#### 例题均是考研真题或期末题出现过的。

### 数据结构

**南软的数据结构45分总的来说比较温和重视基础概念的理解，每年题型变化不一但是你如果研究了期末卷和历年真题就会发现还是重点突出有的来回考变着考例如递归，时间复杂度，树的遍历，散列表，堆排序及每种排序的特点，平衡树调整，二叉搜索树，图的最小生成算法，最短路径等。也有些每年都会出现的少的冷点，当然也有从来不考的比如KMP、图的存储以及外部排序。只要复习全面重点部分强化练习，重点算法类型好好准备没啥大的问题。**

#### 第1章：

数据，数据结构，基本类型，抽象数据类型的概念。（南软专业课复习一定要重视基础概念理解）

主要掌握Java语言的面向对象编程、**递归的概念与实现** 。

要求：• 主要能用递归思想写出算法（很重要算法题喜欢考复试机试也喜欢考）

例如

求数组中的最大值（数组实现和链表实现）

public static int findMax(int[] a, int n){

//n表示n个元素，它们在数组a中

if(n= =1){

return a [0];

}

else{

int temp=findMax(a,n-1);

return temp>a [n-1]?temp:a [n-1];

}

}

int max(int a[],int n)

{ if(n = = 1) return a[0];

int m = max(a,n-1);

if( m > a[n-1] )

return m;

else

return a[n-1];

}

例2. 求数组中的最大值

如果用链表来实现表：

求链表中的最大值

int GetMaxInt( ListNode f )

{ if( f.link = = NULL ) return f .data ;

else

{ int i = GetMaxInt( f.link );

if ( i > f .data ) return i ;

else return f .data ;

}

}

或 else return ( f . data) > ( GetMaxInt( f . link ) ) ? f . data :

GetMaxInt( f . link )

#### 第2章 算法分析

最佳、最差和平均情况下的复杂度差异;

大O、Ω和 θ 符号

1）分析某个语句的执行次数（频度）

2）分析某个程序段执行的时间复杂度（用大O表示，要求写出推导过程）

例：求以下算法时间复杂度

int x = 91; int y = 100;

while(y>0)

{ if(x>100) { x -= 10; y--; }

else x++;

}

1100次

#### 第3章 表、栈和队列

表、栈和队列的（基本概念，顺序存储结构，链式存储结构，应用），

表：

逻辑-----（e1

, e2

, …..en

)

物理------数组实现

**链表实现------单链表**

循环链表

双向链表

cursor

表头结点

**用链表实现**

操作------查找、插入、删除等

----多项式相加

约瑟夫问题

**双链表的插入、删除**

例题----逆转链表等题（今年机试第一题类似的）

public void inverse( ListNode f )

{ if ( f = = NULL ) return;

ListNode p = f . link ; pr = NULL;

while ( p ! = NULL )

{ f . link = pr ;

pr = f ;

f = p ;

p = p . link ;

}

f . link = pr ;

}

**栈、队列**

定义-----栈的定义， 队列的定义

机内实现------数组 （循环队列）

单链表

应用

栈-----对表达式求值。中缀----后缀----对后缀表达式求值

递归函数的实现。

第4章中用非递归实现中序,后序遍历(在第4章中讲)

队列---循环队列的补充题：已知队尾元素的位置与元素的个数，

求队头元素的位置。

中缀到后缀：

(a+b)\*((c-d)/2\*e)----- ab+cd-2/e\*\*

用了什么栈？

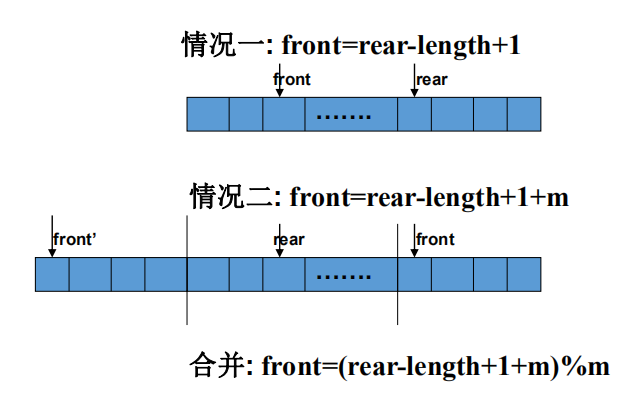
对后缀表达式求值：

用了什么栈？

队列---循环队列的补充题

**已知队尾元素的位置与元素的个数， 求队头元素的位置。（很重要后面经常涉及到）**

先用实例来分析，然后归结到一般情况。



**几种特殊矩阵及其压缩**

习题：（常考弄清楚）

设有一个n\*n的对称矩阵A。为了节约存储，可以只存对角线及对角线以上的元素，或者只存对角线或对角线以下的元素。前者称为上三角矩阵，后者称为下三角矩阵。我们把它们按行存放于一个一维数组B中，并称之为对称矩阵A的压缩存储方式。试问：

