希望
Beijing Hope Electronic Press
www.bhp.com.cn

2004年新大纲编写

全国计算机等级考试二级C++考试辅导与试题集

王 丰 徐翠霞 主编

魏爱敏 于任霞 刘宗国 副主编

兵器工业出版社

北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

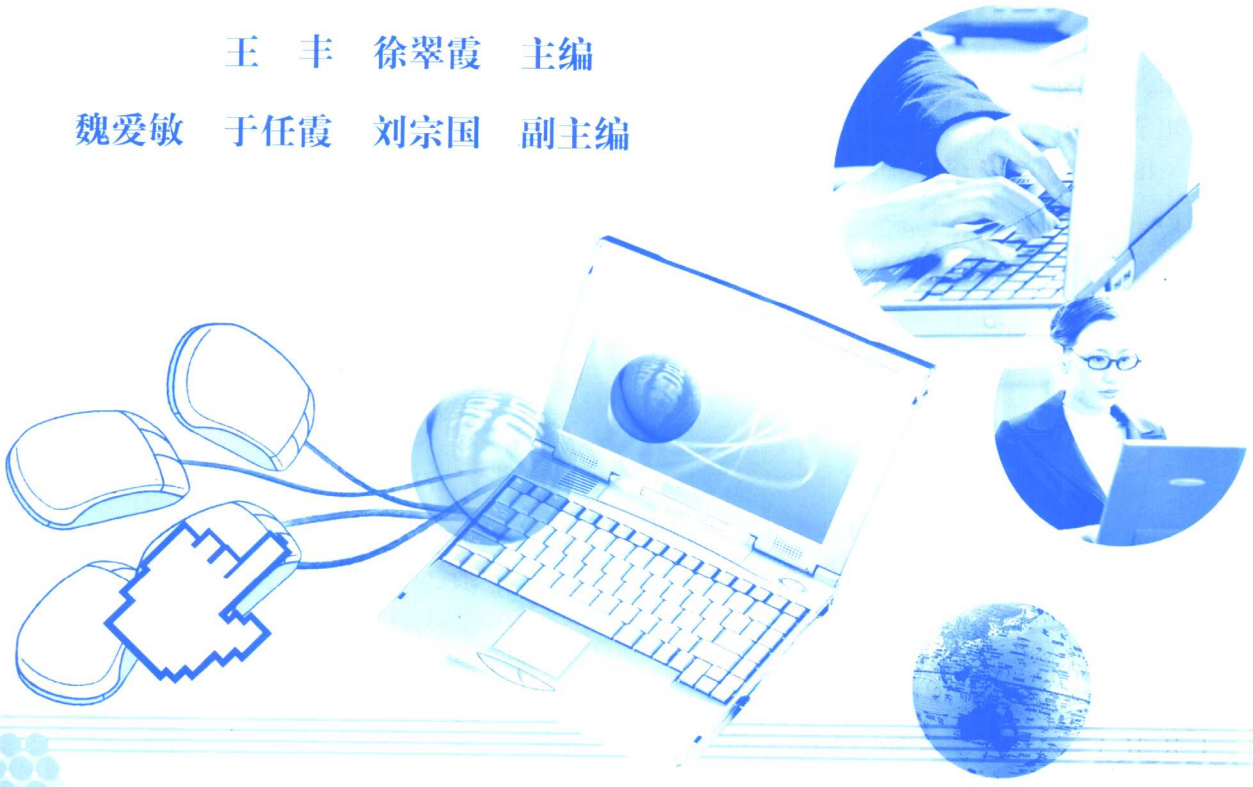
希望
Beijing Hope Electronic Press
www.bhp.com.cn

2004年新大纲编写

全国计算机等级考试二级C++考试辅导与试题集

王 丰 徐翠霞 主编

魏爱敏 于任霞 刘宗国 副主编



兵器工业出版社

北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

本书是全国计算机等级考试二级 C++的辅导教材，书中内容以 2004 年启用的新大纲为基准，包含二级公共基础、二级 C++笔试和二级 C++上机 3 大部分内容。全书共 13 章，分别为基础知识、C++语言基础、数据类型与表达式、基本程序设计、函数、数组与字符串、指针与引用、类与对象、继承、运算符重载与模板、输入/输出流、上机和全真模拟题。

本书内容的组织力求深入浅出，主要目的在于使考生迅速掌握二级 C++考点。每一部分由考点概述、难点剖析、经典例题、同步练习和同步练习答案 5 部分组成，考生可由此迅速掌握考试重点并得到强化训练。

