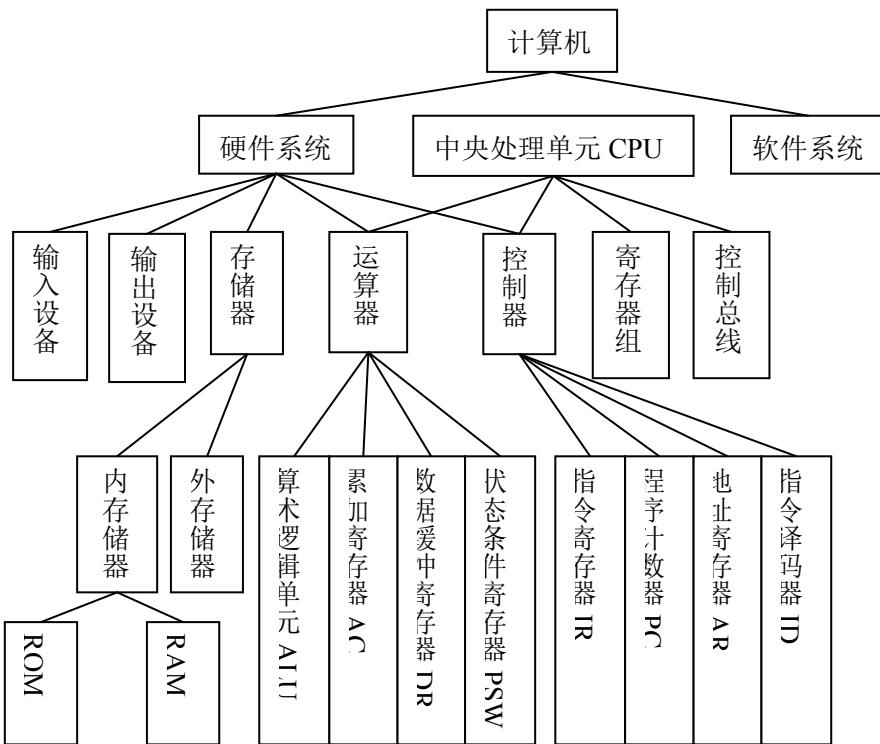


软件设计师笔记

一、计算机系统基础知识

1. CPU 的功能: 程序控制、操作控制、时间控制、数据处理。

2. 计算机系统组成示意图:



3. 数据表示: 原码、反码、补码、移码。(0 为正, 1 为负)

4. 移码: 如果机器字长为 N, 偏移量为 2^{N-1} , 则 $[X]_{\text{移}} = 2^{N-1} + [X]_{\text{补}}$ (X 为纯整数)。 $[X]_{\text{移}} = 1 + X$ (X 为纯小数)。

5. IEEE754: 符号位 (1 位, 0 正 1 负) + 阶码 (8 位, +127) + 尾数 (23 位, 小数点在最高位之后, 随后省略最高位)。

6. 浮点数加减: 0 操作数检查, 对阶操作(小阶变大阶), 尾数加减(阶码用双符号位, 尾数用单符号位), 结果规格化, 舍入处理(判定溢出)。

7. 常用校验码: 奇偶校验码 (只能检错)、海明码 (纠错加检错)、循环冗余校验码 (CRC)。

8. 计算机体系结构分类: 单处理系统, 并行处理与多处理系统, 分布式处理系统。

9. 指令集的发展: CISC (复杂指令集计算机): 增强原有指令的功能, 用更为复杂的新指令取代之; RISC (精简指令集计算机): 减少指令总数简化指令功能, 优化编译降低复杂度。

10. 指令控制方式: 顺序方式, 重叠方式, 流水方式。

11. RISC 中的流水线技术: 超流水线技术, 超标量技术, 超长指令字技术。

12. 并发性的解决: 阵列处理机, 并行处理机, 多处理机。

13. 主存与 cache 地址映射方式: 全相联映射-将主存一个块的地址与内容一起存入 cache 中,

拷贝灵活但是比较器电路难设计实现；直接映射-一个主存块只能拷贝到 cache 的特定行上去，硬件简单成本低但是容易冲突效率下降；组相联映射-主存块放到哪个组是固定的，但是放到哪一行是灵活的，折中办法广泛采用。

14. **虚拟存储器**：页式：页表硬件小，查表速度快但不利于存储保护；段式：界限分明便于程序的模块化设计，易于编译修改和保护，但主存利用率低，产生大量碎片，查表速度慢；段页式：折中办法，广泛采用，但地址变换速度比较慢。

15. **计算机与外设数据交换**：无条件传送，查询方式传送，中断方式传送，直接存储器存取方式（DMA）：CPU 仅在过程开始和结束时有处理，过程中 DMA 占用系统总线传送数据。

16. **多中断处理办法**：多中断信号线法，中断软件查询法，菊花链法，总线仲裁法，中断向量表法。

17. **总线分类**：内部总线：芯片的互连，系统总线：CPU，内存，接口等的连接；外部总线：数据交换。

18. **信息安全 5 要素**：机密性，完整性，可用性，可控性，可审查性。

19. **加密技术**：对称加密（私人密钥加密）：数据加密标准（DES）+三重 DES+RC-5+国际数据加密算法（IDEA）+高级加密标准（AES）；非对称加密（公开密钥加密，比如 RSA 算法）：加密模型+认证模型。

20. **PKI**：一种遵循既定标准的密钥管理平台，能够为所有网络应用提供加密和数字签名等密码服务及所必需的密钥和证书管理体系，必须具有权威认证机构、数字证书库、密钥备份和恢复系统、证书作废系统、应用接口。

21. **计算机可靠性**：可靠性：串联系统 $R=R_1 \cdot R_2$ ，并联系统 $R=1 - (1-R_1)(1-R_2)$ ；平均无故障时间：串联系统 $K=K_1+K_2$ ，并联系统 $K=1/K \cdot (1+1/2)$ 。

22. **计算机系统性能评测常用方法**：时钟频率，指令执行速度，等效指令速度法，数据处理速率（PDR）。

二、程序语言基础知识

1. **编程语言之间的翻译形式**：汇编，解释，编译。

2. **程序设计语言的定义**：语法，语义，语用。

3. **程序语言的分类**：命令式程序设计语言：FORTRAN，PASCAL，C 语言；面向对象的程序设计语言（C++，JAVA，smalltalk）；函数式程序设计语言（LISP）；逻辑型程序设计语言（PROLOG）。

