



WE'RE THINKING OF INNOVATIONS...

《程序设计综合实验》课程介绍

北京邮电大学
信息与通信工程学院
张治

zhangzhi@bupt.edu.cn

2019年3月29日

程序设计？软件开发？

程序设计

编辑



程序设计是给出解决特定问题程序的过程，是软件构造活动中的重要组成部分。程序设计往往以某种程序设计语言为工具，给出这种语言下的程序。程序设计过程应当包括分析、设计、编码、测试、排错等不同阶段。专业的程序设计人员常被称为程序员。

中文名

程序设计

外文名

Programming

软件开发

编辑



软件开发是根据用户要求建造出软件系统或者系统中的软件部分的过程。软件开发是一项包括需求捕捉、需求分析、设计、实现和测试的系统工程。软件一般是用某种程序设计语言来实现的。通常采用软件开发工具可以进行开发。软件分为系统软件和应用软件，并不只是包括可以在计算机上运行的程序，与这些程序相关的文件一般也被认为是软件的一部分。软件设计思路和方法的一般过程，包括设计软件的功能和实现的算法和方法、软件的总体结构设计和模块设计、编程和调试、程序联调和测试以及编写、提交程序。

中文名

软件开发

外文名

Software development

目 录

1

软件开发基本概念

2

软件开发主要过程

3

本课程要求

目 录

1

软件开发基本概念

2

软件开发主要过程

3

本课程要求

软件开发历史

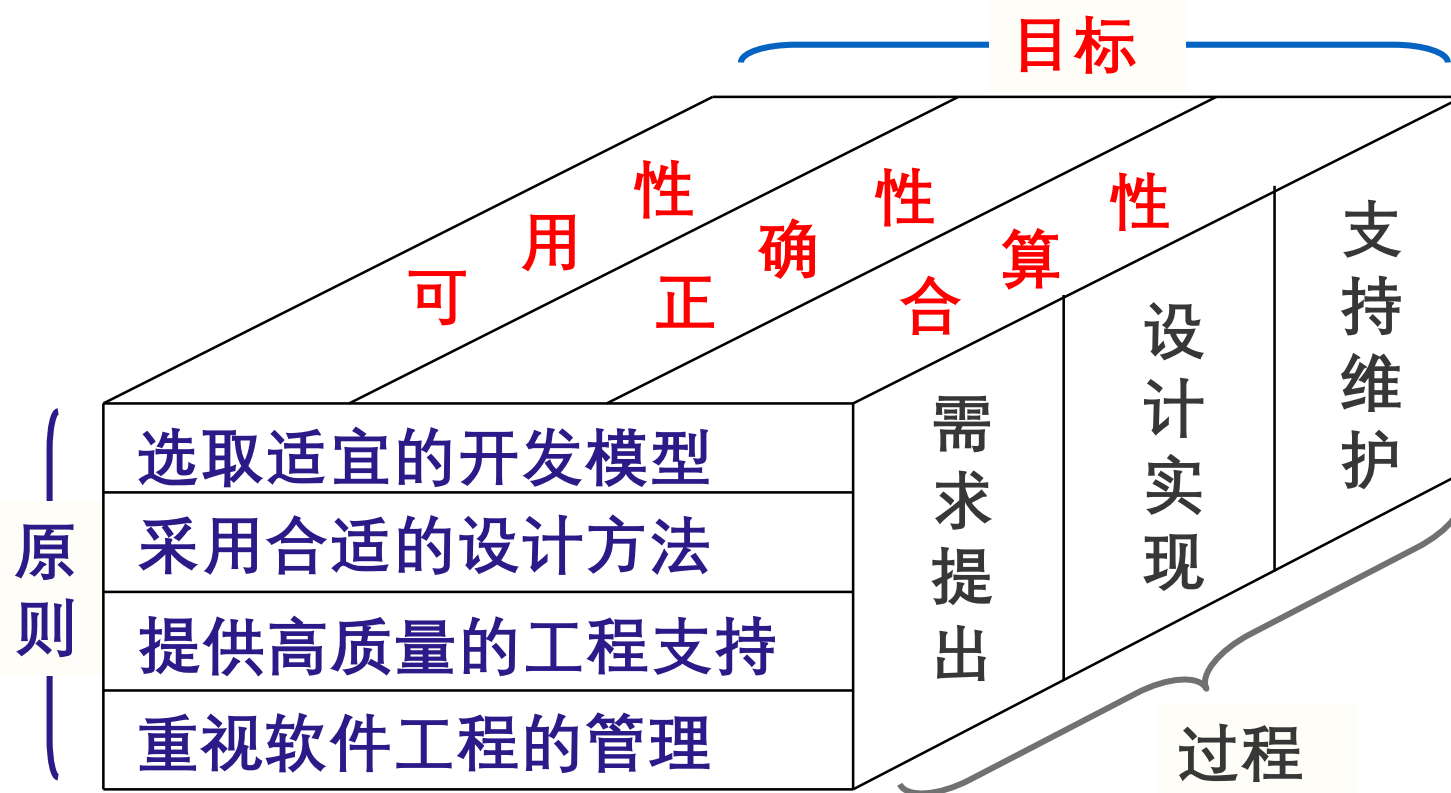
- “软件（开发）”基于计算机，但并不是伴随而生的，实际上当计算机登上历史舞台时，还没有“**software**”这个单词
- 软件开发从雏形到现在的大规模产业化（软件产业），经历了60年左右的历史，大体上可划分为三个阶段：

