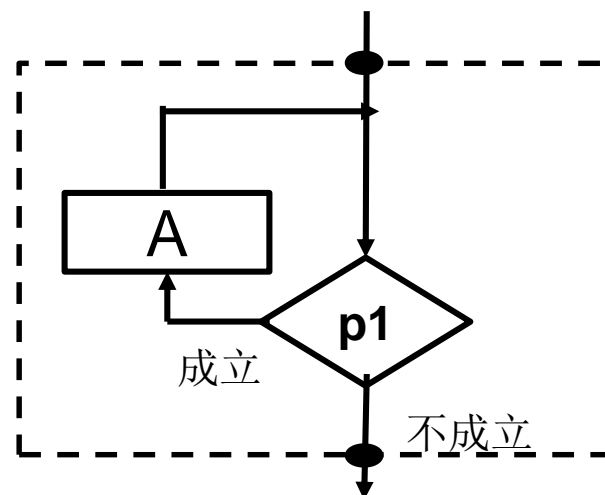
单出口的控制结构



顺序结构



选择结构



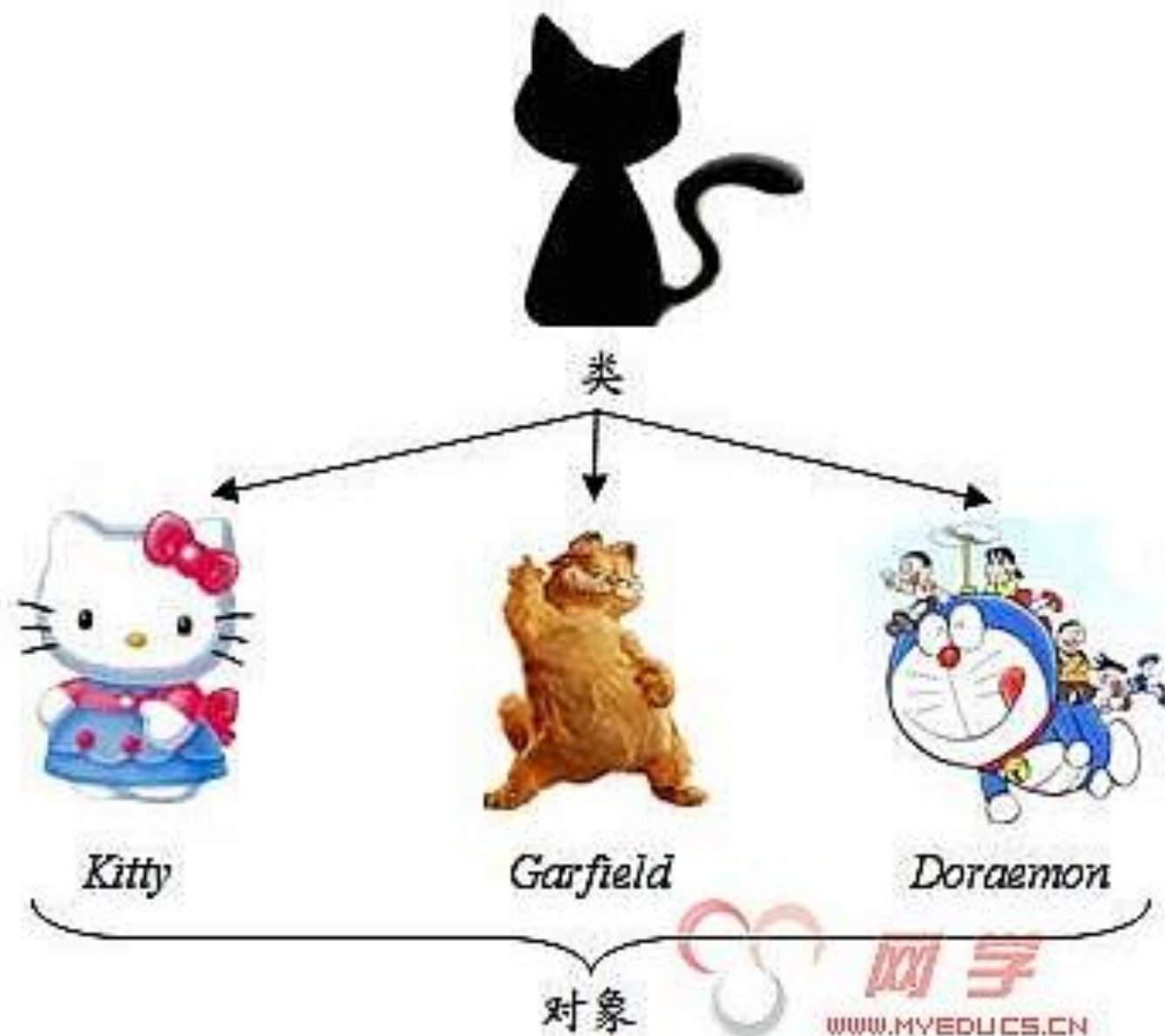
循环结构

- 模块易于识别，每个模块符合单入口、单出口；
- 应该控制goto语句使用



- 面向对象（Object Oriented, OO）方法的出发点和基本原则，是尽可能地模拟现实世界中人类的思维方式，使开发软件的方法和过程尽可能地接近人类解决问题的方法和过程。它具备四个要点：
 - 对象是组成客观世界的基本元素
 - 对象是属于某个类的
 - 继承性
 - 消息传递