4. **程序语言的基本成分**：数据，运算，控制，传输。

5. **汇编语言源程序**：指令语句，伪指令语句，宏指令语句。

6. **汇编程序**：将汇编语言所编写的源程序翻译成机器指令程序。

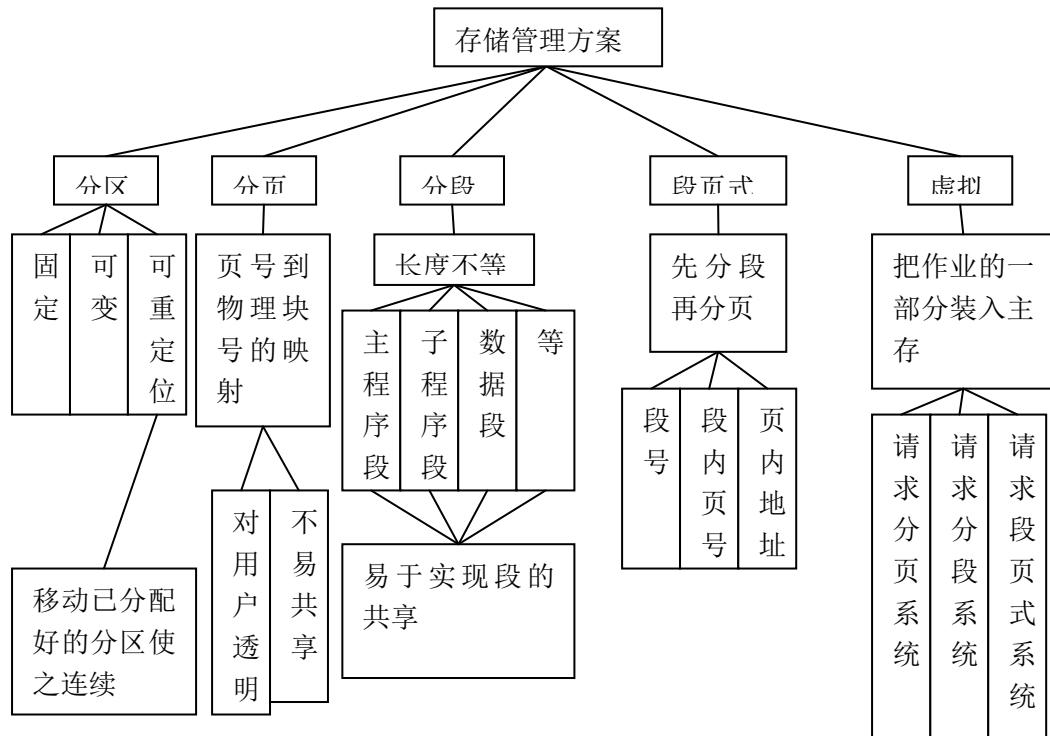
7. **编译过程**：源程序---词法分析---语法分析---语义分析---中间代码生成---代码优化---目标代码生成---目标代码。（另全过程有符号表管理和出错处理）

8. **解释程序**：分析部分：词法分析、语法分析、语义分析->中间代码；解释部分：解释执行中间代码。

9. **编译与解释方式的比较**：前者效率高；后者更具有灵活性和可移植性。

三、操作系统知识

1. **操作系统的应用**: 通过资源管理提高计算机系统的效率；改善人机界面，向用户提供友好的工作环境。
2. **操作系统的特征**: 并发性、共享性、虚拟性和不确定性
3. **操作系统的功能**: 处理机管理、文件管理、存储管理、设备管理、作业管理
4. **操作系统的类型**: 批处理操作系统（单道、多道）、分时操作系统（UNIX, 多路性、独立性、交互性、及时性）、实时操作系统（快速的响应时间、有限的交互能力、高可靠性）、网络操作系统、分布式操作系统、微机操作系统（Windows, Linux）、嵌入式操作系统
5. **程序并发执行的特点**: 失去了程序的封闭性，程序和机器的执行程序的活动不再一一对应，并发程序间的相互制约性
6. **进程的组成**: 程序、数据、进程控制块（PCB）
7. **进程的状态**: 新建、就绪、运行、阻塞、终止
8. 原语（若干条机器指令组成）**的特点**: 执行时不能被分割，要么都做要么都不做
9. **进程的同步**: 进程间完成一项任务时直接发生相互作用的关系
10. **进程的互斥**: 系统中各进程互斥使用临界资源
11. **信号量的意义**: 若 $S \geq 0$ 表示某资源的可用数；若 $S < 0$ 则其绝对值表示阻塞队列中等待该资源的进程数
12. **PV操作**（低级通讯方式）: P 操作申请一个资源，V 操作释放一个资源
13. **高级通讯方式**: 共享存储模式，消息传递模式，管道通信
14. **管城**: 由一些共享数据、一组能为并发进程所执行的作用在共享数据上的操作的集合、初始代码以及存取权组成，一种同步机制
15. **进程调度**（可剥夺+不可剥夺）**算法**: 先来先服务算法：主要用于宏观调度；时间片轮转：微观调度，分时间片占用 CPU；优先级调度：根据优先级（静态+动态）；多级反馈调度：分多个优先级队列前三中调度的综合，先执行新进程。
16. **产生死锁的原因**: 竞争资源及进程推进顺序算法
17. **产生死锁的四个必要条件**: 互斥条件，请求保持条件，不可剥夺条件，环路条件
18. 死锁的处理：预防；避免（银行家算法：先计算，如果分配资源后系统进入不安全状态则不予分配）；检测（系统定时运行死锁检测程序）；解除（资源剥夺法+撤销进程法）
19. **安全状态**: 系统能按某种顺序来为每个进程分配其所需资源，使每个进程都能顺序完成
20. 线程：是进程中的一个实体，是被系统独立分配和调度的基本单位，基本上不拥有资源
21. **存储器管理**: 分配和回收主存空间、提高主存利用率、扩充主存、有效保护主存信息
22. 存储器的层次结构：寄存器、快速缓存（cache）、主存、外存
23. **地址重定位**: 将逻辑地址转变成物理地址的过程，分静态和动态（程序运行时完成转换）
24. **虚拟存储页面置换算法**: 最佳置换（以后最久不会被使用），先进先出置换，最近最少未使用，最近未用
25. **设备管理的目标**: 提高设备的利用率，为用户提供方便统一的界面
26. **磁盘调度**（使平均寻道时间最少）**算法**: 先来先服务，最短寻道时间优先，扫描算法，单向扫描调度
27. **文件**: 具有符号名的、在逻辑上具有完整意义的一组相关信息项的集合
28. **文件系统**: 操作系统中实现文件统一管理的一组软件和相关数据的集合，专门负责管理和存取文件信息的软件机构
29. **存储管理方案示意图**



30. **文件系统的安全**: 一类涉及到技术、管理、法律、道德和政治等问题；另一类涉及操作系统的安全机制

31. **文件系统的可靠性**: 转储和恢复，日志文件，文件系统的一致性

32. **作业**: 系统为完成一个用户的计算任务（或一次事物处理）所做的工作总和

33. **作业状态**: 提交，后备，执行，完成

34. **作业调度算法**: 先来先服务，短作业优先，响应比高优先，优先级调度，均衡调度

35. **网络操作系统**: 集中模式，客户端/服务器模式，对等模式

36. **嵌入式操作系统**: 微型化，可定制，实时性，可靠性，易移植性

37. **shell 变量**: 用户定义变量，系统定义变量，shell 定义变量

四、系统开发和运行知识

1. **软件工程**: 指应用计算机科学、数学及管理科学等原理，以工程化的原则和方法来解决软件工程的问题。其目的是提高软件生产率，提高软件质量，降低软件成本

2. **软件生存周期**: 可行性分析与项目开发计划、需求分析、概要设计、详细设计、编码、测试、维护

3. **软件生存周期模型**: 瀑布模型（按顺序阶段性开发），演化模型（先构造一个初始版本再不断改进），螺旋模型（制定计划、风险分析、实施工程、用户评估），喷泉模型（重视用户需求，允许各步骤交叉进行）