1）存放对称矩阵A上三角部分或下三角部分的一维数组B有多少元素？

2）若在一维数组B中从0号位置开始存放，则如图(a)所示的对称矩阵中的任一元素aij在只存上三角部分的情形下(图(b))应存于一维数组的什么下标位置？给出计算公式。

3）若在一维数组B中从0号位置开始存放，则如图(a)所示的对称矩阵中的任一元素aij在只存下三角部分的情况下\*(图(c))应存于一维数组的什么下标位置？给出计算公式。

答案：

1) 1+2+3+…+n = ½\*(1+n)\*n

2) loc(A[i,j] ) = loc(B[0]) + ( n+n-1+….+n-i+2 + j-i )

t = ½ \*(2\*n-i+1)\*i + j-i i<=j

t = ½ \*(2\*n-j+1)\*j + i-j i>j

3) loc(A[i,j] = loc(B[0]) + (1+2+3+….+i-1+j-1)

t = ½ \*i\*(i+1) + j i>=j

t = ½ \*j\*(j+1) + i i<j

对角线元素的地址：t = i\*(i+3)/2

#### 第4章 树

1.二叉树的定义、性质

2.满二叉树与完全二叉树的概念

3.二叉树的机内存储：数组表示（完全二叉树）、左---右拉链表示、cursor

4.先序、中序、后序遍历（递归非递归）

层次遍历-----用到队列

5. 利用先序、中序可唯一构造一棵树

先序：ABDCEGFHI

中序：DBAEGCHFI

利用中序、后序可唯一构造一棵树

手工画出一棵树

利用算法生成一棵树

\*6. 利用广义表表示来构造一棵树

7. 应用

**树的机内表示：**

**广义表表示、双亲表示、左子女---右兄弟表示**

**树-----二叉树的转换**

**树与森林的遍历**

**树的遍历：深度优先遍历，广度优先遍历**

**• 深度优先遍历**

先序次序遍历（先序）

访问树的根 按先序遍历根的第一棵子树，第二棵子树，

……等。

**后序次序遍历（后序）**

按后序遍历根的第一棵子树，第二棵子树，……等

访问树的根。

**.广度优先遍历**

**森林的遍历**

**深度优先遍历**

**\* 先根次序遍历（对应二叉树的先序）**

访问F的第一棵树的根

按先根遍历第一棵树的子树森林

按先根遍历其它树组成的森林

**\* 中根次序遍历（对应二叉树的中序）**

按中根遍历第一棵树的子树森林

访问F的第一棵树的根

按中根遍历其它树组成的森林

**\* 后根次序遍历（对应二叉树的后序）**

按后根遍历第一棵树的子树森林

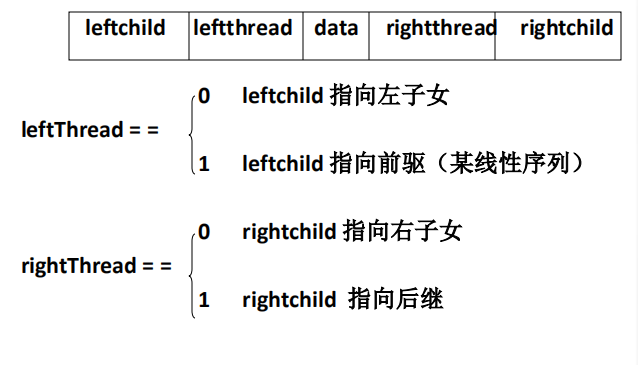
按后根遍历其它树组成的森林

访问F的第一棵树的根

**广度优先遍历（层次遍历）**

线索树

Thread Tree

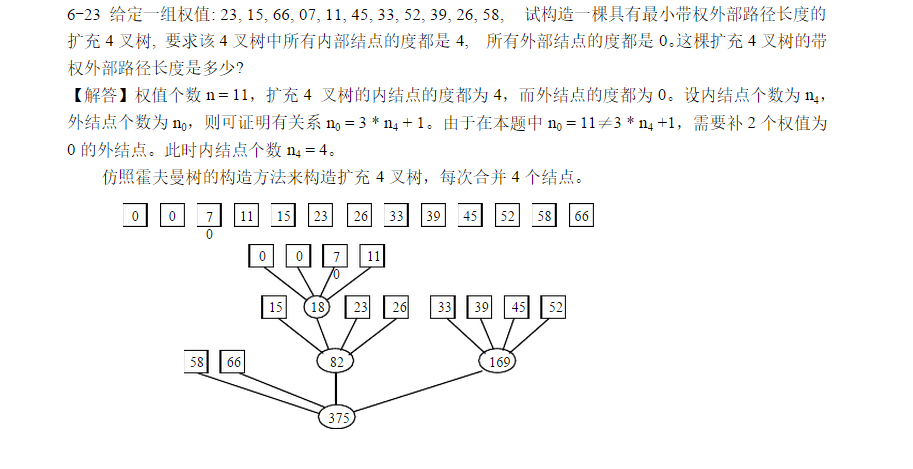


8. 哈夫曼树

哈夫曼树的构造

哈夫曼编码

扩充的二叉、三叉、….、t叉树



15, 3, 14, 2, 6, 9, 16, 17 构造扩充的三叉树。

9. 等价类问题

第8章

##### 第4.1章：二叉搜索树

1.二叉搜索树的概念

2.带索引的二叉搜索树的概念