本书的主要读者对象为参加全国计算机等级考试二级 C++的考生，另外也可供 C++语言自学者参考。

图书在版编目（CIP）数据

全国计算机等级考试二级 C++考试辅导与试题集 / 王丰，徐翠霞主编. —北京：兵器工业出版社；北京希望电子出版社，2005.5
ISBN 7-80172-365-1

Ⅰ.全… Ⅱ. ①王…②徐… Ⅲ.C 语言—程序设计—水平考试—自学参考资料 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2005）第 011641 号

出　　版：兵器工业出版社 北京希望电子出版社
邮编社址：100089 北京市海淀区车道沟 10 号
100085 北京市海淀区上地信息产业基地 3 街 9 号 金隅嘉华大厦 C 座 610
发　　行：北京希望电子出版社
电　　话：（010）82702660（发行）（010）62541992（门市）
经　　销：各地新华书店 软件连锁店
印　　刷：北京双青印刷厂
版　　次：2005 年 5 月第 1 版第 1 次印刷

封面设计：梁运丽
责任编辑：王　琦　宋丽华　刘海芳
责任校对：叶　子
开　　本：787×1092　1/16
印　　张：16.125
印　　数：1-5000
字　　数：373 千字
定　　价：26.00 元

前　言

为了使广大考生能够轻松愉快地通过全国计算机等级考试，我们编写了这本《全国计算机等级考试—— 二级 C++考试辅导与试题集》。本书根据教育部考试中心 2004 年颁布的计算机等级考试二级 C++语言程序设计考试大纲的要求编写，是集教师教学、学生自学、考前系统复习于一体的新思维教材，是《全国计算机等级考试大纲（2004 年版）》的配套用书。

新的全国计算机等级考试大纲于 2005 年正式实施，与原大纲相比，变化较大。对二级而言主要表现在：

（1）新增考试科目 Java、Access、C++并停考了 Fortran 等科目。

（2）大幅度改变了公共基础部分，删去了已过时的 DOS 部分，新增了数据结构、数据库、软件工程、程序设计语言等内容。

C++语言博大精深，知识点很多，再加上公共基础部分 4 门课的内容以及机试内容，摆在考生面前的问题是如何在短时间内迅速掌握众多的考试内容。本书提供的策略是：抽取重点、释疑难点、练习巩固和测试强化。全书内容的选取完全依据大纲，文字讲解和试题的深度、难度均以实际考试为基准，并兼顾了课堂教学和学生考前系统自学或复习的需要，每章均由考点概述、难点剖析、经典例题、同步练习、同步练习答案 5 个部分组成。

考点概述：按照大纲所要求的内容，对所考查的知识点进行了总结、归纳。

难点剖析：对读者较难掌握的知识点进行了详细的分析和释疑。

经典例题：对知识点以例题形式进行了详细的分析和解答。

同步练习及答案：提供了大量的针对性很强的练习题（附答案），练习题均经过精心设计，采用标准题型，突出考点、重点、难点，应试导向准确。

本书还提供了 5 套模拟试题，供读者自我考查。

本书由王丰、徐翠霞主编。谭树琴、魏爱敏、于任霞、刘宗国参与了编写工作。在本书编写过程中，得到了肖孟强教授的大力支持，在此一并表示致谢。

编　者

目　录

第 1 章　基础知识

1.1　考点概述

1.1.1　基本数据结构与算法

1. 算法

（1）算法及其特性

算法是对特定问题求解步骤的一种描述，它是指令的有限序列，其中每一条指令表示一个或多个操作。一个算法具有 5 个重要特性：有穷性、确定性、可行性、输入和输出。

（2）算法评价

评价好的算法有 4 个方面：一是算法的正确性；二是算法的易读性；三是算法的健壮性；四是算法的时空效率——时间复杂度与空间复杂度（运行）。

例如，下列三个程序段：

```
① {++x; s=0;}
② for(i=1;i<=n;++i){++x; s+=x;}
③ for(j=1;j<=n;++j)
    for(k=1;k<=n;++k){++x; s+=x;}
```

这三个程序段的时间复杂度分别为 O（1），O（n）和 O（n^2）。

2. 数据结构

数据结构是相互之间存在一种或多种特定关系的数据元素的集合。数据结构一般包含三个方面的内容：数据的逻辑结构、数据的存储结构以及数据的运算。

数据元素相互之间的关系称为数据的逻辑结构。数据的逻辑结构分为两大类：线性结构和非线性结构。线性结构中的数据元素之间存在一对一的关系，非线性结构中的数据元素之间存在一对多的关系或多对多的关系。

数据结构在计算机中的表示称为数据的物理结构，又称存储结构。它包括数据元素的表示和关系的表示。常用的存储表示方法有 4 种：顺序存储、链式存储、索引存储、散列存储。