4. **软件开发方法**: 结构化方法（面向数据流，自顶向下逐层分解），jackson 方法（面向数据结构），原型化方法（开发一个对用户透明的框架，然后根据用户需求壮大），面向对象开发方法

5. **需求分析**: 确定软件系统的综合要求，分析软件系统的数据要求，导出系统的逻辑模型，修正项目开发计划，可开发一个原型系统

6. **数据域的属性**: 数据流，数据内容，数据结构

7. **需求工程**: 需求开发 (需求捕获、需求分析、编写规格说明书、需求验证), 需求管理 (定义需求基线、处理需求变更、需求跟踪)
8. **软件开发项目管理**: 成本估算 (自顶向下估算方法、自底向上估算方法、差别估算方法), 风险分析 (风险识别、风险预测、风险评估), 进度管理 (Gantt 图、PERT 图), 人员管理 (主程序员组、无主程序员组、层次式程序员组)
9. **软件配置管理的目标**: 标识变更, 控制变更, 确保变更正确地实现, 报告有关变更
10. **基线**: 软件生存期中各开发阶段的一个特定点, 相当于断点, 便于检查和肯定阶段成果
11. **软件开发工具**: 需求分析工具, 设计工具, 编码与排错工具, 测试工具
12. **软件维护工具**: 版本控制工具, 文档分析工具, 开发信息库工具, 逆向工程工具, 再工程工具
13. **软件管理和软件支持工具**: 项目管理工具, 配置管理工具, 软件评价工具
14. **软件过程评估的意义**: 改进软件过程, 降低软件风险
15. **软件能力过程成熟度**: 初始级, 可重复级, 已定义级, 已管理级, 优化级
16. **统一过程**: 起始阶段, 精化阶段, 构建阶段, 移交阶段, 产生阶段
17. **敏捷开发**: 极限编程 (计划游戏、小型发布、隐喻、简单设计、测试先行、重构、结对编程、持续集成), 水晶法 (不同的项目需要不同的策略、约定和方法论), 并列征求法 (多个小组并行递增实现产品), 自适应软件开发
18. **软件质量特性**: 功能性, 可靠性, 易使用性, 效率, 可维护性, 可移植性
19. **软件质量保证**: 应用技术方法, 进行正式的技术评审, 测试软件, 标准的实施, 控制变更, 度量, 记录保存和报告
20. **软件复杂性**: 规模, 难度, 结构, 智能度
21. **软件质量**: 设计质量 (设计的规格说明书符合用户的要求), 程序质量 (程序按照设计规格说明所规定的情况正确执行)
22. **设计质量评审内容**: 是否合乎用户要求, 可靠性, 保密措施实现情况, 操作特性实施情况, 性能实现情况, 是否具有可修改性可扩充性、可互换性、可移植性、可测试性、复用性
23. **程序质量评审内容**: 功能结构, 功能的通用性, 模块的层次, 模块结构, 处理过程的结构
24. **软件实现容错的主要手段就是冗余**
25. **冗余**: 结构冗余, 信息冗余, 时间冗余, 冗余附加调用
26. **系统分析**: 对当前系统进行详细调查, 收集数据; 建立当前系统的逻辑模型; 分析现状, 提出意见和新目标; 建立新系统的逻辑模型; 编写系统方案说明书
27. **结构化分析方法** (自顶向下逐层分解): 数据流图, 数据字典, 小说明, 补充材料
28. **数据流图** (DFD: 描述系统的分解): 数据流, 加工, 数据存储, 外部实体-----P190
29. DFD 信息流的类型: 变换流 (输入、变换、输出), 事务流 (输入到事务中心、选择执行)
30. **数据字典** (DD: 对图中各成分进行说明): 数据流条目, 数据存储条目, 数据项条目, 加工条目-----P194
31. **加工逻辑描述方法**: 结构化语言, 判定表, 判定树
32. **系统分析报告作用**: 描述目标系统的逻辑模型, 作为用户与开发人员之间的协议或合同, 作为目标系统验收和评价的依据
33. **系统分析报告内容**: 组织情况概述, 现行系统概述, 系统逻辑模型, 新系统在各个业务处理环节拟采用的管理方法、算法或模型, 与新系统相配套的管理制度和运行体制的建立, 系统设计与实施的初步计划, 用户领导审批意见
34. **系统设计**: 概要设计 (设计软件系统总体结构、数据结构及数据库设计、编写概要设计文档、评审), 详细设计 (分模块进行详细的算法设计、对模块内的数据结构进行设计、对

数据库进行物理设计、其他设计、编写详细设计说明书、评审)

35. **系统设计基本原理**: 抽象, 模块化, 信息隐蔽, 模块独立 (耦合: 模块之间联系的紧密程度; 内聚: 模块内部各元素之间联系的紧密程度)

36. **系统结构设计原则**: 分解-协调, 自顶向下, 信息隐蔽、抽象, 一致性, 明确性, 模块间耦合尽可能小、内聚尽可能高, 模块的扇入、扇出系数要合理, 模块的规划适当

37. **子系统划分的原则**: 要具有相对独立性, 之间的数据依赖性要尽可能小, 结果应使数据冗余较小, 应考虑今后管理发展的需要, 应便于系统分阶段实现, 应考虑各类资源的利用

38. **模块** (组成系统的基本单位) **四要素**: 输入和输出, 处理功能, 内部数据, 程序代码

39. **模块结构图**: 模块, 调用, 数据, 控制, 转接-----P203

40. **变换分析** (从变换流型的 DFD 导出程序结构图): 确定输入流和输出流、分离出变换中心, 第一级分解, 第二级分解, 事务分析-----P205

41. **面向数据结构的设计方法** (**jackson 方法**): 分析确定输入输出数据的逻辑结构, 找出输入与输出数据结构有对应关系的数据单元, 从描述数据结构的 jackson 图导出描述程序结构的 jackson 图, 列出所有操作并分配到程序结构图上, 用伪码表示程序-----P207

42. **Jackson 图**: 表达数据元素间的逻辑关系 (顺序, 选择, 重复) 或程序结构 (顺序, 选择, 重复)

