# 第一章

1. 软件的定义：软件是计算机程序、规程以及运行计算机系统可能需要的相关文档和数据。
2. 开发过程中存在很多问题，比如时间，操作预期成本、操作运算、质量都很难把控，所以在当时的时期内，称为软件危机。（此为软件危机的一个背景，要知道什么是软件危机。）
3. **软件危机**：是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。
4. 软件危机的主要表现：进度严重拖后，成本超出多达一个数量级；生产率极低，软件质量低劣到甚至根本不能使用，而且难于管理。
5. 从管理和技术两方面解决软件危机。
6. 软件工程定义：软件工程是为了经济地获得能在实际机器上高效运行的可靠软件，而确立一系列工程原理（方法）

软件工程本质上就是说我们如何把程序编的更好、质量更好、时间上更节省、成本上更降低，主要研究从过程、方法、工具这几个角度来为软件开发提供更好的指导思想，这些过程中告诉我们该怎么做，过程中的方法指导我们如何做一些具体事情，方法背后具体工具。（这是软件工程主要的内容）

1. 结构化方法和面向对象方法的主要思想、区别和联系

结构化方法是传统的开发方法，其基本思想是模块化，以分解和抽象为主要手段。

面向过程的方法是现代的开发方法，其基本思想是将现实世界的事物映射为对象，并将数据和操作进行封装。

# 第二章

1. 软件生命周期的定义：软件产品从考虑其概念开始到交付使用，直至最终退役为止的整个过程。
2. 软件生命周期的几个阶段和各阶段的基本任务：
3. 软件定义时期：问题定义，可行性研究，需求分析。
4. 软件开发时期：a)系统设计：总体设计，详细设计

b)系统实现：编码和单元测试，综合测试

3）软件维护时期。

问题定义：要解决什么问题。

可行性研究：定义的问题是否有可行的解决方法——可行性研究报告

需求分析：目标系统必须做什么——需求规格说明书

总体设计：怎么实现目标系统——概要设计说明书

详细设计：如何具体实现系统——详细设计说明书

1. 软件过程模型的作用：告诉人们应该去遵循一个什么样的过程去开发软件系统
2. 过程模型（前半句优点后半句缺点）
   1. 瀑布模型：规范，文档驱动；不灵活，产生大量文档
   2. 原型模型：尽早发现需求，揭示风险；对于不是与用户密切交互的系统，较难获取用户评价，容易忽视文档
   3. 螺旋模型：将瀑布、原型、增量三种模型结合起来，增加了风险分析，弥补他们的缺点；要求具有丰富经验的风险分析专家
   4. 喷泉模型：提高效率，节省开发时间；不严格区分开发阶段，不便于管理
   5. 增量模型：快速提交产品，控制风险；增加了不同构件集成的风险

# 第三章

1. 可行性研究的四个方面：技术可行性、经济可行性、社会可行性和操作可行性。

# 第四章

1.什么是需求工程：应用已经证实有效的技术、方法进行需求分析，确定客户需求，帮助分析人员理解问题并定义目标系统的所有外部特征的一个学科.