3. 线性表、顺序表及其插入和删除

线性表是由 n 个数据元素（结点）组成的有限序列。对于非空的线性表，有且仅有一个开始结点，它没有直接前驱，有 34 且仅有一个终端结点，它没有直接后继，内部的结点都有，且仅有一个直接前驱和直接后继。

顺序表指的是用一组地址连续的存储单元依次存储线性表的数据元素。一般情况下，在顺序表的第 i（1≤i≤n）个位置上插入一个新的数据元素时，需将第 n～i（共 n - i+1）个元素向后移动一个位置。删除顺序表第 i（1≤i≤n）个元素时，需将从第 i+1～n（共 n - i）个元素依次向前移动一个位置。

顺序表的优点是：无须为表示结点间的逻辑关系而增加额外的存储空间；可以方便地随机存取表中任一结点。顺序表的缺点是：插入或删除运算不方便，除表尾的位置外，在

表的其他位置上进行插入或删除操作都必须移动大量的结点，其效率较低；由于顺序表要求占用连续的存储空间，存储分配只能预先进行（静态分配）。

4. 单链表、双向链表与循环链表

链表是用一组任意的存储单元来存放线性表的结点，这组存储单元既可以是连续的，也可以是不连续的，甚至是零散分布在内存中的任何位置上。因此，链表中结点的逻辑次序和物理次序不一定相同。

在链表中，即使知道被访问结点的序号i，也不能像顺序表中那样直接按序号i访问结点，而只能从链表的头指针出发，顺链域next逐个结点往下搜索，直至搜索到第i个结点为止。因此，链表不是随机存取结构，但链表的插入和删除不需要移动元素。链表的结点空间可动态分配，所以链表也不需要是先估计存储空间。

单链表的结点包含一个指针，指明直接后继元素，只能在一个方向上遍历其后的元素。单链表中每个结点的存储地址是存放在其前驱结点指针域中，而开始结点无前驱，故设头指针head指向开始结点。同时，由于终端结点无后继，故终端结点的指针域为空，即NULL。单链表的存取必须从头指针开始进行。

在双向链表的结点中有两个指针域，其一指向直接后继，其二指向直接前驱。循环链表的特点是表中最后一个结点和头结点链接起来，整个链表形成一个环。由此，从表中任一结点出发均可找到表中其他结点。

5. 栈和队列

栈称为后进先出的线性表，限定仅在栈顶进行插入或删除操作。

队列是先进先出的线性表，它只允许在表的一端进行插入，而在另一端删除元素。最早进入队列的元素最早离开。在队列中，允许插入的一端叫做队尾（rear），允许删除的一端则称为队头（front）。

6. 树和二叉树

树是n（n≥0）个结点的有限集。在任意一棵非空树中：（1）有且仅有一个特定的称为根的结点；（2）n>1时，其余结点分成m（m>0）个互不相交的有限集T_1，T_2，...，T_m，其中每一个集合本身又是一棵树并且称为根的子树。

二叉树中每个结点至多只有二棵子树（即二叉树中不存在度大于2的结点），并且二叉树的子树有左右之分，其次序不能任意颠倒。

二叉树具有下列重要特性：

1）在二叉树的第i层上至多有2^{i-1}个结点（i≥1）。

2）深度为k的二叉树至多有2^{k-1}个结点（k≥1）。

3）对任何一棵二叉树T，如果其终端结点数为n_0，度为2的结点数为n_2，则$n_0=n_2+1$。

4）具有n个结点的完全二叉树的深度为$\log_2 n+1$。

一棵深度为k且有2^{k-1}个结点的二叉树称为满二叉树。深度为k的，有n个结点的二叉树，当且仅当其每一个结点都与深度为k的满二叉树中编号从1～n的结点一一对应时，称之为完全二叉树。

由二叉树的先序和中序序列、中序和后序序列均可唯一确定一棵二叉树的结构。

7. 顺序查找与二分法查找

顺序查找是一种最简单的查找方法。在等概率的情况下，查找成功时的平均查找长度为（n+1）/2；在等概率的情况下，查找不成功时的平均查找长度为 n+1；在等概率的情况下，查找成功与查找不成功时的平均查找长度为 3*（n+1）/4。

二分查找，它的基本思想是：以处于区间中间位置记录的关键字和给定值比较。若相等，则查找成功；若不等，则缩小范围，直至新的区间中间位置记录的关键字等于给定值或者查找区间的大小小于零时（表明查找不成功）为止。

二分查找过程可用二叉树来描述，称为描述二分查找的判定树。成功查找表中任一记录的过程就是走了一条从根结点到与该记录相应的结点的路径，和给定值进行比较的关键字个数恰为该结点在判定树上的层次数。因此，二分查找在查找成功时进行比较的关键字个数最多为 $\log_2 n+1$。

