

## 第一章 计算机技术概论

计算机：一种可以接受输入、处理数据、存储数据、可编程并能产生输出的一种电子装置。

### 1.1 信息及数字化

信息数字化：将字符、图像、语音和视频等表示为0和1组合，进一步使用计算机存储、传输和处理。

#### 1.1.1 信息概述

1. 信息：客观存在的事物及其运动状态的表征。

#### 2. 信息的基本特征

普遍性 只要有事物存在或变化，就存在信息。

寄载性 信息寄载于事物，不能独立于事物之外

共享性 信息是一种资源，可为人类所共享。

时效性 价值因时间或地点不同而发生变化。

可识别性 通过眼、耳和鼻等直接或间接识别。

可加工性 从中提炼出更有价值的信息。

#### 1.1.2 数据及信息数据化

##### 1. 数据

信息的载体，包括数值、字符、图形、图像、声音和视频等。

数据处理：对各种数据进行采集、存储、传送、转换、分类、排序、计算和输出等操作。在计算机内，任何形式的数据都用二进制数表示。

##### 2. 信息数字化

将事物的信息进一步抽象、提取和规范化。计算机是信息处理机，处理信息时，必须将现实世界中的信息转换为计算机能识别、存储和处理的二进制形式，经加工处理，再将结果提供给外界。

### 1.2 数制及其转换

现代计算机系统采用二进制存储、处理和传输。为书写和表示方便，引进八进制和十六进制。

#### 1.2.1 进位计数制

三要素：

基数 是指在某种进位计数制中所使用的数码个数。

位权 是按所采用的基数和对应数位来表示一个固定数。

数码

十进制数的基数是10。数码有0、1、2、3、4、5、6、7、8和9共10个数字符号。

2B 80 10D 16H

(1) R进制转10进制：按权展开式

$$(307.16)_R = 3 \times R^2 + 0 \times R^1 + 7 \times R^0 + 1 \times R^{-1} + 6 \times R^{-2} \quad (\text{起始为 } R^{n-1})$$

3、0、7 ( $a_i$ ) 为数码

8 (R) 为基数，可取最大为 R-1

$R^i$  ( $R^i$ ) 为位权

(2) 十进制转R进制

整数除R取余，由下到上

小数乘R取整，由上到下，计算到n+1位， $<R/2$ 舍去， $\geq R/2$ 进位

### (3) 二、八、十六进制数间的相互转换

一位八进制数对 3 位二进制数

一位十六进制对 4 位二进制数

## 1.3 文本数据编码

### 1.3.1 西文字符编码

采用 ASCII 码。即美国信息交换标准字符码。用 7 位二进制数表示一个字符，编码范围为 (0000000)B~(1111111)B，可表示 2<sup>7</sup> 即 128 个字符。

用 8 位二进制存储，最高位为 0。

- 控制字符：第 000、001 列和最后字符 (DEL) 共 33 个，传输/打印或显示时起控制作用。  
如：111 (BEL) 响铃；1101 (CR) 回车；11011 (ESC) 逃脱键等。
- 可打印(显示)字符：其余 95 个，可在显示器和打印机等设备上输出，也称图形字符。
- 从 0~9、A~Z、a~z 顺序排列，且同一字母小写比大写码值大 32。

### 1.3.2 汉字编码

1. 汉字输入码：指使用键盘输入汉字的编码，也称外码。有区位码、拼音和字形编码等。
2. 国标码：国标 GB2312-80 定义 7445 个字符编码，有 6763 个汉字和 682 个符号。用 2 个字节存储一个汉字或符号，每个字节最高位为 0 等。用 4 位十六进制数表示。“中”字码为 5650H。
3. 汉字内码：计算机内部存储和处理汉字码，是国标码各个字节加 80H，即各字节最高位为 1，存储一个汉字需要 2B。
4. 汉字字形码：用于输出 (显示或打印) 字符，也称字模。在 N×N 表格中画字，字覆盖的单元格填 1，其余填 0，即生成一个字的输出码(字模)，全部字的字模构成字模库。  
一个单元格对应一个像素点，即一个字由 N×N 个点阵组成。通常有 16×16、24×24 和 32×32 点阵。
5. 汉字的显示原理  
键盘输入的汉字输入码变换成机内码，进行处理和存储；  
输出汉字时，从字模库中检索字模送到输出设备(如显示器或打印机)。