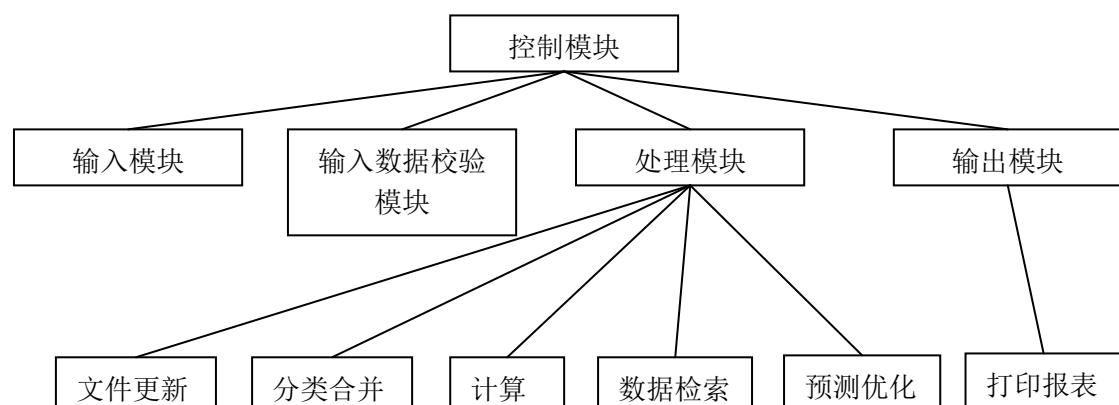
43. **系统详细设计**: 代码设计 (唯一性、合理性、可扩充性、简单性、适用性、规范性、系统性), 输入设计 (确定输入数据内容、输入方式设计、输入格式设计、校对方式设计), 输出设计 (确定输出内容、选择输出设备与介质、确定输出格式), 处理过程设计 (程序流程图、盒图: NS 图、形式语言、决策树、决策表), 用户界面设计 (菜单方式、会话管理方式、提示方式与权限管理), 安全控制设计 (环境性因素、数据处理因素)

44. **系统实施的任务**: 按总体设计方案购置和安装计算机网络系统, 软件准备, 培训, 数据准备, 投入转换和试运行

45. **系统实施的步骤**: 按总体设计方案购置和安装计算机网络系统, 建立数据库系统, 程序设计, 收集有关数据并进行录入工作和系统测试, 人员培训、系统转换和试运行

46. **程序设计方法**: 结构化方法 (模块内部程序各部分要自顶向下地结构化划分、各程序部分应按功能组合、各程序部分的联系尽量使用调用子程序方式而不是 GOTO 方式), 原型方法 (将带有普遍性的功能模块集中、开发完成通用模块、生成程序模块原型), 面向对象的方法 (一个简单直接的映射过程)

47. **程序模块图**



48. **测试**: 以最少的人力和时间发现潜在的各种错误和缺陷

49. **测试过程**: 制定测试计划, 编制测试大纲, 设计生成测试用例, 实施测试, 生成测试报告

50. **软件测试**: 单元测试 (模块接口、局部数据结构、重要的执行路径、出错处理、边界条

件), 集成测试 (把模块按系统设计说明书的要求组合起来进行测试), 确认测试 (有效性测试、软件配置审查、验收测试), 系统测试 (恢复测试、安全性测试、压力测试、性能测试、可靠性及可用性和可维护性测试、安装测试)

51. **测试方法**: 静态测试 (人工检测、计算机辅助静态分析), 动态测试 (白盒测试: 逻辑覆盖、循环覆盖、基本路径测试; 黑盒测试: 等价类划分、边值分析、错误猜测、因果图)-----P223

52. **调试** (改正测试时发现的错误): 试探法, 回溯法, 对分查找法, 归纳法, 演绎法

53. **系统文档**: 软件开发过程中的文档, 硬件采购和网络设计中的文档, 系统建设过程中的各种往来文件、会议纪要、会议单据等, 程序资料和培训教程等

54. **系统转换** (新系统试运行成功后与旧系统相互转换): 直接转换, 并行转换, 分段转换

55. **系统可维护性的评价指标**: 可理解性, 可测试性, 可修改性

56. **系统维护**: 硬件维护, 软件维护 (正确性维护、适应性维护、完善性维护、预防性维护), 数据维护

57. **系统维护的步骤**: 提出维护或修改要求, 领导审查并做出答复, 领导分配任务、维护人员执行修改, 验收维护成果并登记修改信息

58. **系统评价**: 立项评价, 中期评价, 结项评价

五、网络基础知识

1. **计算机网络** (计算机技术+通信技术) **的发展**: 具有通信功能的单机系统->具有通信功能的多机系统->以共享资源为目的的计算机网络->以局域网及因特网为支撑环境的分布式计算机系统

2. **计算机网络的功能**: 数据通信, 资源共享, 负载均衡, 高可靠性

3. **计算机网络的分类**: 局域网 (LAN: 10m~1000m), 城域网 (MAN: 10km), 广域网 (WAN: 100km 以上)

4. **网络拓扑结构**: 总线型, 星型, 环型, 树型, 分布式 (无严格的布线规定和形状, 各节点有多条线路相连)

5. **OSI 七层参考模型**: 物理层 (物理地传送比特流), 数据链路层 (负责两相邻节点间无差错传送以帧为单位的数据), 网络层 (提供端到端的交换网络数据传送功能), 传输层 (提供可靠的数据传输服务), 会话层 (提供会话管理服务), 表示层 (提供格式化的表示和转换数据服务), 应用层 (提供网络与用户应用软件之间的接口服务)

6. **网络互连设备**: 中继器 (物理层上实现局域网网段互连, 用于扩展局域网网段长度), 集线器 (特殊的多路中继器, 有信号放大功能并便于网络维护); 网桥 (工作与数据链路层, 用于连接两个局域网网段), 交换机 (按每一个包中的 MAC 地址相对简单地决策信息转发); 路由器 (网络层异构互连, 连接多个逻辑上分开的网络), 网关 (在两个不同类型协议的网络系统之间进行通信)

7. **网络传输介质**: 有线介质 (双绞线, 同轴电缆: 直接传输数字信号, 光纤: 传输光信号、需信号转换); 无线介质 (微波: 利用无线电波传输, 红外线: 传输红外光信号, 激光: 传激光信号, 卫星通信: 传输电磁波信号)

8. **局域网组成部件**: 服务器 (文件服务器、打印服务器、通信服务器), 客户端 (用户与网络应用接口设备), 网络设备 (网卡, 收发器, 中继器, 集中器, 网桥, 路由器等), 通信介质 (数据的传输媒体), 网络软件 (底层协议软件、网络操作系统等)

9. **协议**: 规定通信时的数据格式、数据传送时序以及相应的控制信息和应答信号等内容

10. **网络的标准**: 电信标准, 国际标准 (IEEE 标准等), Internet 标准 (自发标准非政府干预)

11. **决定局域网特性的主要技术**: 用以传输数据的传输介质, 用以连接各种设备的拓扑结构, 用以共享资源的介质访问控制方法

12. **局域网协议**: LAN 模型 (物理层, 数据链路层: 逻辑链路控制子层、介质访问控制); 以太网 (CSMA/CD 技术: 边发送边接收、时刻侦听信道); 令牌环网 (适用于环型网络结构的分布式介质访问控制: 广播发送令牌、目标站进行处理); FDDI (类似令牌环网协议、光纤作为传输介质)

13. **广域网协议**: 点对点协议 (PPP: 主要用于拨号上网, 建立点对点连接发送数据), 数字用户线 (xDSL: 不对称数字用户线 ADSL, 甚高速数字用户线 VDSL), 数字专线 (电信数字数据网固定专线, 电信铺设), 帧中继 (在用户网络接口之间提供用户信息流的双向传送, 并保持顺序不变), 异步传输模式 (ATM: 面向分组的快速分组交换模式, 使用异步时分复用技术), X.25 协议 (在本地数据终端设备和远程数据终端设备之间提供一个全双工、同步的透明信道)