阶段	时间	特征	关键词
1	上世纪60年代中期前	个体化的设计体制	
2	60年代中期到80年代	软件业开始“膨胀”并系统化	软件危机、 软件工程 、 结构化程序设计
3	80年代后	理论化、体系化	面向对象程序设计 、 软件建模语言与工具

软件工程基本概念

- 1968年德国人Bauer在北大西洋公约组织会议上的定义：
“建立并使用完善的工程化原则，以较经济的手段获得能在实际机器上有效运行的可靠软件的一系列方法”
- 1983年IEEE的软件工程定义：
“软件工程是开发，运行，维护和修复软件的系统方法”
- 1993年IEEE的一个更加综合的定义：
“将系统化的，规范的，可度量的方法应用于软件的开发，运行和维护的过程，即将工程化应用于软件中”

软件工程基本概念



软件工程的框架

软件工程基本概念

一种最常见的软件开发过程模型（瀑布型）：



- 软件开发不仅限于编码，还包括前期（需求）分析与（系统）设计，分析与设计实际上就是对现实世界建模的过程
- 分析、设计和编程的关系：分析的结果是生成一个模型，基于这个模型可以进行设计；而设计的结果就是利用各种编程方法实施整个软件系统

程序设计方法一：结构化程序设计

- 整个程序就是一个对数据顺序加工的**过程**；始终以数据的流向为线索、对数据的流向表达得很清晰，故也称为面向数据流的设计方法
- 特点：
 - ① 自顶向下、逐步求精，从抽象到具体
 - ② 按功能划分模块，各模块相互独立
 - ③ 每一级的“子模块化”都必须正确才能确保整体功能的正确性
- 问题：
 - ① 以过程为目标不符合现实生活的一般情况，系统越复杂偏差越大
 - ② 只能针对个别情况，难以重用到更多的场景
 - ③ 任何元素的微小调整或升级都将导致整个软件的巨大变动