对象和类





- 抽象是人类认识问题的基本手段之一，是选择性忽略。抽象的过程，就是对问题进行分析 and 认识的过程。

三大特征

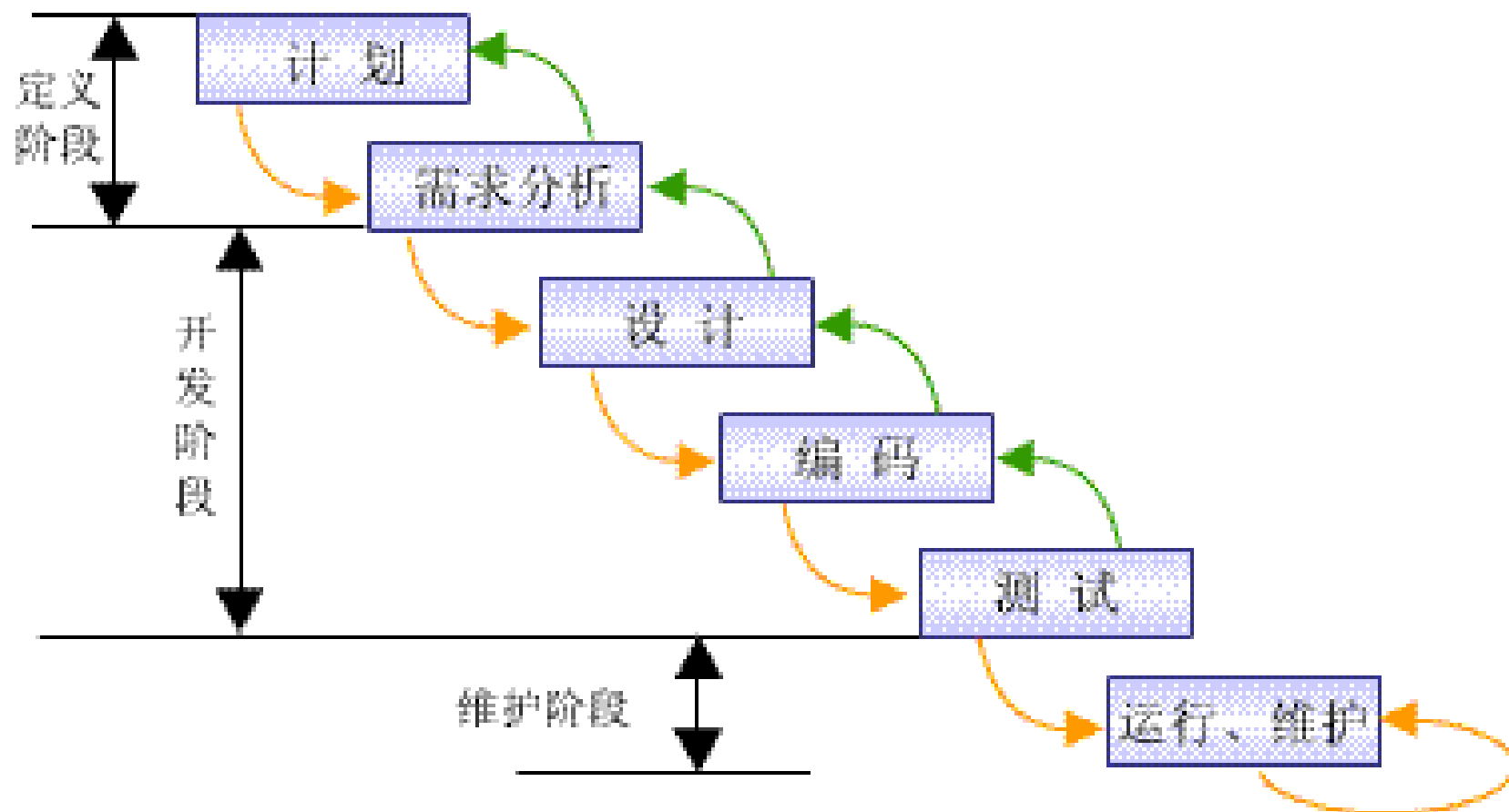


