1. 软件工程学概述
2. 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。
3. 软件危机包含下述两方面的问题：如何开发软件，以满足对软件日益增长的需求；如何维护数量不断膨胀的已有软件。
4. 软件危机的典型表现：
5. 对软件开发成本和进度的估计常常很不准确。
6. 用户对“已完成的”软件系统不满意的现象经常发生。
7. 软件产品的质量往往靠不住。
8. 软件常常是不可维护的。
9. 软件通常没有适当的文档资料。
10. 软件成本在计算机系统总成本中所占的比例逐年上升。
11. 软件开发生产率提高的速度，远远跟不上计算机应用迅速普及深入的趋势。
12. 产生软件危机的原因：
13. 与软件本身的特点有关
14. 软件开发与维护的方法不正确
15. 软件配置主要包括程序，文档和数据等成分
16. 消除软件危机的途径：
17. 首先对计算机软件有一个正确的认识。如彻底消除“软件就是程序“的错误观念。
18. 充分认识到软件开发是一种组织良好，管理严密，各类人员协同配合，共同完成的工程项目。
19. 推广使用在实践中总结出来的开发软件的成功的技术和方法，并且研究探索更好更有效的技术和方法。
20. 应该开发和使用更好的软件工具。
21. 软件工程的基本原理
22. 用分阶段的生命周期计划严格管理
23. 坚持进行阶段评审
24. 实行严格的产品控制
25. 采用现代程序设计技术
26. 结果应能清楚地审查
27. 开发小组的人员应该少而精
28. 承认不断改进软件工程实践的必要性
29. 软件工程方法学包含3个要素：方法，工具和过程
30. 目前使用最广泛的软件工程学方法分别是：传统方法学和面向对象方法学
31. 传统方法学也称为生命周期方法学或结构化范型。使用结构化技术。
32. 面向对象方法学是一种以数据为主线，把数据和对数据的操作紧密结合起来的方法。
33. 面向对象方法学具有以下4个要点：
34. 把对象作为融合了数据及在数据上的操作行为的统一的软件构件
35. 把所有对象都划分成类
36. 按照父类与子类的关系，把若干个相关类组成一个层次结构的系统。在类等级中，下层派生类自动拥有上层基类中定义的数据和操作，这种现象称为继承。
37. 对象彼此之间仅能通过发送消息互相联系。
38. 对象的所有私有信息都被封装在该对象内，不能从外界直接访问，这就是封装性。
39. 面向对象方法学的出发点和基本原则，是尽量模拟人类的思维方式
40. 软件的生命周期由软件定义，软件开发和维护运行3个时期组成
41. 软件生命周期的8个阶段：
42. 问题定义
43. 可行性研究
44. 需求分析
45. 总体设计
46. 详细设计
47. 编码和单元测试
48. 综合测试
49. 软件维护
50. **瀑布模型**的特点
51. 阶段间具有顺序性和依赖性
52. 推迟实现的观点
53. 质量保证的观点
54. **快速原型模型**：在软件开发的早期，快速开发一个目标软件系统的原型；让用户对其进行评价并提出修改意见；然后开发人员根据用户的意见对原型提出的意见对原型进行修改。

第二章 可行性研究

1. 可行性研究的目的就是用最小的代价在尽可能短的时间内确定问题是否能够解决。
2. 可行性研究的任务：
3. 首先要进一步分析和澄清问题定义
4. 导出系统的逻辑模型
5. 探索若干种可供选择的主要解法
6. 技术，经济，操作可行性+法律社会效益
7. 可行性研究最根本的任务是对以后的行动方针提出建议
8. 可行性研究过程（8个步骤）
9. 复查系统规模和目标
10. 研究目前正在使用的系统
11. 导出新系统的高层逻辑模型
12. 进一步定义问题
13. 导出和评价供选择的解法
14. 推荐行动方针
15. 草拟开发计划
16. 书写文档提交审查
17. **数据流图**
18. **数据字典**

第三章 需求分析

1. 需求分析是软件定义时期的最后一个阶段，它的基本任务是准确地回答“系统必须做什么”这个问题
2. 需求分析方法的准则
3. 必须理解并描述问题的信息域，根据这条准则应该建立数据模型。E-R图
4. 必须定义软件应完成的功能，这条准则要求建立功能模型。数据流图
5. 必须描述作为外部事件结果的软件行为，这条准则要求建立行为模型。状态转换图
6. 必须对描述信息，功能和行为的模型进行分解，用层次的方式展示细节
7. 需求分析的任务
8. 确定对系统的综合要求：1.功能需求

2.性能需求

3.可靠性和可用性需求

4.出错处理需求

5.接口需求

6.约束

7.逆向需求

8.将来可能提出的要求

1. 分析系统的数据要求
2. 导出系统的逻辑模型
3. 修正系统开发计划
4. **面向数据流自顶向下求精。**
5. 实体联系图：数据模型中包含3种相互关联的信息：数据对象，数据对象的属性及数据对象彼此间相互关联的关系。
6. **状态转换图**

第五章 总体设计

1. 总体设计的基本目的就是回答“概括地说，系统应该如何实现”这个问题。总体设计阶段的另一项重要任务是设计软件的结构
2. 总体设计过程：
3. 设想供选择的方案
4. 选取合理的方案
5. 推荐最佳方案
6. 功能分解
7. 设计软件结构
8. 设计数据库
9. 制定测试计划
10. 书写文档：系统说明，用户手册，测试计划，详细的实现计划，数据库设计结果
11. 审查和复审
12. 通常把编码和测试统称为实现
13. 所谓编码就是把软件设计结果翻译成用某种程序设计语言书写的程序。
14. 测试步骤：
15. 模块测试
16. 子系统测试
17. 系统测试
18. 验收测试
19. 平行运行
20. 软件维护的定义：软件维护就是在软件已经交付使用之后，为了改正错误或满足新的需要而修改软件。
21. 决定软件可维护性的因素：
22. 可理解性
23. 可测试性
24. 可修改性
25. 可移植性
26. 可重用性

38.