程序设计方法二：面向对象程序设计

- 将实际的事或物视为**对象**（object）并作为目标来进行设计和实现；首要考虑的是对象本身，而各种处理过程仅是附属在对象上的
- 特点：
 - ① 数据及对数据的处理封装在一起，以对外隐蔽的方式把每个组件之间的耦合度降到最低，通过标准接口相互通信，使程序的易维护性和可进化性大大提高
 - ② 将相似对象抽象为**类**（class）并引入继承和多态的概念，程序可重用于较多的场景，可重用性好
 - ③ 符合现实的一般情况（各个事物均有自身的属性、处理过程是依附于事物本身的），使得程序设计的步骤和人的思维方式一致
- 问题：？？

程序设计方法二：面向对象程序设计

- 进行面向对象程序设计的困难：

① 大部分人第一次接触到程序设计时均是从过程式语言（如C、Basic等）开始的，致使他们在学习OOP（Object Oriented Programming）时需要超越以往的知识——当然最好的办法也许是直接从C++开始学习编程

② 人们往往简单的将OOP理解为一种编程风格，甚至局限为某些语言；而实际上OOP只是OOD的自然延续。要想做好面向对象的系统实现，必须从设计时就要以对象的概念来考虑系统；若在设计时还按照过程式设计只注重眼前“过程”的模块化拆分，在编程的时候就很难真正引入OOP

什么是面向对象

- 所谓**面向对象**（OO），就是以对象的观点来分析现实世界中的问题。从普通人认识世界的观点出发，把事物归类、综合，提取共性并加以描述
- 在面向对象的系统中，世界被看成是独立对象的集合，对象之间通过“消息”相互通信。对象被描述为数据（又称为**属性**）以及基于这些数据的行为（又称为**方法**）的复合体
- 软件学中：
面向对象 = 分析方法（面向对象分析） + 设计方法（面向对象设计）
+ 实现方法（面向对象编码）（按步骤分）
面向对象 = 对象 + 类 + 继承 + 封装 + 聚合 + 消息传送 + 多态性 +
（按技术手段分）

什么是面向对象

- 面向对象的基本思想：

- ① 对问题空间进行自然分割，以更接近人类思维的形式（或者准确说是按照OO的规则），建立问题域模型
- ② 通过该模型对客观实体进行结构模拟和行为模拟，使所设计出的软件尽可能直接地描述现实世界
- ③ 构造出模块化的、可重用的、维护性好的软件
- ④ 控制软件的复杂性和降低开发维护费用

在具体进行时，已经有理论定义来规范上述内容，但现实中更依赖于实现人员/团队的能力和經驗

面向对象的分析和设计

(1) 面向对象的分析 (OOA, Object Oriented Analysis)

- 利用面向对象的概念和方法为软件需求建造模型的分析过程，同时也是提取系统需求的过程
- 面向对象分析中建造的模型主要有对象模型、动态模型和功能模型

(2) 面向对象的设计 (OOD, Object Oriented Design)