14. **TCP/IP 协议簇特性**: 逻辑编址, 路由选择, 域名解析, 错误检测和流量控制

15. **TCP/IP 模型**

应用层	文件传输 协议 FTP	远程登录 协议 Telnet	电子邮件协 议 SMTP	网络文件 服务协议 NFS	网络管理 协议 SNMP	应用层	
表示层							
会话层							
传输层	TCP				UDP	传输层	
网络层	IP		ICMP	ARP (地址解析协议)		RARP	
数据链路 层	Ethernet IEEE 802.3	FDDI	Token-Ring/ IEEE 802.5	ARCnet	PPP/SLIP	网际层	
物理层						网络接口 层	
						硬件层	

16. **Internet 地址格式**: 域名格式, IP 地址格式

17. **解决 IP 地址短缺问题**: 长期 (使用 Ipv6), 短期 (使用网络地址翻译技术 NAT: 在子网内部使用局部地址, 外部使用少量的全局地址, 通过路由器进行内部地址和外部地址的转换)

18. **Ipv6**: 40 个字节的首部长度, 16 个字节的 IP 地址长度

19. **服务端口**: 公共端口 (0~1023) 注册登记端口 (1024~65535)

20. **Internet 高层协议**: 域名服务 (UDP 端口, 53, 通过 DNS 服务器将域名变换为 IP 地址); 远程登录服务 (TCP 端口, 23, 将用户计算机与远程主机连接起来); 电子邮件服务 (TCP 端口, 25: SMTP and 110: POP3, 发送: SMTP, 接收: SMTP+POP3); WWW 服务 (TCP 端口, 80, 基于客户端/服务器模式的信息发送技术和超文本技术的综合); 文件传输服务 (TCP 端口, 20: 数据连接 and 21: 控制连接)

21. **网络安全的威胁**: 计算机存储着国家、机构、组织的秘密信息或个人的隐私成为某类人攻击的目标; 软件规模的膨胀容易使系统存在缺陷; 信息传输的安全性存在隐患; 网络协议本身的漏洞也会引发安全问题

22. **网络安全**: 运行系统安全, 信息系统的安全, 信息传播的安全, 信息内容的安全

23. **信息系统的需求**: 保密性, 完整性, 可用性, 可控性, 可核查性

24. **网络的安全威胁**: 物理威胁, 网络攻击, 身份鉴别, 编程威胁, 系统漏洞

25. **网络的信息安全**: 信息的存储安全 (用户的标识与验证, 用户存取权限限制, 系统安全监控, 计算机病毒防治, 数据的加密, 计算机网络安全); 信息的传输安全 (链路加密, 节点加密, 端到端加密)

26. **防火墙**: 建立在内外网络边界的过滤封锁机制, 防止不良数据包进出被保护的内部网络

27. **防火墙的分类**: 包过滤型 (直接转发报文, 对用户透明), 应用代理网关型 (通过服务器建立连接), 状态检测型 (建立状态连接表, 跟踪检测每一个会话状态)

28. **典型防火墙的体系结构**: 包过滤路由器（在网络层对进出内部网络所有信息进行分析限制），双宿主主机（代理服务器软件在双宿主主机上运行，每一个接口连接不同网段），被屏蔽主机（由过滤路由器和应用网关组成，包过滤+代理服务，内网和外网双重保障），被屏蔽子网（由两个包过滤路由器和一个应用网关组成，最安全的防火墙系统）

六、多媒体基础知识

1. **媒体**: 感觉媒体（使人产生感觉的媒体），表示媒体（传输感觉媒体的中介媒体），表现媒体（进行信息输入和输出的媒体），存储媒体（用于存储表示媒体的物理介质），传输媒体（传输表示媒体的物理介质）
2. **多媒体的特性**: 多样性，集成性，交互性，非线性，实时性，信息使用的方便性，信息结构的动态性
3. **虚拟现实**: 运用计算机对现实世界进行全面仿真，创建与现实社会类似的环境，通过多种传感设备使用户“投入到该环境中”，实现用户与该环境直接进行自然交互
4. **虚拟现实技术的特征**: 多感知（听觉感知、力觉感知、触觉感知、运动感知、味觉感知、嗅觉感知），沉浸（用户感受到的模拟环境的真实程度），交互（用户对模拟环境内物体的可操作程度和从环境得到反馈的自然程度）
5. **虚拟现实的分类**: 桌面虚拟现实，完全沉浸的虚拟现实，增强现实性的虚拟现实，分布式虚拟现实
6. **声音感觉的三个指标**: 音量，音调，音色
7. **声音信号的数字化**: 采样，量化，编码
8. **数字语音的数据压缩方法**: 波形编码，参数编码，混合编码
9. **声音合成**: 语音合成（发音参数合成、声道模型参数合成、波形编辑合成），音乐合成
10. **MIDI**（乐器数字接口）: 数字音乐的国际标准
11. **色彩三要素**: 亮度，色调，色饱和度
12. **彩色空间**: RGB，CMY，YUV 等
13. **图形数据表示形式**: 矢量图形（用数学的方式描述图像），位图图像（用像素点来描述的图）
14. **图像的属性**: 分辨率，图像深度，真彩色和伪彩色
15. **图像的数据量**: 图像的总像素*图像深度/8
16. **数据压缩**: 有损压缩（压缩过程中损失一定信息），无损压缩（行程长度编码，增量调制编码，霍夫曼编码）
17. **多媒体数据压缩编码标准**: JPEG，MPEG，H.261
18. **动画**（本质是运动）: 实时动画（用各种算法来实现运动物体的运动控制），矢量动画（由矢量图衍生出的动画形式），二维动画（对传统动画的一个改进），三维动画（根据三维数据模型）
19. **彩色电视制式**: NTSC 制，PAL 制，SECAM 制
20. **数字视频标准**: 采样频率，分辨率，数据量
21. **超文本**（将文本中遇到的一些相关内容通过链接组织在一起）三要素：节点，链，网络
22. **超媒体**: 用超文本方式组织和处理多媒体信息
23. **流媒体**: 在网络中使用流式传输技术的连续时基载体。信息经过压缩之后放到专用流服务器上，让浏览器一边下载一边观看或收听
24. **多媒体计算机硬件系统**: 音频卡，视频卡，光盘驱动器，扫描仪，光学字符阅读器，触

摸屏, 数字化仪, 操纵杆, 绘图仪、投影仪和激光视盘播放器

25. **多媒体计算机软件系统**: 多媒体操作系统, 多媒体应用软件的开发工具, 多媒体应用软件

七、数据库技术基础

1. **数据库系统**: 数据库, 硬件, 软件, 人员

2. **DBMS** (数据库管理系统) **的功能**: 数据定义, 数据库操作, 数据库运行管理, 数据组织、存储和管理, 数据库的建立和维护, 与其他软件系统的通信功能等

3. **DBMS 的特征**: 数据结构化且统一管理, 有较高的数据独立性, 数据控制功能 (数据库的安全性保护、数据的完整性、并发控制、故障恢复)