- 封装
- 继承
- 多态

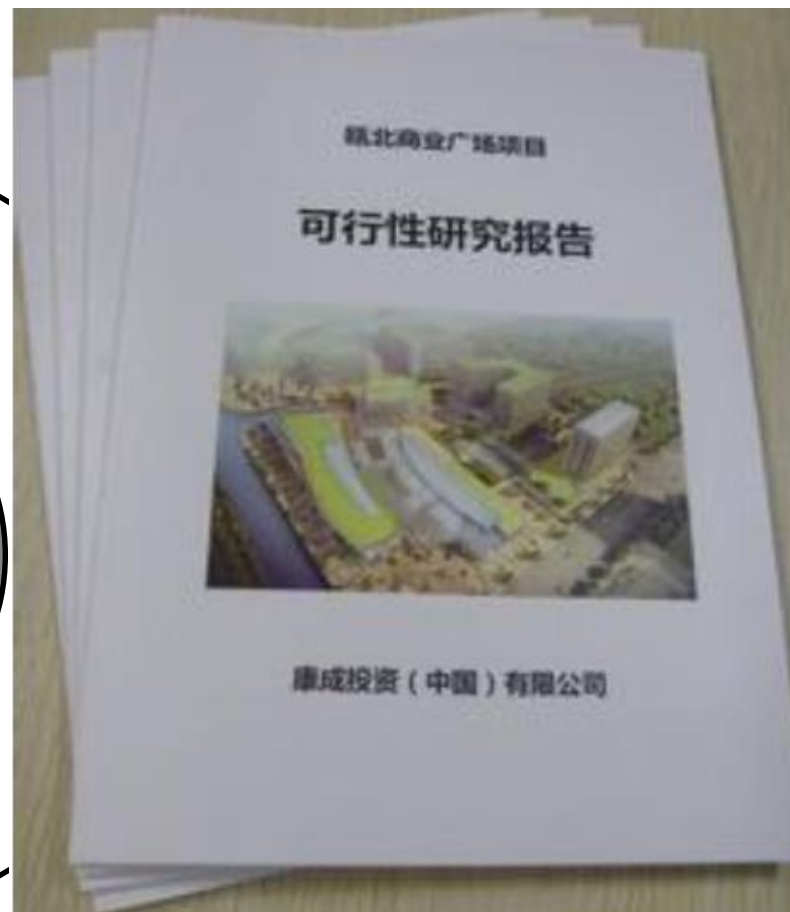
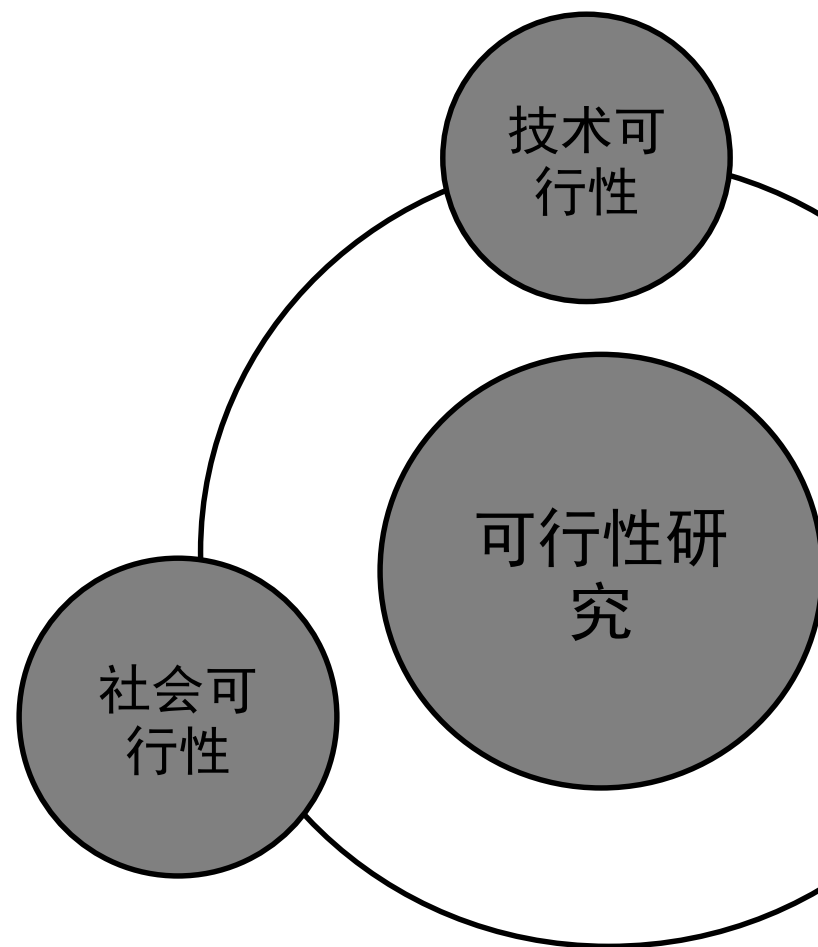