2.需求工程的分类：系统需求工程和软件需求工程。

3.需求开发的四步：需求获取、需求建模与分析、需求文档化、需求评审。

# 第五章

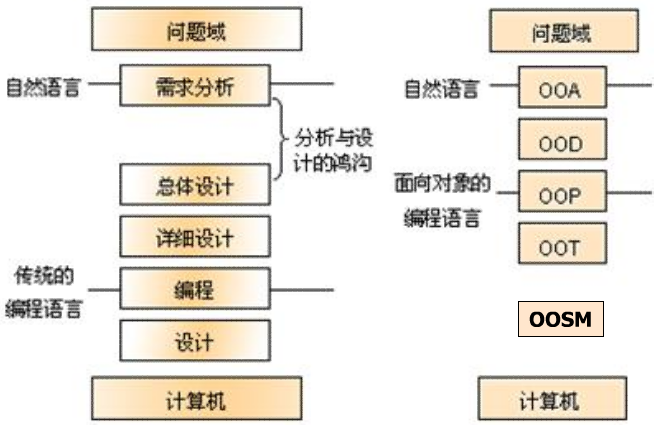
结构化分析方法

基本思想

数据流图

# 第六章

1.传统开发与面向对象开发的区别与主要思想



1. 面向对象的三大基本概念：封装、继承、多态
2. 什么是UML：Unified Modeling Language (统一建模语言或标准建模语言），为软件开发的所有阶段提供模型化和可视化支持。
3. UML的九种图：用例图、类图、对象图、组件图、分布图、顺序图、协作图、状态图、活动图。
4. UML的九种图的用途：

功能：用例图

动态行为：顺序图、协作图、状态图、活动图

静态结构：类图、对象图

物理架构：部署图、构件图。

细化：

用例图：需求捕获、测试依据

类图：类以及类之间的相互关系

对象图：对象以及对象之间的相互关系

构件图：构件及其相互依赖关系

部署图：构建在各节点上的部署

顺序图：强调时间顺序的交互图

协作图：强调对象协作的交互图

状态图：类所经历的各种状态

活动图：对工作流程建模

# 第七章

1. 面向对象三种分析类：

边界类：描述与参与者直接打交道的对象，一般是一些UI界面

控制类：描述系统的行为，即系统做什么

实体类：描述系统的中的数据信息（有些数据是实体类，而有些是实体类的属性）

# 第八章

1. 模块化：解决一个复杂问题时，自顶向下逐层把软件划分成若干模块的过程
2. 模块化目的：为了降低软件的复杂性
3. 耦合

# 第十章

1.面向对象的几个设计原则：

单一职责原则

开放——封闭原则

借口隔离原则

Liskov替换原则

依赖倒置原则

2.设计模式

广义上讲：设计模式是对被用来在特定场景下解决一般设计问题的类和相互通信的对象的描述。

狭义上讲：设计模式就是对特定问题的描述或解决方案，往往直接对应一段程序代码。

# 第十一章

用户界面设计的原则：以用户为中心，减轻用户的记忆负担，保持界面的一致性

# 第十三章

1. 软件缺陷：未实现要求的功能；出现了不应该出现的错误；实现了未提到的功能；未实现虽未明确提及但应该实现的目标；最终用户会认为不好
2. 黑盒测试：完全不考虑程序的内部结构和处理过程，只知道软件产品应该具有的功能，通过测试检验每个功能是否都能正常使用。又称为功能性测试或行为测试。
3. 白盒测试：有时称为透明盒测试，已知产品内部工作过程，通过测试检验产品内部动作是否按照产品规格说明的规定正常进行。
4. 静态测试：是指不运行被测程序本身，仅通过分析或检查源程序的语法、结构、过程、接口等来检查程序的正确性。
5. 动态测试：是指通过运行被测程序，检查运行结果与预期结果的差异，并分析运行效率和健壮性等性能。
6. 单元测试也称模块测试，是针对软件设计的最小单元程序模块进行测试的工作。
7. 集成测试也称组装测试，它的任务是按照一定的策略对单元测试的模块进行组装，并在组装过程中进行模块接口与系统功能测试
8. 确认测试也称为有效性测试，是验证软件的有效性，即验证软件的功能和性能及其他特性是否符合用户要求。
9. 系统测试是测试软件系统是否能与硬件协调工作，测试与其他软件协调运行的情况。
10. 软件测试过程:单元测试,集成测试,确认测试,系统测试
11. 软件调试和软件测试的区别：调试作为测试的后续工作主要是解决和排除测试中出现软件缺陷的工作。

# 第十四章

1.软件维护的4种类型：纠错性维护、完善性维护、预防性维护、适用性维护

2.什么是逆向工程：逆向工程就是正向过程的逆过程，指在软件生存周期中，将软件的某种形式描述转换成更抽象形式的活动。

3.什么是再工程：再工程是指在逆向工程所获信息的基础上修改重构已有的系统，产生的一个新版本，或者说利用这些信息修改或重构软件系统的工作。