4. **DBMS 分类**: 关系数据库系统 (实体间的联系用关系表示), 面向对象的数据库系统 (以对象形式对数据建模), 对象关系数据库系统 (在关系数据模型基础上提供处理新的数据类型操作的能力)

5. **数据库系统体系结构**: 集中式 (数据、数据管理、数据库功能等都集中在一起), 分布式 (物理上分布+逻辑上分布), C/S 模式 (客户端负责数据表示服务、服务器负责数据库服务), 并行结构 (多个 CPU 物理上连在一起处理)

6. **数据库的三级模式**: 概念模式 (数据库中全部数据的逻辑结构和特征的描述、只涉及型的描述而不涉及具体的值), 外模式 (用户与数据库系统的接口、用户用到那部分数据的描述), 内模式 (数据物理结构和存储方式的描述、数据在数据库内部的表示方式)

7. **数据库的两级映像**: 模式/内模式映像 (实现概念模式与内模式的转换), 外模式/模式映像 (实现外模式与概念模式的转换)

8. **数据的独立性**: 物理独立性 (数据库的内模式改变时数据的逻辑结构不变), 逻辑独立性 (用户的应用程序与数据库的逻辑结构相互独立)

9. **数据模型**: 概念数据模型 (E-R 模型等), 基本数据模型 (层次模型: 用树型结构表示数据间的联系、网状模型: 用网络结构表示数据间的联系、关系模型: 用表格结构表示实体间的联系、面向对象模型: 对象标识+封装+对象的属性+类和类层次+继承)

10. **数据模型三要素**: 数据结构, 数据操作, 数据的约束条件

11. **E-R 图**: 实体 (矩形), 联系 (菱形), 属性 (椭圆形)

12. **完整性约束**: 实体完整性, 参照完整性, 用户自定义完整性

13. **关系代数运算**: 并, 交, 差, 笛卡尔积, 投影, 选择, 连接, 除

14. **SQL 语言的特点**: 综合统一, 高度非过程化, 面向集合的操作方式, 两种使用方式 (自含式、嵌入式), 语言简洁易学易用

15. **SQL 语言的组成**: 数据定义语言, 交互式数据操纵语言, 事务控制, 嵌入式 SQL 和动态 SQL, 完整性, 权限管理

16. **SQL 数据定义**: 创建 (create), 修改 (alter), 删除 (drop): 表 (table), 视图 (view[as select]), 索引 (index[on])

17. **SQL 数据查询**: select...from...where...group by...having...order by...-----P375

18. **插入数据**: insert into...values...

19. **修改数据**: update...set...=...where...

20. **删除数据**: delete from...where...

21. **授权**: grant...on...to...(with grant option)

22. **回收权限**: revoke...on...from...

23. **函数依赖**: 反映属性间的联系 ($X \rightarrow Y$); 完全函数依赖: (学生 ID, 所修课程 ID) \rightarrow 成绩; 部分函数依赖: (学生 ID, 所修课程 ID) \rightarrow 学生姓名; 平凡函数依赖: $X \rightarrow Y$ 且 Y 包含于 X ; 非平凡函数依赖: $X \rightarrow Y$ 且 Y 不包含于 X ; 传递函数依赖: $X \rightarrow Y$, $Y \rightarrow Z$
24. **规范化**: 1NF: 每个分量都不可再分; 2NF: 消除非主属性对码的部分函数依赖; 3NF: 消除非主属性对码的传递函数依赖
25. **模式分解标准**: 无损连接, 保持函数依赖
26. **事务的 ACID 性质**: 原子性, 一致性, 隔离性, 持久性
27. **事务管理**: 事务开始 (begin transaction), 事务提交 (commit), 事务回滚 (rollback)
28. **数据库故障**: 事务内部故障, 系统故障, 介质故障, 计算机病毒
29. **数据备份方法**: 静态转储和动态转储, 海量转储和增量转储, 日志文件
30. **数据恢复步骤**: 反向扫描文件日志, 对事物的更新操作执行逆操作, 继续反向扫描和更新, 直到事务的开始标志
31. **并发控制的技术**: 封锁 (写锁、读锁)
32. **数据不一致性**: 丢失修改, 不可重复读, 读脏数据