1. 程序设计概述
2. 基本的数据结构
3. 程序设计方法
4. 软件生命周期

软件生命周期



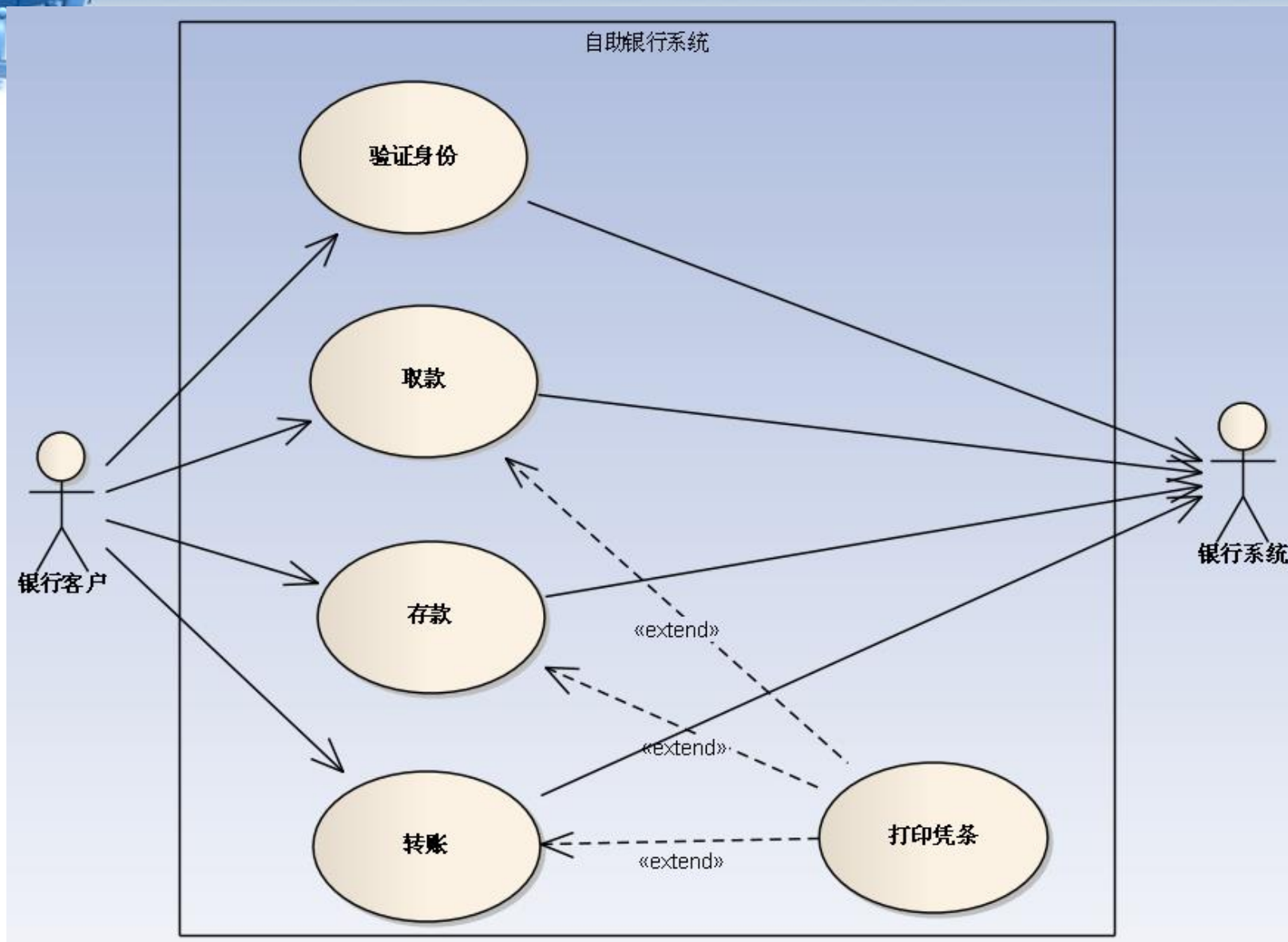
可行性研究

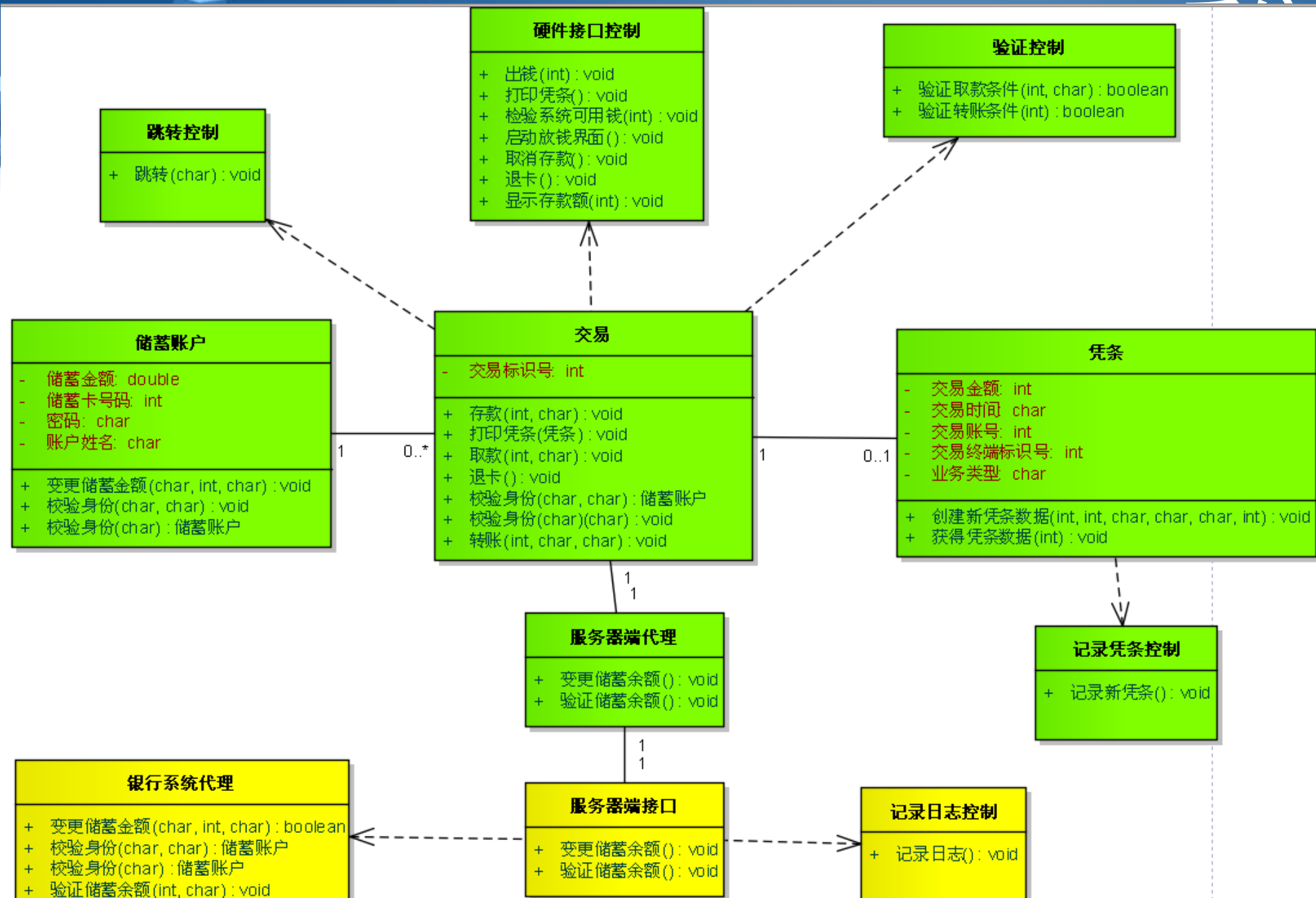




- 软件需求分析就是把软件计划期间建立的软件可行性分析求精和细化，分析各种可能的解法。是确定系统必须完成哪些工作，也就是对目标系统提出完整、准确、清晰、具体的要求。
- 软件需求包括三个不同的层次：业务需求、用户需求和功能需求(也包括非功能需求)。

用例图





编码实现



```
137 <html>
138 <head>
139
140     <meta http-equiv="content-type" content="text/html; charset=utf-8">
141     <meta http-equiv="X-UA-Compatible" content="IE=Edge">
142     <meta content="always" name="referrer">
143     <meta name="theme-color" content="#2932e1">
144     <link rel="shortcut icon" href="/favicon.ico" type="image/x-icon" />
145     <link rel="search" type="application/opensearchdescription+xml" href="/content-search.xml" title="百度搜索" />
146     <link rel="icon" sizes="any" mask href="//www.baidu.com/img/baidu_85beaf5496f291521eb75ba38eacbd87.svg">
147
148
149     <link rel="dns-prefetch" href="//s1.bdstatic.com"/>
150     <link rel="dns-prefetch" href="//t1.baidu.com"/>
151     <link rel="dns-prefetch" href="//t2.baidu.com"/>
152     <link rel="dns-prefetch" href="//t3.baidu.com"/>
153     <link rel="dns-prefetch" href="//t10.baidu.com"/>
154     <link rel="dns-prefetch" href="//t11.baidu.com"/>
155     <link rel="dns-prefetch" href="//t12.baidu.com"/>
156     <link rel="dns-prefetch" href="//b1.bdstatic.com"/>
157
158     <title>百度一下，你就知道</title>
159
```