8. 基本排序算法

（1）交换排序

起泡排序空间复杂度为 O（1）；起泡排序时间复杂度为 O（n^2）。若初始序列为“正序”序列，则只需进行一趟排序，需进行 n－1 次比较，且不移动记录；若初始序列为“逆序”序列，则需要进行 n－1 趟排序，需进行 n（n－1）/2 次比较，需进行 3n（n－1）/2 次的记录移动。

快速排序的基本思想是，通过一趟排序将待排序记录分割成独立的两部分，其中一部分记录的关键字均比另一部分记录的关键字小，再分别对这两部分记录继续进行排序，以达到整个序列有序。

通常，快速排序在所有同数量级（O（$n\log_2 n$））的排序方法中平均性能好。但是，若初始记录序列按关键字有序或基本有序时，快速排序将蜕化为起泡排序，时间复杂度为 O（n^2）。快速排序需要一个栈空间来实现递归，栈的最好深度为 O（$\log_2 n$），最坏深度为 O（n）。

（2）选择排序

选择排序的基本操作是：每一趟从待排序的记录中选出关键字最小的记录，顺序放在已排好序的子文件的最后，直到全部记录排序完毕。

简单选择排序的空间复杂度为 O（1），时间复杂度为 O（n2）。简单选择排序过程中，所需进行记录移动的次数较少，最小为 0，最大为 3（n－1）。然而，不管记录的初始排列如何，简单选择排序所需进行的关键字间的比较次数相同，均为 n（n－1）/2。

堆排序在最坏的情况下，其时间复杂度也为 O（nlogn）。相对于快速排序来说，这是堆排序的最大优点。此外，堆排序仅需一个记录大小供交换用的辅助存储空间。

（3）插入排序

直接插入排序的基本操作是将一个记录插入到已排好序的有序表中，从而得到一个新的、记录数增 1 的有序表。

直接插入排序空间复杂度为 O（1）；直接插入排序时间复杂度为 O（n^2）。若初始序列为“正序”序列，则只需进行一趟排序，需进行 n－1 次比较，且不移动记录；若初始序列为“逆序”序列，则需要进行 n－1 趟排序，需进行（n+2）（n－1）/2 次比较，需进行（n+4）（n－1）/2 次的记录移动。

希尔排序又称“缩小增量排序”，它的基本思想是：先将整个待排记录序列分割成为若干子序列分别进行直接插入排序，待整个序列中的记录“基本有序”时，再对全体记录进

行一次直接插入排序。希尔排序空间复杂度为O（1），时间复杂度为O（$n^{1.3}$）。

1.1.2 程序设计基础

1. 程序设计方法与风格

程序设计方法包括：结构化程序设计方法和面向对象的程序设计方法。

程序设计风格包括：源程序文档化，数据说明的方法，语句结构和输入/输出方法。源程序文档化包括恰当的标识符、适当的注解和程序的视觉组织等。选择含义鲜明的名字，使它能正确地提示程序对象所代表的实体，这有助于对程序的理解；注解是程序员和程序读者通信的重要手段，正确的注解非常有助于对程序的理解；程序的清单布局对于程序的可读性也有很大影响，应该利用适当的阶梯形式使程序的层次结构清晰明显。

2. 结构化程序设计方法

结构化程序设计方法的主要原则可以概括为自顶向下逐步求精、模块化及限制使用goto语句。

3. 面向对象的程序设计方法

对象可以定义为系统中用来描述客观事物的一个实体，它是构成系统的一个基本单位，由一组属性和一组对属性进行的操作组成。

属性一般只能通过执行对象的操作来改变。

操作又称为方法或服务，在C++中称为成员函数，它描述了对象执行的功能，若通过消息传递，还可以为其它对象使用。

消息是一个对象与另一个对象的通信单元，是要求某个对象执行类中定义的某个操作的规格说明。

把具有相同特征和行为的对象归在一起就形成了类。属于某个类的对象叫做该类的实例。类定义了各个实例所共有的结构，类的每一个实例都可以使用类中定义的操作。

继承是使用已存在的定义作为基础建立新定义的技术。如果某几个类之间具有共性的东西，抽取出来放在一个一般类中，而将各个类的特有的东西放在特殊类中分别描述，就可建立起特殊类对一般类的继承。