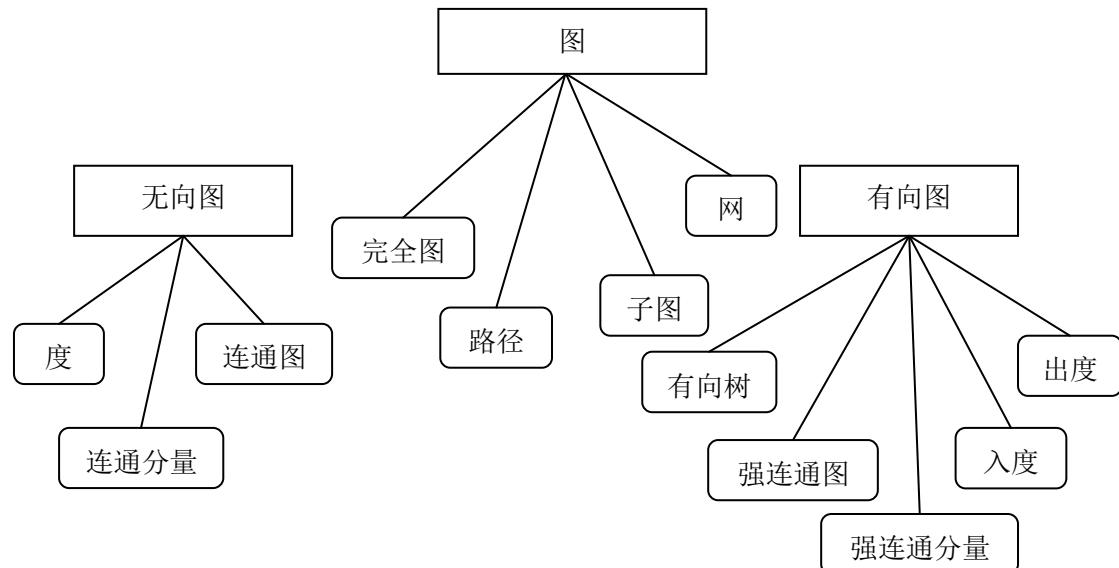
八、数据结构

1. **数据结构**: 数据元素的集合及元素间的相互关系和构造方法
2. **线性表的存储结构**: 顺序存储, 链式存储
3. **单链表节点**: `typedef struct node{ int data; struct node *link; }NODE,*LinkList;`
4. **双向链表**: 每个节点有两个指针, 分别指出直接前驱和直接后继
5. **循环链表**: 尾节点指针指向第一个节点
6. **静态链表**: 借助数组来描述线性表的链式存储结构
7. **栈**: 后进先出; 初始化栈: `InitStack(S)` 判栈空: `StackEmpty(S)` 入栈: `Push(S,x)` 出栈: `Pop(S)` 读取栈顶元素: `Top(S)`-----顺序存储+链式存储
8. **队列**: 先进先出, 尾入头出; 初始化队: `InitQueue(Q)` 判队空: `Empty(Q)` 入队: `EnQueue(Q,x)` 出队: `DeQueue(Q)` 读队头元素: `FrontQue(Q)`-----顺序存储+链式存储
9. **串**: 仅由字符构成的有限序列, 是取值范围受限的线性表-----串的模式匹配
10. **数组**: 定长线性表在维数上的扩张, 一般不做插入删除运算
11. **矩阵**: 特殊矩阵 (元素分布有一定的规律: 对称矩阵、三角矩阵、对角矩阵); 稀疏矩阵 (非零元素远少于零元素且分布无规律: 用三元组存储(行号, 列号, 值))
12. **广义表** (表中有表): 表头 (表中第一个元素); 表尾 (表中除去表头剩下的部分)
13. **树**: 递归的, 元素之间有明显的层次关系
14. **完全二叉树** 应采用顺序存储结构, 一般二叉树则应采用链式存储结构
15. **二叉树的链式存储结构**: `typedef struct BiTnode{ int data; struct BiTnode *lchild, *rchild; }BiTnode, *BiTree;`
16. **二叉树的遍历**: 先序遍历 (先访问根节点), 中序遍历 (第二访问根节点), 后序遍历 (最后访问根节点), 层序遍历 (利用队列、每次出同一层的节点时进他们的子节点层)
17. **线索二叉树**: 加上线索 (直接前驱和直接后继) 的二叉树
18. **最优二叉树** (哈夫曼树): 一类带权路径长度最短的树
19. **树的存储结构**: 双亲表示法 (顺序存储); 孩子表示法 (链式存储); 孩子兄弟表示法 (链式存储, 两个指针分别为第一个孩子和下一个兄弟)
20. **图**: 一个节点的前驱节点和后继节点数目没有任何限制

21. 图的表示: $G=(V, E)$; V : 顶点的集合; E : 边的集合

22. 网: 边带权值的图

23. 图的相关概念



24. 图的存储结构: 邻接矩阵表示法, 邻接链表表示法

25. 图的遍历: 深度优先搜索, 广度优先搜索

26. 生成树: 极小连通子图, 针对连通图

27. 最小生成树 (权值和最小的生成树) 算法: 普尼姆算法 (在相邻边的基础上求最小, 与边数无关, 适于边稠密的网); 克鲁斯科尔算法 (在不构成环的基础上找最小边直至连通, 与顶点数无关, 适于边稀疏的网)

28. AOV 网: 有向图中顶点表示活动, 有向边表示活动间的优先关系

29. 拓扑排序: 将 AOV 网中所有顶点按优先顺序排成一个线性序列的过程

30. AOE 网: 有向图中有向边表示活动, 边上的权值表示该活动持续的时间

31. 关键路径: 从源点到汇点的路径中长度最长的

32. 最短路径: 从源点到其余各顶点的最短路径----迪杰斯克拉算法

33. 平均查找长度: 关键字和给定值进行过比较的记录个数的平均值

34. 静态查找方法: 顺序查找; 折半查找; 分块查找

35. 动态查找: 表结构本身在查找过程中是动态生成的

36. 二叉排序树: 左子树上所有节点的值小于根节点的值, 右子树上所有节点的值大于根节点的值

37. 平衡二叉树 (AVL 树): 左子树和右子树高度之差的绝对值不超过 1

38. B 树 (m 阶): 每个节点子树个数 $\leq m$, 根节点子树个数 = 0 或 ≥ 2 , 其他节点子树个数 = 0 或 $\geq m/2$

39. 哈希表: 通过哈希函数 (以记录的关键字为自变量) 得到记录的存储地址; 定长按一定函数规律存放数据; 哈希地址+关键字

40. 哈希表的重点: 构造哈希函数 (直接定址法, 数字分析法, 平方取中法, 折叠法, 随机数法, 除留余数法); 解决冲突 (开放定址法, 链地址法, 再哈希法)

41. 简单排序 (时间复杂度 $O(n^2)$, 空间复杂度 $O(1)$): 直接插入排序 (插入第 i 个时前 $i-1$ 个以排序好); 冒泡排序 (相邻两个比较排序, 每次循环确定一个极值); 简单选择排序 (第 i 个依次与后面每个元素比较排序, 每次循环确定一个极值, 不稳定)

42. 高端内部排序: 希尔排序 (先将整个序列分割成若干序列分别进行直接插入排序, 再对

整个序列进行一次直接插入排序, 不稳定); 快速排序 (将整个记录分割成独立的两部分, 两个指针分别指向对应部分的两端, 往中间移动比较排序, 递归, 不稳定); 堆排序 (建立初始堆输出并删除堆顶关键字, 再建立新堆得到新的关键字依次输出, 不稳定); 归并排序 (将若干个有序序列合并为新的有序序列); 基数排序 (按组成关键字的各个数位的值进行排序)

9、算法设计与分析

1. **算法的特性**: 有穷性, 确定性, 可行性, 输入, 输出
2. **算法的表示方法**: 自然语言, 流程图, 程序设计语言, 伪代码
3. **递归算法求时间复杂度**: 展开法 (将等式依次展开); 代换法 (用所猜测的值代替函数的解)
4. **分治法** (将一个难以直接解决的大问题分解成一些规模较小的小问题各个击破): 分解, 求解, 合并
5. **动态规划法** (将待求解问题分解成若干子问题分别求解, 再根据子问题的解得到原问题的解; 分解得到的子问题往往不是独立的): 找出最优解性质并刻画其结构特性; 递归地定义最优解的值; 以自底向上的方式求出最优值; 根据最优值构造最优解
6. **贪心算法** (仅根据当前已有信息做出选择, 重点考虑局部最优以达到全局最优): 最优子结构; 贪心选择性质
7. **回溯法** (以深度优先的方式系统地搜索问题): 定义问题的解空间; 确定解空间结构; 深度优先的方式搜索解空间
8. **分支限界法** (以广度优先或以最小耗费优先的方式搜索解空间): 队列式分支限界法; 优先队列式分支限界法
9. **概率算法** (把随机性的选择加入到算法中, 允许较小概率的错误来提高运行效率): 数值概率算法; 蒙特卡罗算法; 拉斯维加斯算法; 舍伍德算法
10. **近似算法** (放弃求最优解, 而用近似最优解代替) 衡量标准: 算法的时间复杂度; 解的近似程度
11. **NP 完全性理论**: 研究计算问题难易以及一类特殊的难解问题的理论
12. **P 类问题**: 能够以 $O(n^k)$ 的时间运行一个确定性算法得到准确答案
13. **NP 类问题**: 能够以 $O(n^k)$ 的时间运行一个不确定性算法得到准确答案
14. **NP 完全问题**: 能够证明用多项式时间的确定性算法得到准确答案

10、面向对象技术

1. 面向对象技术: 以客观世界的对象为中心, 其分析和设计思想符合人们的思维方式, 分析和设计的结果与客观世界的实际比较接近, 容易被人们所接受
2. 面向对象: 对象 (对象名、属性、操作), 分类 (所包含的方法和数据描述一组对象的共同行为和属性), 继承 (父类与子类之间共享数据和方法的机制), 通过消息的通信 (对象之间进行通信的一种构造)
3. 多态: 不同的对象收到同一消息产生完全不同的结果 (重载+重写+类属)
4. 绑定: 把过程调用和响应调用所需要执行的代码加以结合的过程