部署



软件维护





- 瀑布模型的优点

- 为项目提供了按阶段划分的检查点
- 当前一阶段完成后，您只需要去关注后续阶段

- 瀑布模型的缺点

- 在项目各个阶段之间极少有反馈。
- 只有在项目生命周期的后期才能看到结果。
- 通过过多的强制完成日期和里程碑来跟踪各个项目阶段。
- 不适应用户需求的变化



- RUP、Scrum、ICONIX等等
- RUP模型（Rational Unified Process，统一软件开发过程，统一软件过程）是一个面向对象且基于网络的程序开发方法论。
 - 因为软件越发复杂，不可能一次性就能把软件做的完整，而RUP可以用迭代开发来更好的应对变化，使软件的实践更加完整。

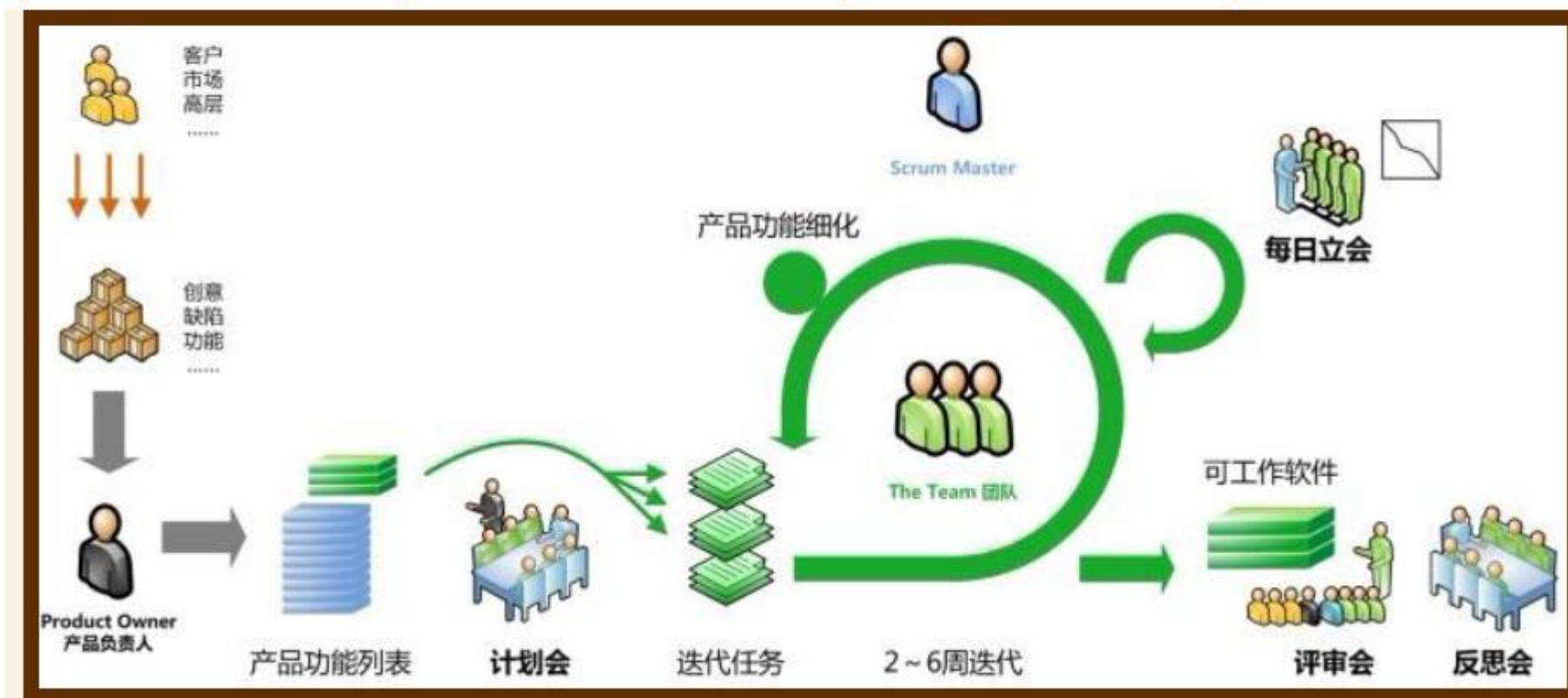
Scrum敏捷方法一分钟扫盲