- 把分析阶段得到的需求转变成一个抽象的系统实现方案，并且该方案符合成本和质量要求

目 录

1

软件开发基本概念

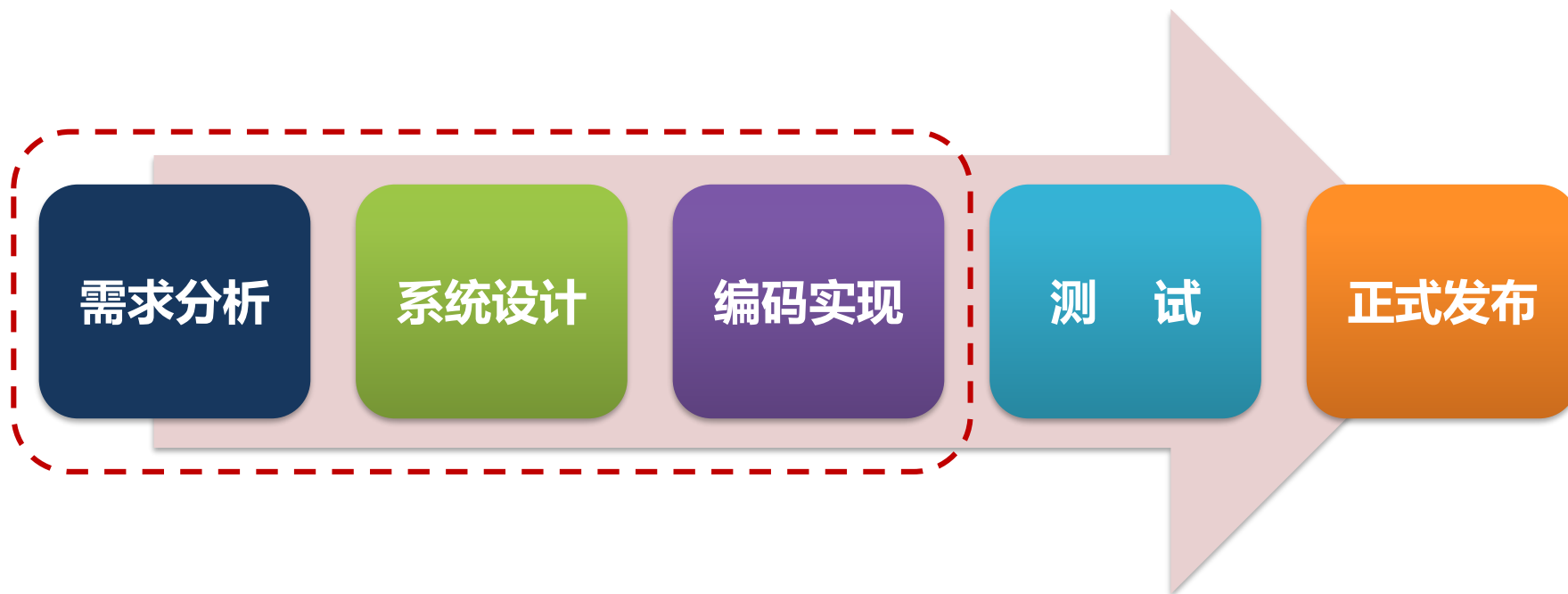
2

软件开发主要过程

3

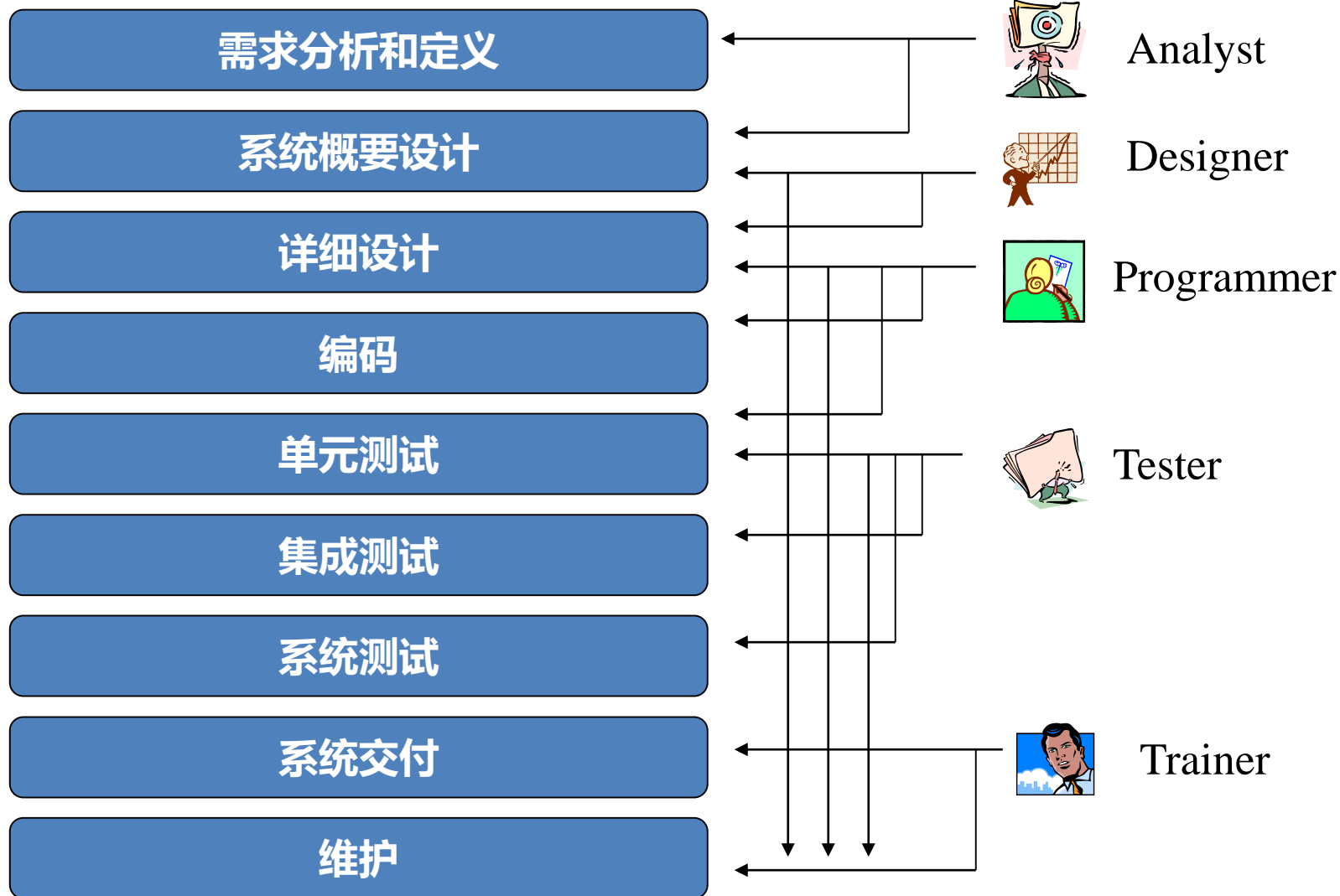
本课程要求

软件开发的全过程



- 软件开发不仅限于编码，还包括前期（需求）分析与（系统）设计，分析与设计实际上就是对现实世界建模的过程
- 分析、设计和编程的关系：分析的结果是生成一个模型，基于这个模型可以进行设计；而设计的结果就是利用各种编程方法实施整个软件系统