**3. AVL树-----平衡的二叉搜索树，主要是插入删除（十分重要）**

**4. B-树插入删除**

**二叉搜索树主要操作：**

**查找、插入、删除**

写一递归函数实现在带索引的二叉搜索树（IndexBST)中查找第k个小

的元素。（这是道重要的算法题）

public Comparable findK( BinaryNode root, int k)

{

if( root==null) return null;//空

if( k< root. leftSize) //在左子树

findK( root. left, k);

else if( k>root. leftSize) //在右子树

findK( root. right, k-root. leftSize);//注意减去

else return root.element;

}

#### 第5章：散列

1.散列函数的选择

2.解决冲突的方法

开地址法：线性探查法

平方探查法

二次散列

链地址法

第6章：优先队列

1.优先队列的概念

2.优先队列的实现

用无序的线性表来实现

用堆来实现----堆的定义

初始化一个堆

**堆排序（重要常考）**

#### 第7章：排序

• 各种排序方法的算法思想与时间复杂度的分析

1.排序的有关概念

内排序：对内存中的n个对象进行排序。

外排序：内存放不下，还要使用外存的排序。

排序算法的稳定性：

如果待排序的对象序列中，含有多个关键码值相等的对象，用某种方法排序后，这些对象的相对次序不变的，则是稳定的, 否则为不稳定的

2.插入排序(直接插入排序，二分法插入排序，shell排序)

**3.交换排序(起泡排序，快速排序）**

4.选择排序（直接选择排序，堆排序）

5.归并排序

6.基数排序

例题---在O(n)时间内实现将负数排在所有非负数之前。（重要2016年真题）

void sort ( float [ ] a, int n )

{ int i = 0 , j = n-1 ;

while ( i != j )

{ while ( a[j] >= 0.0 && i < j ) j-- ;

while ( a[i] < 0 && i < j ) i++ ;

float temp = a[i] ; a[i] = a[j]; a[j] = temp;

j-- ; i++ ;

}

}

#### 第9章：图

1.无向图、有向图的有关概念

2.图的机内存储

邻接矩阵

邻接表

3.图的若干算法

1) 图的遍历----DFS

BFS

**2) 最小代价生成树---Prime算法**

**Kuscal算法（考过真题）**

**3) 最短路径-----Dijkstra算法（去年刚考过）**

**Floyed算法（常考）**

4) 活动网络----AOV——拓扑排序

AOE——关键路径（期末题考过真题没考过今年可能会考）

### 操作系统

**南软的操作系统35分形式每年都较为统一，每年出一点前一年没有设计的考点不过还是比较稳定的，知识点固定。有的重点年年都考比如索引分级存储，进程模型转换，页面调度算法，PV操作，磁盘调度算法，银行家算法。有的每年换着考，分页分段存储进程调度。但是通过一定有针对性的练习着重基础概念拿下问题不大，有的会涉及到教材中的一些linux等部分不过总体偏少，不必过于慌张。**