1.1.3 软件工程基础

1. 软件工程基本概念

软件工程是开发、运行、维护和修复软件的系统方法，其中，软件的定义为：计算机程序、方法、规则、相关文档资料以及在计算机上运行时所必需的数据。

软件工程包括三个要素：方法、工具和过程。软件工程方法为软件开发提供了“如何做”的技术。它包括了多方面的任务，如项目计划与估算、软件系统需求分析、数据结构、系统总体结构的设计、算法过程的设计、编码、测试以及维护等。软件工具为软件工程方法提供了自动的或半自动的软件支撑环境。软件工程的过程是将软件工程的方法和工具综合起来以达到合理、及时地进行计算机软件开发的目的。过程定义了方法使用的顺序、要求交付的文档资料、为保证质量和协调变化所需要的管理、及软件开发各个阶段完成的里程碑。实施软件工程项目，希望得到项目的成功。所谓成功指的是达到以下几个主要目标：付出较低的开发成本、达到要求的软件功能、取得较好的软件性能、开发的软件易于移植、

需要较低的维护费用、能按时完成开发工作，及时交付使用。

软件产品是从形成初始的概念开始，经过设计、开发、使用和维护，直至该软件最后退役，这个全过程称为软件生存周期。软件的生存周期包括：软件定义、软件开发、软件的使用和维护。软件定义包括：问题定义、可行性研究、需求分析。软件开发包括：概要设计、详细设计、编码、测试。维护是软件生命周期的最后一个阶段，也是持续时间最长，花费代价最大的一个阶段。

软件维护活动包括以下几类：改正性维护、适应性维护、完善性维护和预防性维护。改正性维护是指在软件交付使用后为了识别和纠正软件错误、改正软件性能上的缺陷、排除实施中的误使用所应当进行的诊断和改正错误的过程；适应性维护是指为了使软件适应变化而修改软件的过程；完善性维护是指为了满足用户对软件提出的新功能与性能要求，需要修改或再开发软件，以扩充软件功能、增强软件性能、改进加工效率、提高软件的可维护性；预防性维护是为了提高软件的可维护性、可靠性等，为以后的进一步改进软件打下良好基础。

2. 结构化分析方法

结构化分析方法是面向数据流进行分析的方法，它是按照功能分解的原则，根据软件内部数据传递、变换的关系，自顶向下逐层分解，直到找到满足功能要求的所有可实现的软件为止。结构化分析的常用工具有数据流图、数据字典、判定树和判定表。

数据流图的基本图形元素包括 4 种：

1）加工：输入数据在此进行变换产生输出数据，其中要注明加工的名字。

2）数据流：被加工的数据与流向，箭头边应给出数据流名字。

3）数据存储文件：处理过程中存放各种数据的文件，也必须加以命名。

4）数据输入的源点和数据输出的汇点

数据词典是对于数据流图中出现的所有命名元素，包括数据流、加工、数据文件，以及数据的源点、汇点等，在数据词典中作为一个词条加以定义，使得每一个图形元素的名字都有一个确切的解释。

软件需求规格说明书是需求分析阶段的最后成果，是软件开发中的重要文档之一。它有以下几个方面的作用：

1）便于用户、开发人员进行理解和交流。

2）反映出用户问题的结构，可以作为软件开发工作的基础和依据。

3）作为确认测试和验收的依据。

3. 软件设计

软件设计是一个把软件需求变换成软件表示的过程。软件设计的任务是把分析阶段产生的软件需求说明转换为用适当手段表示的软件设计文档。

软件设计的内容包括：数据设计、体系结构设计、接口设计和过程设计。数据设计将分析阶段建立的信息模型转换成实现软件所需的数据结构。体系结构设计定义软件主要组成部件之间的关系（相当于功能设计）。接口设计描述软件内部、软件和接口系统之间以及软件与人之间是如何通信的（包括数据流和控制流）。过程设计将软件体系结构的组成部件转变成对软件组件的过程性描述（也就是对功能的具体描述）。

软件设计是一个把软件需求变换成软件表示的过程。软件设计分两步完成。首先做概要设计，将软件需求转化为数据结构和软件的系统结构，并建立接口。然后是详细设计，

即过程设计。通过对结构表示进行细化，得到软件的详细的数据结构和算法。

概要设计又称为初步设计或总体设计，概要设计的基本目的是概要地说明系统应该怎样实现。在过程设计阶段，要决定各个模块的实现算法，并精确地表达这些算法。过程设计的表达工具可以分为：程序流程图、N-S图、PAD、判定表、PDL 、HIPO图 。

软件设计遵循软件工程的基本目标和原则：抽象化、自顶向下，逐步细化、模块化和信息隐蔽。

模块独立性可用两个定性的标准来度量：内聚与耦合。

内聚是一个模块内部各成分之间相互关联程度的度量（块内联系）。耦合是模块之间相互依赖程度的度量（块间联系）。内聚和耦合是密切相关的，与其他模块存在强耦合的模块通常意味着弱内聚，而强内聚的模块通常意味着与其他模块之间存在弱耦合。模块设计追求强内聚，弱耦合。