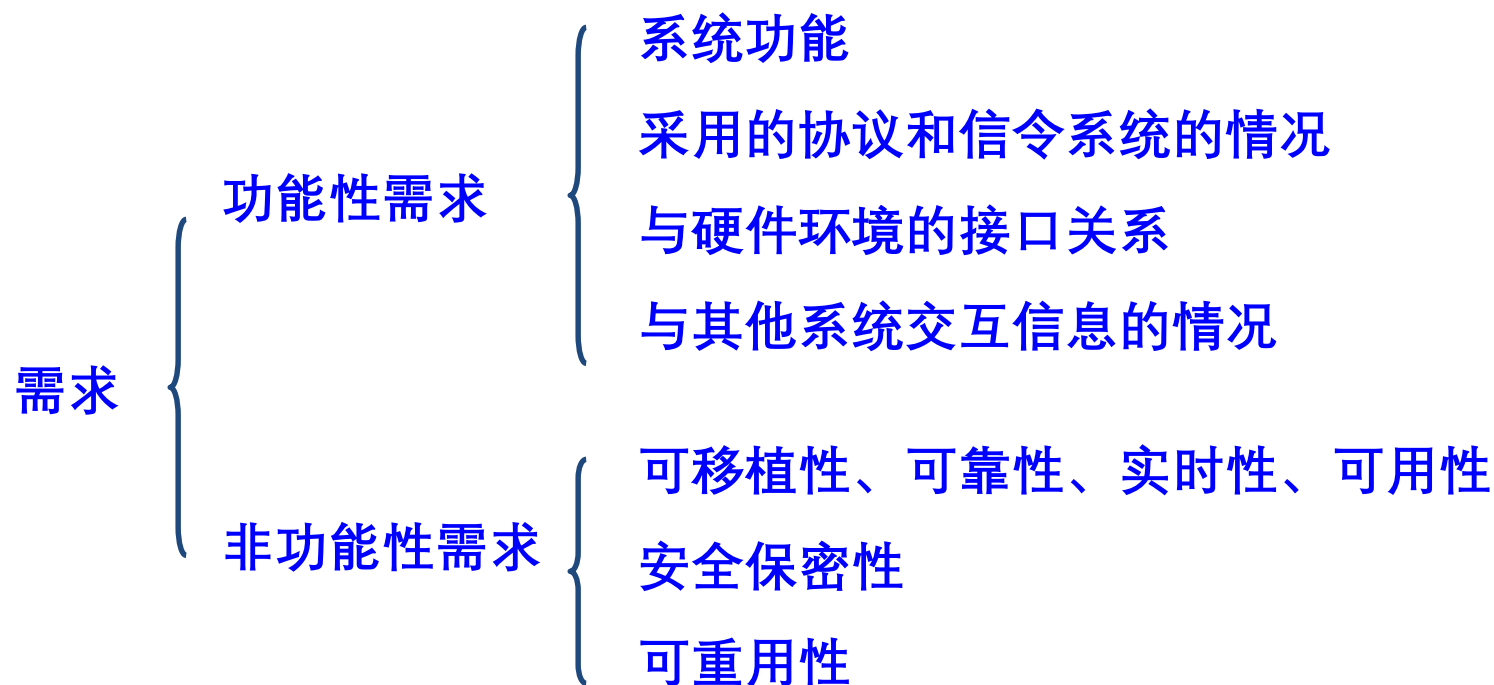
软件开发的全过程



1. 需求分析

- 要解决的问题：系统应该“做什么”

- 主要任务：用户的需求+运行环境 → 形式化描述 → 需求规格说明书



1. 需求分析

- 输出：需求分析文档或需求规格说明书，一般包括

- 任务概述：软件系统的设计目标、运行环境、条件与限制
- 功能需求：软件的功能划分情况、类的定义、系统类图、系统与外部环境之间的消息交互图
- 非功能性能需求：软件系统的可移植性、可靠性、实时性、可用性、安全保密性和可重用性
- 数据描述：给出软件系统用到的数据项的描述



需求分析文档示例

2. 概要设计

- 要解决的问题：怎么做，解决“系统架构”问题
- 主要任务：
 - ➡ 根据需求规格说明书，采用形式化语言，给出系统的结构设计
 - ➡ 划分功能模块，定义模块间的接口
 - ➡ 设计数据结构
 - ➡ 设计主要的算法
- 概要设计中常使用一些形式化说明语言来进行描述，例如SDL
(Specification and Description Language, 规范说明和描述语言)

2. 概要设计

- 输出：概要设计说明书，一般包括
 - ➡ 总体设计：说明软件系统的总体结构
 - ➡ 模块设计：说明功能模块的划分、各模块的功能
 - ➡ 数据结构设计：设计数据的逻辑结构和物理结构
 - ➡ 接口设计：设计系统与环境的接口、功能模块之间的接口
 - ➡ 操作维护管理设计：设计系统故障检测和处理策略



概要设计文档示例

3. 详细设计

- 要解决的问题：怎么做，指导编码
- 主要任务：
 - ➡ 在概要设计的基础上（继续）细化各模块的功能
 - ➡ 详细设计系统行为，用形式化的语言来描述设计结果
 - ➡ 编写测试计划，包括测试方法和测试用例（本课程不涉及）
- 详细设计往往分功能模块进行，往往需要将功能模块分解成多个有限状态自动机，画出状态机的状态转移图
- 详细设计也往往使用SDL等语言进行描述