计算机专业考研全国统考《操作系统》考查目标

1. 了解操作系统在计算机系统中的作用、地位、发展和特点。

2. 理解操作系统的基本概念、原理，掌握操作系统设计方法与实现技术。

3. 能够运用所学的操作系统原理、方法与技术分析问题和解决问题。

**明确计算机操作系统的作用与功能**

 掌握操作系统实现的基本原理与方法

 在微观上，掌握设计实现各个操作系统**模块的方法、策略与算法**

 在宏观上，掌握操作系统的结构和设计实现方法，进一步了解大型软件系统的结构和设计实现方法

掌握**并发程序设计**的基本方法

#### 一、操作系统概述



操作系统的概念、特征、功能和提供的服务

特权指令和处理器状态（**管态和目态**）

**中断（很重要的概念）**与指令周期

操作系统的发展与分类

概念: 分时操作系统、实时操作系统，批处理操作系统

操作系统结构分类1. 单体式结构2. 层次式结构3. 虚拟机结构4. 微内核结构5. 客户/服务

操作系统的运行环境

#### 二、进程管理

(一)进程与线程（考名词解释区别）

1. 进程概念

**2. 进程的状态与转换 （重点经常考三状态七状态）**

3. 进程控制

4. 进程组织 (进程的队列)

**5. 进程通信 (共享存储系统；消息传递系统；管道通信爱考名词解释)**

6. 线程概念与多线程模型

**掌握线程的三种实现模型（用户级内核级）**

了解Windows的进程和线程管理

掌握Solaris的进程和线程管理模型

(二)处理机调度

**1.调度的基本概念 （名词解释）**

2.调度时机、切换与过程

3.调度的基本准则

4.调度方式抢占与非抢占

**5.典型调度算法 （必考大题）**

 FCFS (先来先服务)

 RR (时间片轮转)

 SPN (最短进程优先)

 SRT (最短剩余时间优先)

 HRRF (最高响应比优先)

 Feedback (多级反馈调度)

1. 进程同步的基本概念 (理解)

**2. 实现临界区互斥的基本方法 (重要)**

– 软件实现方法；硬件实现方法。

1. 了解程序的并发性与并发程序设计

2. 掌握临界区互斥及其解决方案

3. 熟练使用PV进行程序设计

4. 掌握Hoare管程

5. 掌握消息传递

3. 信号量

4. 管程 (没时间可以不看)