内聚按强度从低到高有以下 7 种类型：偶然内聚、逻辑内聚、时间内聚、过程内聚、通信内聚、信息内聚、功能内聚。

耦合按从强到弱的顺序可分为以下七种类型：非直接耦合、数据耦合、特征耦合、控制耦合、外部耦合、公共耦合、内容耦合。

典型的数据流类型有两种：变换型和事务型。

4. 软件测试

软件测试是为了发现错误而执行程序的过程，即根据软件开发各阶段的规格说明和程序的内部结构而精心设计一批测试用例（即输入数据及其预期的输出结果），并利用这些测试用例去运行程序，以发现程序错误的过程。

软件测试有三个重要特征：挑剔性、复杂性、完全测试的不可能性及经济性。

软件测试分为动态测试和静态测试两类。静态测试是通过对被测程序的静态审查发现代码中潜在的错误。它一般采用两种方式：一种是采用人工方式脱机完成，亦称为人工测试或代码评审；一种是借助静态分析器在机器上以自动方式进行检查，但不要求程序本身在机器上运行。

动态测试也可分为两类：一类把被测程序看成一个黑盒，根据程序的功能来设计测试用例，称为“黑盒测试”，目的是测试软件功能是否达到了预期的要求；另一类则根据被测程序的内部结构设计测试用例，测试者事先了解程序的内部工作过程，故称为“白盒测试”，目的是为了测试每个内部操作是否符合设计要求。

对软件测试而言，黑盒测试法是指把程序看成一个黑盒子，完全不考虑程序的内部结构和处理过程。也就是说，黑盒测试是在程序接口进行的测试，它只检查程序功能是否能按照规格说明书的规定正常使用，程序是否能适当地接收输入数据产生正确的输出信息，并且保持外部信息（如数据库或文件）的完整性。因此，黑盒测试又称为功能测试。常用的黑盒测试有等价分类法、边值分析法、错误推测法、因果图法。

与黑盒测试法相反，白盒测试法的前提是把程序看成装在一个透明的白盒子里，也就是完全了解程序的结构和处理过程。这种方法按照程序内部的软件测试程序，检验程序中的每条通路是否都能按预定要求正确工作。白盒测试又称为结构测试。白盒测试分成两类：逻辑覆盖测试和路径测试。

多模块程序的测试共包括 4 个层次，即单元测试、集成测试、确认测试和系统测试。单元测试集中对用源代码实现的每一个程序单元进行测试，检查各个程序模块是否正确地实现了规定的功能。然后，进行集成测试，根据设计规定的软件体系结构，把已测试过的

模块组装起来，在组装过程中，检查程序结构组装的正确性。确认测试则是要检查已实现的软件是否满足了需求规格说明中确定了的各种需求，以及软件配置是否完全、正确。最后是系统测试，把已经过确认的软件纳入实际运行环境中，与其它系统成份组合在一起进行测试。

5. 软件调试

软件调试是在进行了成功的测试之后才开始的工作。它与软件测试不同，软件测试的目的是尽可能多地发现软件中的错误，而软件调试的任务则是但进一步诊断和改正程序中潜在的错误。调试活动由两部分组成：确定程序中可疑错误的确切性质和位置；对程序（设计，编码）进行修改，排除这个错误。

常用的调试策略主要有：强行排错、回溯法排错、归纳法排错、演绎法排错。

1.1.4 数据库设计基础

1. 数据库的基本概念

数据库是长期存储在计算机内的有组织可共享的数据集合。数据库中数据按一定的数据库模型组织、描述和存储，具有较小的冗余度，较高的数据独立性和易扩展性，并可为各种用户共享。数据独立性包括物理独立性和逻辑独立性。

数据库管理系统（DBMS）是位于用户和操作系统之间的一层数据管理软件，它的主要功能包括：数据定义功能、数据操纵功能、数据库的运行管理、数据库的建立和维护功能。

数据库是系统中的共享资源，可多用户同时使用数据库。因此，系统必须提供以下几个方面的数据控制功能。

（1）数据安全性控制：防止非法用户的使用造成数据库中数据被泄密或破坏。例如，检查用户身份、定义用户密级和数据存取权限。

（2）数据完整性控制：数据的完整性是指数据的正确性、有效性和相容性。保证数据库中数据在输入或修改过程中语义的正确性和有效性，符合原来的定义和规定。例如，学号是惟一的，性别是男或女，月份是 1～12 之间的整数。