3. 详细设计

● 输出：详细设计说明书，一般包括

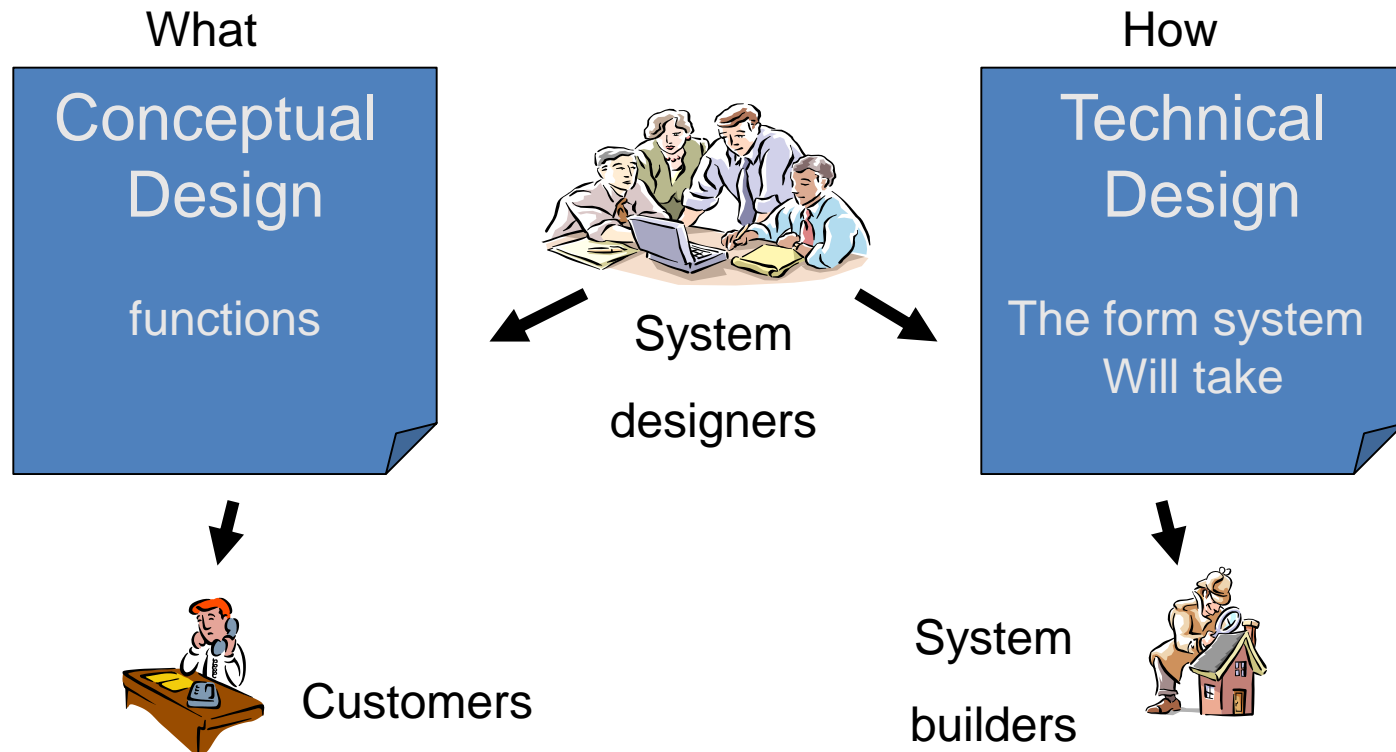
- 全局数据结构说明：描述本系统中的全局数据和数据结构
- 模块详细设计：在概要设计的基础上，详细描述各模块内部功能、数据结构、算法和流程、函数等——在多进程软件设计中，往往改为描述各进程的上述内容
- 接口详细设计：描述各模块内部接口、模块间接口，说明相应的交互方式
- 运行时（Run-time）描述：详细描述系统运行时行为、各进程交互过程等（本课程不要求）