5. **经典同步问题 (必考一题)**

– 生产者-消费者问题；读者-写者问题；哲学家进餐问题。

(四) 死锁

**1. 死锁的概念（重要）**

**2. 死锁处理策略**

**3. 死锁预防**

**4. 死锁避免（必考）**

**– 系统安全状态：银行家算法。**

1. 死锁检测和解除

#### 三、内存管理

(一)内存管理基础

1. 内存管理概念

– 程序装入与链接；逻辑地址与物理地址（理解）空间；内存保护。

2. 交换与覆盖

3. 连续分配管理方式（理解）

– 单一连续分配；分区分配。 固定分区  动态分区  伙伴系统

1. **非连续分配管理方式:分页管理方式；分段管理方式；段页式管理方式。（重要）（页表、多级页表、反置页表）**

(二) 虚拟内存管理

**1. 虚拟内存基本概念（重要常考）**

2. 请求分页管理方式

**3. 页面置换算法 : 最佳置换算法(OPT)；先进先出置换算法(FIFO)；最近最少使用置换算法(LRU)；时钟置换算法(CLOCK)。（必考一题）**

4. 页面分配策略

5. 抖动 : 抖动现象；工作集。

6. 请求分段管理方式

7. 请求段页式管理方式

#### 四、文件管理

(一) 文件系统基础

1. 文件概念

2. **文件结构**

**– 顺序文件；索引文件；索引顺序文件。（常考）**

3. 目录结构

– **文件控制块和索引节点（必考一题大题）**；单级目录结构和两级目录结构；树形目录结构；图形目录结构。

4. 文件共享

– 共享动机；共享方式；共享语义。

5. **文件保护（常考名词解释）**

– 访问类型；访问控制。

(二) 文件系统基础

1. 文件系统实现

2. 文件系统层次结构

3. 目录实现

4. 文件实现

(三) 磁盘组织与管理

1. 磁盘的结构

**2. 磁盘调度算法（常考重要）**

3. 磁盘的管理

#### 五、输入输出(I/O)管理

(一) 输入输出(I/O)管理

1. I/O管理概述

2. I/O设备

3. I/O管理目标

4. I/O管理功能

5. I/O应用接口

**6. I/O控制方式或者I/O系统各层软件及其功能（常出简答）**

(二) I/O核心子系统

**1. I/O调度概念**

2. 高速缓存与缓冲区

3. 设备分配与回收

**4. 假脱机技术(SPOOLing)（经常考重理解）**

5. 出错处理

#### 计算题型分析

 **多道程序设计**

** CPU调度算法**

** 死锁避免银行家算法 ，死锁检测**

** 连续分配，分区分配：适配算法，伙伴系统**

** 地址转换计算：分页管理方式；分段管理方式。**

** 页面置换算法 :**

** 抖动现象，工作集**

** 磁盘调度算法**

** 文件系统的计算**

** PV操作、管程**

### 计算机网络

**网络部分25分它的复习没必要搞得那么细致，因为一共就这么多分，重点又很明显，不可能出偏到哪里去，也不可能出现所谓的全出偏的略过真正的重点。而考题出的往往也比我们预料的简单。南大的网络依然是重视基础概念的理解，第一点强调的就是要从宏观上去把握网络的结构层次如几种模型，涉及到具体协议的一些层次结构。第二微观上就是各种协议概念工作原理甚至优缺点，各种设备和使用到的方法，以网络层和传输层为重中之重。第三就是涉及子网划分，编码，CIDR，数据报传输，路由转存的一些大题。复习全面把握重点协议并宏观理解不会有什么问题。**

#### 一 计算机网络体系结构

1.1概述

计网的概念，组成，功能和**分类，性能指标；**

1.2 结构

SDU，PDU，协议定义及三要素，服务及分类，**OSI参考模型及各层含义，TCPIP参考模型及各层含义。**

**如： 真题** 写出五层的网络模型以及各层的功能  (5分）

TCP/IP有哪几层？请简述每层的功能。

#### 二 物理层

**2.1一些名称的概念比较重要的如信道，信号，带宽，码元**

**2.2**奈式准则，香农定理

**2.3**编码与调制**，重点掌握编码中的曼切斯特编码和差分曼切斯特**

**2.4**电路交换，报文交换和分组交换的比较，数据报服务与虚电路服务的比较

**2.5**传输介质，**重点掌握双绞线同轴电缆光纤无线的特点。**物理层四大特性

**2.6**中继器和集线器的特点和应用**。**

#### 三 数据链路层

**3.1**链路层的功能，组帧，差错控制，**流量控制滑动窗口基本原理和可靠传输机制大致了解**

**3.2信道划分;时分多路复用，波分多路复用，码分多路复用的概念和基本原理了解**

**3.3CSMA|CD协议**

**3.4局域网的基本概念和体系结构；以太网几个标准**

**3.5**广域网的基本概念，**ppp协议**

**3.6**网桥交换机的基本原理

#### 四 网络层（重要）

4.1网络层的功能；**路由与转发（会出大题）**；拥塞控制。

4.2路由算法与路由协议：**静态路由动态路由；距离向量算法；链路状态路由算法**

**RIP协议；OSPF协议；（基本年年考）**

自治系统,BGP路由协议

**如真题**

4.21 距离矢量协议中，写出两种避免回路发生的技术。 4分

4.22 给出B和C两个路由器的路由表，让你写出B路由器收到C路由表，更新后 的路由表。5分

4.23 描述单域OSPF的工作流程 (5分）

4.24 比较 RIP 和 OSPF 的区别

4.25 请分析静态路由适用于那些场景，给出原因。

4.3 IPv4：

**分组地址（要详细）与NAT（去年刚考过）；子网划分与子网掩码（经常出大题），CIDR路由聚集（期末考过真题没考过），ARP协议，DHCP协议与ICMP协议，**IPV6不怎么考要了解。

4.4 IP组播，移动IP了解即可

4.5路由器工作原理

#### 五 运输层

* 1. 传输层的功能，端口，**端口号**，套接字的概念
  2. **UDP协议，UDP数据报的格式和**校验
  3. **TCP协议，TCP报文段各个部分及其含义**
  4. **TCP连接三次握手原则（去年考了非常重要）**
  5. **TCP的流量控制，拥塞控制，可靠传输。（了解）**
  6. **如真题**：写出TCP、UDP的差别 4分  
      简述三次握手过程并说明为什么要进行第三次握手？