产品负责人建立条目的产品待开发项，并进行优先级排序。

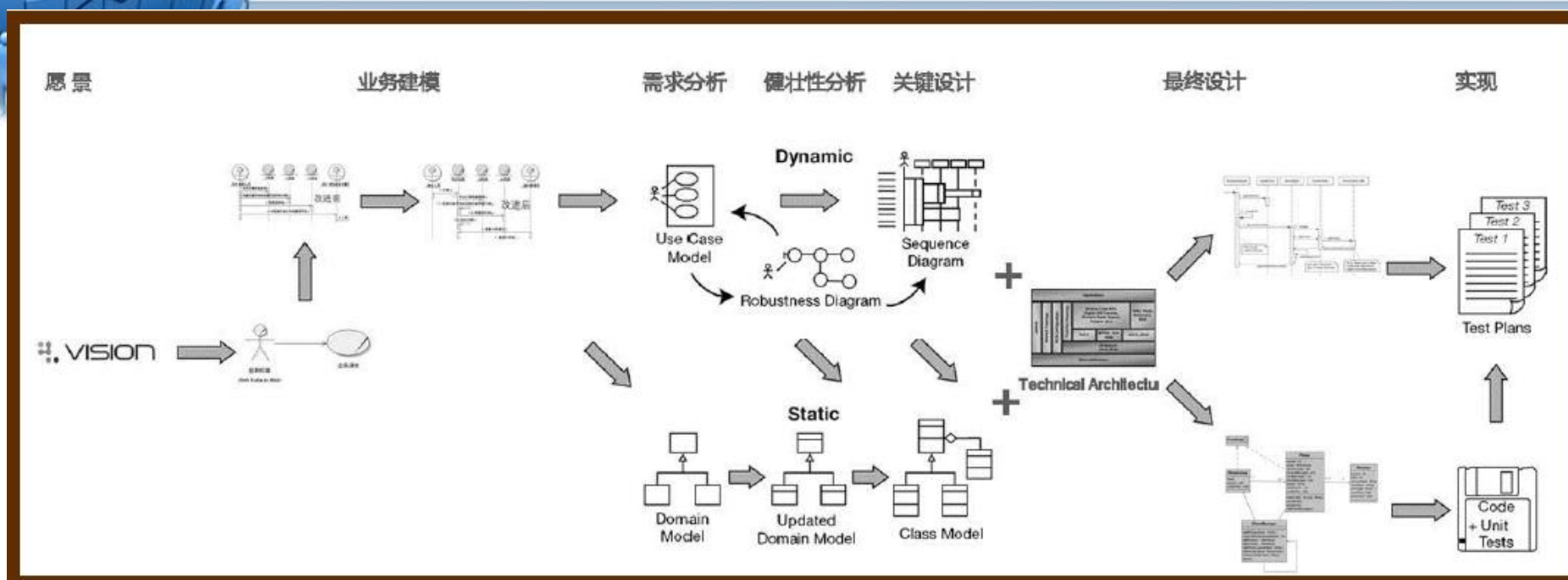
在迭代计划会上，产品负责人讲解本迭代要开发的条目，团队进行估算并放入下一个迭代。

团队在迭代内完成所列需求，每天都开每日“立”会以沟通进度和问题。

在迭代终点的迭代评审会上，团队向产品负责人等展示开发成果。



Scrum是一种迭代式增量软件开发过程，通常用于敏捷软件开发。Scrum包括了一系列实践和预定义角色的过程骨架。



- ICONIX是尽早进入编码阶段，缩短分析设计周期的软件开发方法
- ICONIX过程是一种以最少步骤实现从用例到代码OOAD[object-oriented analysis and design]方法学，开覆盖了软件过程中所有关键的环节。

Questions?