详细设计文档示例

3. 详细设计

- 详细设计中，设计人员最终要将需求转化成一个工作系统，并且必须同时满足用户和系统构建者的要求



4. 编码

- 要解决的问题：“系统实现”问题

- 主要任务：

 - 代码自动生成

 - 人工编码

 - 编译调试

 - 系统生成

- 输出：

 - 代码和最终系统

 - 程序清单文档

5. 测试

- 要解决的问题：尽可能多地发现软件产品（主要指程序）中的错误和缺陷
- 测试用例：包括测试数据和测试结果
- 测试的方法：
 - 黑盒测试：把程序看成一个黑盒子，测试者不需了解程序结构和处理过程，只需根据需求分析中的功能来测试
 - 白盒测试：测试者必须了解程序结构和处理过程，根据程序的内部逻辑来测试



5. 测试

● 测试步骤:

- 单元测试：对单个模块进行测试，主要发现编码和详细设计中产生的错误，通常采用白盒测试，可由程序员完成
- 集成测试：对由多个模块组成的程序进行测试，检查模块间的接口和通信，主要发现设计阶段产生的错误，通常采用黑盒测试
- 系统测试：以需求规格说明书作为依据，测试软件功能和性能等，通常采用黑盒测试

● 输出：软件测试计划、软件测试报告

6. 系统维护

- 系统维护阶段覆盖了从软件交付使用到软件被淘汰的整个时期
- 要解决的问题：改正软件中隐藏的错误，使软件适应新的环境，或者为了扩充和完善软件的功能或性能
- 在系统维护阶段，在修改程序代码的同时要修改相应的需求说明文档、设计文档等，还要进行**回归测试**和软件配置复审等
 - 回归测试是指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误

目 录

1

软件开发基本概念

2

软件开发主要过程

3

本课程要求

本课程概况

- 程序设计综合实验
Comprehensive Experiment of Programming
- 2学分/48学时（12周，4学时/周）

本课程进度

- 第1周 课程介绍；引入MOOC资源

<http://www.icourse163.org/course/BUPT-1003561002>



- 第2周 选题（不限语言，不限开发环境）、
组队（最多2人）



本课程进度

- 第3周 分析阶段：写出需求分析，确定题目的可行性
- 第4周 设计阶段：写出概要设计文档，完成软件总体设计，确定进度安排等
- 第5-9周 实现阶段：写出详细设计文档，完成软件项目的功能开发
- 第10-12周 验收阶段：软件演示（需要做演讲），项目完善

考核方式

1. 书面报告：提交需求分析、概要设计、详细设计、软件演示四份实验报告，按软件工程相关报告及PPT模板格式书写
—— 占总成绩的40%
2. 作品评定：根据程序运行演示、作品测试指标、效果进行考核
—— 占总成绩的50%
3. 平时成绩：课堂表现等
—— 占总成绩的10%



先修课程与学习参考

● 先修课程

- ➡ 《C/C++程序设计与编程方法》，需要具备的知识和技能包括：结构化程序设计、面向对象程序设计的基本概念，使用C/C++语言编写程序的基本技能
- ➡ 《数据结构与算法》，需要具备的知识和技能包括：实际问题的逻辑结构和存储表示，表、栈、队列、树、图等数据结构及相应的操作算法

● 参考

- ➡ 《程序设计实践》北京邮电大学出版社
- ➡ 中国大学MOOC

<http://www.icourse163.org/course/BUPT-1003561002>

开启程序设计之旅

有些路，只能一个人走

有些路看起来很近，可是走下去却很远

有些路，如果不去启程，你永远不知道它有多美丽

程序设计之旅，不可能一天走完

别着急

我愿陪你看遍世间繁华，静待春暖花开



开启程序设计之旅



DO IT IMMEDIATELY !!

有问题吗?