#### 六 应用层

6.1 网络应用模型：客户服务器模型；P2P模型

6.2 DNS主要是域名解析过程理解

6.3 FTP工作原理

6.4 电子邮件涉及到的SMTP，POP3等

6.5 万维网www，包括HTTP，URL，HTML的概念

#### 七 一些重要的名词解释

* 1. **Full duplex**
  2. **OSI reference model**
  3. **Split horizon**
  4. **CSMA/CD**
  5. **DNS**
  6. **Time division multiplexing（TDM）**
  7. **FDM**
  8. **STP**
  9. **RARP**
  10. **DHCP**
  11. **OSPF**
  12. **UTP**
  13. **Socket**
  14. **ICMP**
  15. **SMTP**
  16. **ARP**
  17. **HTTP**
  18. **TCP/IP**
  19. **RIP**
  20. **NAT**
  21. **FTP**
  22. **split horizon**
  23. **Time Division Multiplexing**
  24. **PPP**

### 软件工程与计算

**软件工程这门课所涉及到的东西及其广泛，考试形式也变化多样让人难以预料。能考的东西很多所要复习的范围也广，但是有一章感觉真题中重来没缺席过那就是详细设计中的模块化面向对象的信息隐藏和设计模式，主要包括几大原则和设计模式，基本上都会以各种各样的形式涉及到，其实也更好预料到的形式基本就是判别代码不好的地方并给出修改，并每年放在不同的情境下。然后比较爱考的就是一些重要名词的解释其中最显眼要属软件工程的定义需求工程的定义等等，然后就是需求分类，各种uml图应用（主要是类图用例图顺序图），体系结构，面向对象的一些原则部分，测试部分还考过契约编程防御设计，质量保障。期末题中还比较爱考内聚耦合类型，人机交互原则，黑盒白盒测试，软件演化模型，需求分析，测试思路，几个开发模型等等。复习抓住对几个核心部分的理解书多看几遍，理解透彻才好做题，往年题研究透彻，自己上网找找相关题材的博客多思考多总结上了考场能写拿多少是多少，所谓会者不难难者不会。**