（3）数据并发控制：正确处理好多用户、多任务环境下的并发操作，防止错误发生。

（4）恢复：当数据库被破坏或数据不正确时，使数据库能恢复到正确状态。

数据库系统（DBS）是带有数据库的计算机系统。由以下几部分组成：数据库、数据库管理系统、数据库管理员、硬件和软件。

数据库技术的主要目的是有效地管理和存取数据资源，包括以下几方面：提供数据的共享性，使多个用户能同时访问数据库中的数据；减少数据的冗余度，提交数据的一致性和完整性；提供数据与应用程序的独立性，减少应用程序的开发和维护代价。

2. 数据模型

数据是现实世界符号的抽象，而数据模型则是数据特征的抽象，是对客观事物及联系的数据描述，它反映了实体内部及实体与实体之间的联系。数据模型按不同的应用层次分成 3 种类型，它们是：概念数据模型、逻辑数据模型、物理数据模型。数据模型是严格定义的概念的集合，这些概念精确地描述系统的静态特性、动态特性和完整性约束条件。因此，数据模型通常由数据结构、数据操作和数据约束（又称完整性约束）三部分组成。数据结构是刻画数据模型性质的最重要的方面。

最常用的数据模型有 3 种：层次模型、网状模型和关系模型。用树型结构表示实体类

型及实体间联系的数据模型称为层次模型，用有向图结构表示实体类型及实体间联系的数据模型称为网状模型，用二维表格结构表示实体及其联系的数据模型称为关系模型。层次模型和网状模型统称为非关系模型。关系模型的完整性约束条件包括实体完整性、参照完整性和用户定义的完整性。其中实体完整性、参照完整性是由系统自动支持的。

3. 关系代数运算

在关系模型中把数据看成一个二维表，所以，关系模型数据库的基本结构是二维表，每一个二维表称为一个关系。表中的每一列称为一个属性，相当于记录中的一个数据项，一个具有N个属性的关系称为N元关系。表中的每一行称为一个元组，相当于记录值。

关系的基本运算有两种：传统的集合运算和专门的关系运算。传统的集合运算按行进行，包括并、交、差、笛卡尔积等运算，专门的关系运算包括选择、投影和连接，该类运算涉及行和列。笛卡尔积是将指定集合中的每个元组逐个进行乘积运算，所花费的时间比选择、联接等运算要长。

设S和R是两个关系，S与R的并运算产生既包含S中的元组也包含R中的元组的集合。S与R的交运算产生既属于S又属于R的元组的集合。S与R的差运算产生属于S但不属于R的元组的集合。S与R的积运算产生的是两个关系的笛卡尔积。选择是从关系中找出满足指定条件元组的操作，从水平方向抽取记录，产生的新关系属性与原来关系中的相同。投影是从关系中指定若干个属性组成新的关系，对关系进行垂直分解，与元组个数无关，产生的新关系元组个数与原来关系中元组个数相等。连接是两个关系的横向结合，拼成一个更宽的关系模式。

数据语言包括数据定义语言（DDL）和数据操纵语言（DML）。数据定义语言负责数据的模式定义与数据的物理存取构建，数据操纵语言负责数据的操纵，包括查询、删、改等操作。数据操纵语言可分为两类：关系代数和关系演算。关系代数是用对关系的运算来表达查询的，而关系演算是用谓词表达查询的。关系演算又分元组关系演算和域关系演算。

4. 数据库设计方法和步骤

数据库设计分为6个阶段：需求分析阶段、概念设计阶段、逻辑设计阶段、物理设计阶段、实施阶段及数据库运行和维护阶段。在需求分析阶段形成概念模式，逻辑设计阶段先形成数据库的逻辑模式，再形成数据的外模式，物理设计阶段形成数据库的内模式。

1.2 难点剖析

1.2.1 二叉树形态的确定

1. 由二叉树的先序和中序序列确定一棵二叉树

依据先序遍历序列可确定整个二叉树的根结点，再依据中序遍历序列可知其左子树的结点和右子树的结点，由左子树的先序遍历序列可知其左子树的根结点，又由右子树的先序遍历序列可知其右子树的根结点，依此而推，直到所有非空子树均找到根结点为止。

2. 由二叉树的中序和后序序列确定一棵二叉树

依据后序遍历序列可确定整个二叉树的根结点，再依据中序遍历序列可知其左子树的

后序遍历序列可知其右子树的根结点，依此而推，直到所有非空子树均找到根结点为止。

1.2.2　ER 模型向关系模型的转换

ER 模型向关系模型的转换，实际上就是把 ER 图转换成关系模式的集合。ER 图的主要成份是实体类型和联系类型，转换规则就是如何把实体类型、联系类型转换成关系模式。