5. 动态绑定：当一个对象发送消息请求服务时，根据接受对象的具体情况将请求的操作与实现的方法进行连接
6. 面向对象程序设计语言应支持：被封装的对象；类和实例概念；类间的继承性；多态
7. 程序设计范型：人们在程序设计时所采用的基本方式模型；（+类+继承性=面向对象程序设计范型）
8. 面向对象的好处：解决了产品质量与生产率之间的权衡；使系统具有很高的灵活性和易扩充性；能管理复杂性并增强伸缩性；把某一领域分割成各种对象分析设计更合理；从概念模型化到分析、设计、编码可以实现无缝传递；通过封装进行信息隐蔽有助于建立安全的系统
9. 典型的面向对象设计语言：Smalltalk, Eiffel, C++, JAVA
10. 类的特征：同一个类的不同实例具有相同的数据结构和规律相同的行为；同一个类的不同实例可以持有不同的值因而可以具有不同的状态；实例的初始状态可以在实例化时确定
11. 消息传递：使对象具有交互能力的主要模型
12. 对象自身引用（this）：值使得方法体中引用的成员名与特定的对象相关；类型则决定了方法体被实际共享的范围
13. 虚拟函数：只给出方法的定义不给出实现过程，供其子类重置（overriding）
14. 类属：类的模版；参数多态机制；不限定成员的类型，具体对象再赋予具体类型
15. 类库：预先定义的程序库，以程序模块的形式按照类层次结构把一组类的定义和实现组织在一起
16. 抽象类：无实例的类，以供继承
17. 面向对象分析：认定对象；组织对象；描述对象间的相互作用；定义对象的操作；定义对象的内部信息
18. 面向对象设计：设计分析模型，实现相应源代码
19. 面向对象软件的测试：算法层，类层，模版层，系统层
20. 面向对象的方法：OOA 方法：主题层，对象类层，结构层，属性层，服务层；OOD 方法：问题域部件，人机交互部件，任务管理部件，数据管理部件；Booch 的 OOD 方法：标识类和对象，确定他们的含义，标识他们之间的关系，说明每一个类的界面和实现；OMT（面向对象建模技术）：对象模型，动态模型，功能模型---分析，系统设计，对象设计，实现；UML（统一建模语言）：事务（结构事务、行为事务、分组事务、注释事务），关系（依赖、关联、泛化、实现），图（类图、对象图、用例图、交互图、状态图、活动图、构建图、部署图）
21. 设计模式四要素：模式名称，问题，解决方案，效果
22. **设计模式分类**：创建型（抽象了实例化过程，帮助系统独立于如何创建、组合和表示它的那些对象，如 Singleton 模式：保证类只能有一个实例），结构型（采用继承机制灵活地组合类和对象以获得更大的结构），行为型（使用继承机制在类间分派行为，描述对象或类的模式以及其通信模式，如 Observer 模式：一旦模型发生变化所有视图都会得到通知）

-----P550

11、标准化和软件知识产权基础

1. 标准：对重复性事务和概念所做的统一规定
2. 标准化对象：标准化的具体对象（需要指定标准的具体事务）；标准化的总体对象（各种具体对象的全体所构成的整体）
3. 标准化活动过程：标准的制定，标准的实施，标准的更新

- 4.标准的分类：根据适用范围（国际标准，国家标准，区域标准，行业标准，企业标准，项目规范）；根据标准的性质（技术标准，管理标准，工作标准）；根据标准化的对象和作用（基础标准，产品标准，方法标准，安全标准，卫生标准，环境保护标准，服务标准）；根据法律的约束性（强制性标准，推荐性标准）
- 5.信息技术标准化：信息编码标准化（如 ASCII 码）；汉字编码标准化（如 gb2312）；软件工程标准化（基础标准，开发标准，文档标准，管理标准）
- 6.权威国际标准化组织：国际标准化组织（ISO）；国际电工委员会（IEC）
- 7.知识产权：人们基于自己的智力活动创造的成果和经营管理活动中的经验、知识而依法享有的权利
- 8.知识产权的分类：工业产权（专利、实用新型、工业品外观设计、商标、服务标记、厂商名称、产地标记、原产品名称、制止不正当竞争等）；著作权（版权）
- 9.知识产权的特点：无形性，双重性，确认性，独占性，地域性，时间性
- 10.计算机软件著作权受保护条件：独立创作，可被感知，逻辑合理
- 11.计算机软件著作权的权利：人身权（发表权，署名权）；著作财产权（使用权，复制权，修改权，发行权，翻译权，注释权，信息网络传播权，出租权，使用许可权，获得报酬权，转让权）
- 12.软件经济权利的许可使用：独占许可使用，独家许可使用，普通许可使用，法定许可使用，强制许可使用
- 13.软件著作权：谁开发，谁享有
- 14.计算机软件著作权特点：技术性，依赖性，多样性，运行性
- 15.商业秘密：部位公众所知的、能为权利人带来经济利益、具有实用性并经权利人采取保密措施的技术信息和经营信息；包括经营秘密和技术秘密
- 16.商业秘密的构成条件：未公开性，实用性，保密性
- 17.授予专利权的条件：新颖性，创造性，实用性
- 18.软件企业应建立的合同规范：劳动合同关系，软件开发合同，软件许可使用（或转让）合同

12 软件系统分析与设计

- 1.软件设计师应用性技术内容：结构化分析与设计；数据库分析与设计；面向对象分析与设计；算法分析与设计；面向过程的程序设计与实现；面向对象的程序设计与实现
- 2.结构化分析与设计：确定系统边界，画出系统环境图；自顶向下，画出各层数据流图；定义数据字典；定义加工说明；将图、字典以及加工组成分析模型
- 3.数据库分析与设计：数据库规划；需求描述与分析；数据库与应用程序设计；数据库系统实现；测试阶段；运行维护
- 4.数据库设计的方法：用户需求分析（调查机构情况，熟悉业务活动，明确用户需求，确定系统边界，分析系统功能，分析系统数据，编写分析报告）；概念设计（数据建模，E-R 方法）；逻辑设计（数据模式设计：层次模型、网状模型、关系模型）；物理设计（确定数据分布，确定数据的存储结构，确定数据的访问方式）
- 5.面向对象分析与设计的步骤：建模系统功能（确定参与者，确定需求用例，构造用例模型，记录需求用例描述）；定义领域模型（定义概念、属性和重要的关联）；定义交互、行为和状态（确定并分类用例设计类，确定类属性、行为和责任）；定义设计类图（会议，用户，会

议主题，稿件，审阅意见）

6. 算法设计过程：理解问题；确定相关因素（包括问题的输入与输出、用何种数据结构、用什么样的算法设计策略）；设计算法；证明算法的正确性；分析算法的效率；实现算法

7. 算法问题类型：查找问题；排序问题；图问题；组合问题（最优化问题）；几何问题（点、线、面、体）

8. 面向过程的程序设计与实现的核心：C 语言->指针

9. 面向对象的程序设计与实现的核心：设计模式