#### 一 软件工程基础

1. **软件工程的定义**

#### 二 软件工程的发展稍作了解

#### 四 项目启动

4.1 管理团队：（团队结构有哪几种 在实验中采取了哪些办法 有哪些经验，团队建设）

4.2 **软件质量保障：（有哪些措施 结合实验进行说明：测试和评审）常考**

4.3 软件质量包括：功能性 可靠性 易用性 效率 可维护性 可移植性

4.4 **软件配置管理：（配置管理有哪些活动 实验中是如何进行配置管理的 结合实验，说明一个项目的质量保障包括哪些活动）**考过13选择题

4.5 变更控制过程？项目管理活动有哪些？

五 软件需求

5.1 **需求定义，需求工程定义**

5.2 需求获取方式，分析验证管理

5.3 **需求的层次性及实例（常考）**

5.4 **需求分类（重要）**

#### 六 需求分析方法

6.1结构化分析：数据流图，实体关系图

6.2 面向对象分析：统一建模语言（**类图，用例图，交互图又叫顺序图**，状态图和对象约束语言）

#### 七 需求文档化与验证

7.1 软件需求规格文档组成

7.2 文档化的原因，需要需求规格说明的原因。

7.3 **对给定的需求示例设计功能测试用例： 以需求列表为线索，可以开发测试用例套件。覆盖正常和异常流程，验证需求的几个方法。**

7.4 技术文档要点：简洁、精确、易读（查询）、易修改。

需求书写要点：使用用户术语、可验证、可行性。

#### 八 软件设计基础

8.1 软件设计的核心思想：分解和抽象。

8.2 **软件设计定义**

8.3 **软件工程设计的三个层次和主要思想：高层、中层、低层设计**。

8.4 软件设计过程中的4个主要活动，软件设计的方法。

#### 九 软件体系结构基础

9.1 软件体系结构定义，部件，连接件

**9.2** **几种体系结构风格区别及其优缺点（重要）**

#### 十 体系结构设计与构建

10.1 体系结构设计的过程

10.2 包的创建原则

10.3 **体系结构构件之间接口的定义：**

提供的服务（供接口）：语法、前置条件、后置条件

需要的服务（需接口）：服务名、服务

**10.4 体系结构开发集成测试用例：（几种集成策略定义，桩Stub、驱动Driver）**

10.5 **体系结构评审方法**

#### 十一 人机交互评审 （常考选择题）

**11.1 可用性（易用性）**

**11.2 交互设计原则**

11.3 人机交互设计的人类因素（精神模型、差异性） 计算机因素

11.4 人机交互设计的交互性：（导航、反馈、协作式设计）

#### 十二 详细设计的基础

12.1 详细设计出发点：软件详细设计应该在软件体系结构设计之后进行，以需求开发的结果和软件体系结构的结果为出发点。 **后面一个面向对象设计的思想和过程。**

12.2 **职责分配、协作和控制风格：1）通过职责建立静态模型 2）通过协作建立动态模型**

**职责： 1）抽象对象的职责（属性职责和方法职责）**

**2）抽象类之间的关系（依赖、关联、聚合、组合、继承）（类图 P204）**

**3）添加辅助类**

**协作： 1）抽象对象之间的协作（顺序图、状态图 P207）**

**2）明确对象的创建**

**3）选择合适的控制风格：集中式、委托式、分散式（P209-210）**

**给定分析类图、系统顺序图和设计因素描述建立设计类图或详细设计图**

12.3 协作的测试：（Mock Object）

### 13-16章属于这本书最核心最重要的章节一定要好好看看。

#### 十三 详细设计中的模块化与信息隐藏

13.1 **耦合：描述了两个模块之间的复杂程度。**

**内聚：表达了一个模块内部联系的紧密型。**

**模块化的概念，信息隐藏的概念作用。**

13.2 **耦合和内聚的判断：**（根据例子说明他们之间的耦合程度/内聚，并给出理由）

耦合：（高→低）内容耦合、公共耦合、重复耦合、控制耦合、印记耦合、数据耦合。（P219）耦合越高越不利于软件变更。

内聚：（高→低）信息内聚、功能内聚、通信内聚、过程内聚、时间内聚、逻辑内聚、偶然内聚。（P220）内聚越低越不易实现变更。

13.3 **信息隐藏：（根据例子说明其信息隐藏程度的好坏）**

基本思想：每个模块隐藏一个重要的设计决策，抽象出类的关键细节（职责）。即抽象出接口，隐藏实现。

13.4 两种常见的决策：职责的实现、实现的变更。

（模块说明？主要秘密、次要秘密、角色、对外接口）

#### 十四 详细设计中面向对象方法下的模块化

14.1 模块化原则：（Principles from Modularization）好好看不理解的去找资料理解

**1）Global Variables Consider Harmful 全局变量考虑有害**

**2）To be Explicit 要明确**

**3）Do not Repeat 不要重复**

**（降低访问耦合的方法）**

**4）Programming to Interface 针对接口编程**

**5）Interface Segregation Principle（ISP） 接口分离原则**

**6）Demeter Law 迪米特法则（P232）**

**（降低继承耦合的方法）**

**7）Liskov Substitution Principle（LSP） Liskov替换原则（P235）**

**8）Favor Composition Over Inheritance 使用组合代替继承**

**（提高内聚的方法）**

**9）Single Responsibility Principle（SRP） 单一职责原则**

**10）集中信息与行为**

14.2 面向对象中的模块，耦合与内聚的度量了解下