1. *实体类型的转换规则*

将每个实体类型转换成一个关系模式，实体的属性即为关系模式的属性，实体的标识符即为关系模式的键。

2. *联系类型的转换规则*

1）1：1 的联系：可以在二个实体类型转换成的二个关系模式中任意一个关系模式的属性中加入另一个关系模式的键和联系类型的属性。

2）1：N 联系：在 N 端实体类型转换成的关系模式中加入一端实体类型的键和联系类型的属性。

3）M：N 联系：将联系类型也转换成关系模式，其属性为二端实体类型的键加上联系类型的属性，而键为二端实体键的组合。

1.3　经典例题

一、选择题

【例题 1】链表不具有的特点是（　　）。

A）不必事先估计存储空间　　B）可随机访问任一元素

C）插入或删除不需要移动元素　　D）所需空间与线性表长度成正比

【答案】B

【解析】链表采用的是链式存储结构，它克服了顺序存储结构的缺点：它的结点空间可以动态申请和释放；它的数据元素的逻辑次序靠结点的指针来指示，不需要移动数据元素。但是链式存储结构也有不足之处：每个结点中的指针域需额外占用存储空间；链式存储结构是一种非随机存储结构。所以选项 B）正确。

【例题 2】栈和队列的共同特点是（　　）。

A）都是先进先出　　B）都是先进后出

C）只允许在端点处插入和删除元素　　D）没有共同点

【答案】C

【解析】栈和队列都是一种特殊的操作受限的线性表，只允许在端点处进行插入和删除。二者的区别是：栈只允许在表的一端进行插入或删除操作，是一种“后进先出”的线性表；而队列只允许在表的一端进行插入操作，在另一端进行删除操作，是一种“先进先出”的线性表。

【例题 3】如果入栈序列为 1，2，3，4，则可能的出栈序列为（　　）。

A）3，1，4，2　　B）2，4，3，1

C）3，4，1，2　　D）任意顺序

【答案】B

【解析】由栈“后进先出”的特点可知：A）中1不可能比2先出，C）中3不可能比4先出，且1不可能比2先出，D）中栈是先进后出的，所以不可能是任意顺序。故答案为选项B）。

【例题4】已知二叉树后序遍历序列为DABEC，中序遍历序列为DEBAC，它的先序遍历序列是（　　）。

A）ACBED　　B）DECAB　　C）DEABC　　D）CEDBA

【答案】D

【解析】依据后序遍历序列可确定根结点为C；再依据中序遍历序列可知其左子树由DEBA构成，右子树为空；又由左子树的后序遍历序列可知其根结点为E，由中序遍历序列可知其左子树为D，右子树由BA构成。求得该二叉树的先序遍历序列为选项D）

【例题5】在深度为5的满二叉树中，叶结点的个数为（　　）。

A）32　　B）31　　C）16　　D）15

【答案】C

【解析】在满二叉树中，每层上的结点数都达到最大值，即在满二叉树的第k层上有2^{k-1}个结点，且深度为m的满二叉树有2^m-1个结点。满二叉树的所有叶结点均出现在最后一层上，所以深度为5的满二叉树中，叶结点的个数为16。

【例题6】已知数据表A中每个元素距其排序后的最终位置不远，为节省时间，应采用的排序算法是（　　）。

A）堆排序　　B）直接插入排序

C）快速排序　　D）直接选择排序

【答案】B

【解析】当数据表A中每个元素距其最终位置不远，说明数据表A按关键字值基本有序，在待排序序列基本有序的情况下，采用插入排序所用时间最少，故答案为选项B）。

【例题7】在下列几种排序方法中，要求内存量最大的是（　　）。

A）插入排序　　B）选择排序　　C）快速排序　　D）归并排序

【答案】D

【解析】本题考查排序方法的实现机制。插入排序和选择排序的空间复杂度为O（1），快速排序空间复杂度为O（$\log_2 n$）或O（n）；归并排序的空间复杂度为O（n）所以选项D）正确。

【例题8】下面关于二分查找的叙述正确的是（　　）。

A）表必须有序，表可以顺序方式存储，也可以链表方式存储。

B）表必须有序且表中数据必须是整型，实型或字符型。

C）表必须有序，而且只能从小到大排列。

D）表必须有序，且表只能以顺序方式存储。

【答案】D

【解析】本题考查的是二分查找。二分查找时，表必须有序，但不一定是从小到大排列。表只能以顺序方式存储，故答案为D。

【例题9】在设计程序时，应采纳的原则之一是（　　）。

A）不限制goto语句的使用　　B）减少或取消注解行

C）程序越短越好　　D）程序结构应有助于读者理解

【答案】D

【解析】滥用goto语句将使程序流程无规律，可读性差，因此A）有误；注解行有利