十五 详细设计中面向对象方法下的信息隐藏

15.1 信息隐藏的含义：1）封装类的职责，隐藏职责的实现 2）预计将会发生的变更，抽象它的接口，隐藏它的内部机制

**封装的概念：1）将数据和行为同时包含在类中 2）分离对外接口和内部实现**

**多态的概念实现**

15.2 OCP： **Open Close Principle 开闭原则**

好的设计应该对“扩展”开放，好的设计应该对“修改”关闭。即发生变更时，好的设计只需要添加新的代码就能实现变更。

15.3 DIP： **Dependency Inversion Principle 依赖倒置原则**

抽象不应该依赖于细节，高层模块不依赖于低层模块，都应依赖于抽象

#### 十六 详细设计中的设计模式

16.1 提高可修改性：（如何提高可修改性、可扩展性、灵活性）

包括狭义的可修改性、可扩展性和灵活性。

方法： 需要能够将接口和实现分离。

Java等面向对象语言中的实现方式：

1）通过接口和实现该接口的类完成接口和实现的分离。

2）通过子类继承父类将父类的接口，将父类的接口和子类的实现相分离。

**16.2 设计模式（Design Pattern）：（给定场景，应用设计模式写出代码 给出代码用设计模式改写） 这本书核心部分**

**策略模式**： （**常考**）

抽象Strategy，实现不同的ConcreteStrategy方法，Context拥有Strategy的一个引用。

**抽象工厂模式**：

AbstractFactory声明接口，ConcreteFactory实现对产品的创建。

AbstractProduct定义产品接口，ConcreteProduct实现具体工厂创建出来的产品，实现接口。

Client使用AbstractFactory和AbstractProduct来创建。

**单件模式**：

Singleton提供访问单件的接口，负责实现单件。

Client通过getInstance使用单件。

**迭代器模式：**

Iterator定义访问和遍历的接口，ConcreteIterator实现接口。

Aggregate定义创建相应迭代器对象的接口，ConcreteAggregate实现接口。

#### 十七 软件构造

17.1 软件构造；**重构； 测试驱动并发； 结对编程**的几个概念。

17.2 构造包含的活动：详细设计、编程、测试、调试、代码评审、集成与构建、构造管理。

#### 十八 代码设计

18.1 代码设计：（给定代码段示例，对其进行改进或者发现其中的问题考过选择题）

**易读性：**

1）格式：使用缩进和对齐、将相关逻辑组织在一起、使用空行分割逻辑、语句分行。

2）变量命名：惯例和规则。

3）注释：文档注释。

**易维护性：**

1）小型任务：分解为多个高内聚、低耦合的小型任务。

2）复杂决策：使用布尔变量决策、有意义的名称来封装决策、表驱动编程。

3）数据使用。

4）明确依赖关系。

**考虑：**

1）简洁性、可维护性。

2）使用数据结构消减复杂判定。

3）控制结构、变量使用、语句处理。

4）How to write unmaintainable code

5）防御与错误处理。

18.2 设计可靠的代码：**契约式设计包括了异常和断言。（P309）、防御式编程：（P310） ，表驱动编程：（P307） 单元测试用例设计：（P313）**

使用辅助模型：**决策表、伪代码、流程图**。

18.3 代码复杂度度量了解下

#### 十九 软件测试

19.1 （**测试层次的划分？分别根据测试对象测试目标划分，测试的功能目标？验证和确认的区别，随机测试的概念，了解一些特定的测试技术，测试有哪些活动，如何度量？**）

19.2 **白盒测试和黑盒测试：（给出一个场景，判断应该使用哪种测试方法，如何去写）**

**17年考过**

黑盒测试：完全基于输入和输出来判断测试对象的正确性

常见方法：1）等价类划分 2）边界值分析 3）决策表 4）状态转换

白盒测试：（能解释并区别白盒测试三种不同的方法）

常见方法：

1）语句覆盖：保证每一行代码都至少执行一次。

2）分支覆盖：保证每个判断结果都至少满足一次。

3）路径覆盖：保证每条独立的执行路径都至少执行一次。

19.3 **给出场景和要求的测试方法。设计测试用例**

1.给出功能需求，要求写功能测试用例

2.给出设计图，要求写集成测试用例，Stub and Driver

3.给出方法描述，要求写单元测试用例，Mock Object

4.Junit的基本使用方法

#### 二十 软件交付

20.1 了解安装部署，培训和**文档支持，**项目评价**。**

#### 二十一 软件维护和演化

31.1 了解软件维护类型困难，如何理解软件维护的重要性？**开发可维护软件的方法与活动？**了解软件演化定律**，熟悉演化式生命周期模型**

31.2 软件维护演化常用技术：**遗留软件类型，逆向工程和再工程的定义应用。**

二十二、二十三 软件开发过程模型以及职业基础

**22.1 软件生命周期模型：需求工程→软件设计→软件实现→软件测试→软件交付→软件维护。 主体框架**

**22.2 对给定的场景，判定适⽤的开发过程模型：**

**构建-修复模型（Build-Fix Model）：**

**瀑布模型（Waterfall Model）**

**增量迭代模型（Incremental Model）**

**演化模型（Evolutionary Development）**

**原型模型（Prototyping）**

**螺旋模型（Spiral Model）**

**敏捷开发（Rational 过程）**

**这几个模型要详细了解定义考研专业课面试也会经常问到**