输入→输入码→机内码→字形码→输出

## 1.4 文本信息数字化

文本信息数字化：将汉字、英文和数字等符号以自身编码的形式转化为计算机能存储、处理和传输的数据。

手工输入：通过终端设备向计算机传递信息，常见终端设备有键盘、触摸屏和手写板等。效率较低，出错率较高。

典型的软件有记事本和 Office 办公软件等。

### 1.4.1 机读卡信息采集

机读卡：常用于标准化考试、问卷调查和选举投票等，常见有 8421 和 普通机读卡 两种类型。

光标阅读机：也称光电读卡机，是机读卡信息的数字化设备。

#### 1. 8421 机读卡

采用 **8421 编码**，也称 BCD 编码、8421BCD 编码。用二进制编码表示十进制数。每列印刷 **[8]**、**[4]**、**[2]**和**[1]**4 个选项，涂写的选项之和表示要输入的本列数字，如涂写[2]和[4]表示 6。涂写的选项之和大于 10 时无效，等于 10 时表示 0。

#### 2. 普通机读卡

在普通机读卡上印刷各类选项，**直接涂写**某项，表示输入该项数据，不涂写任何项时表示无数据。

与 8421 机读卡比较，同一个数据的**选项更多**，占用版面**空间更大**，涂写信息时**更容易被理解和接受**。

### 1.4.2 条形码的制作与识别

**条码**：是表示一串符号的黑白条形图形，常印制或粘贴在商品、固定资产、药品或图书上，用于表示物品的编码。

#### 1. 条形码分类

**EAN 码**：常用于商品管理，印刷在商品包装上。

**Codebar 码**：常用于医疗管理。

**ISBN 码**：常用于出版物管理，印刷在图书封底。

**Code39 码**：可以表示数字和字母，常用于通用信息管理。

#### 2. 条形码的特点

**速度快**：可以达到键盘输入的 5-10 倍。

**信息量大**：一次可采集几十个字符。

**可靠性高**：误码率低于百万分之一。

**制作成本低**：制作条码标签对设备和材料没有特殊要求，用普通打印机和纸张即可制作条码标签。

#### 3. 条形码标签的制作

需要通过**专门软件**制作条形码标签，在网上通常可以下载各种制作条形码标签的软件或在线制作。

#### 4. 条形码信息数字化

需要计算机连接**条形码扫描器**，不需要专门的软件接收数据，凡是能接收键盘输入的软件都能接收条形码扫描器的数据。

### 1.4.3 磁卡的信息存储

**磁卡**：以液体**磁性材料**或**磁条**为信息**载体**，将磁性材料涂覆在纸卡片上或将磁条压贴在塑料卡片上。磁材料的破损将导致无法读写磁卡中信息。

**特点**：制作**成本低**、**可靠性强**、记录**数据密度大**、**误读率低**、信息**存入**和**读出速度快**。

遵循 ISO7811 系列标准，规定有 **3** 个磁道。

**第一磁道**：可以存储 76 个字母或数字，通常存储磁卡的类型信息；

**第二磁道**：可以存储 37 个数字，通常存储账户信息；

**第三磁道**：能存储 104 个数字，通常存储余额等信息。

第一、二磁道**仅**可以**写入数据一次**，写入后不能修改，第三磁道可以**写入数据多次**。

与 IC 卡比较，磁卡存在不足：

- 存储**空间**和存储**信息量小**

- 磁条易读出和伪造，安全性差
- 磁材料外漏，容易磁化或破损

#### 1.4.4 IC卡的信息存储与识别

集成电路卡：简称 IC卡，将集成电路芯片嵌入塑料基片指定位置，利用集成电路的存储特性，保存、读取和修改信息。

优缺点：

- 存储容量是磁卡的几倍至几十倍；
- 具有防伪造、防篡改的能力，安全性高。
- 存在价格高，抗静电和紫外线能力弱等缺点。

#### 1.4.5 二维码的制作与识别软件

二维码：也称二维条码，通过特定几何形状按一定规律在平面上组合记录信息，使用图像输入设备或光电扫描设备读取信息。

特点：存储数据量更大；可以存储字符（汉字）、数字、网络地址（URL）和商品编码等信息。

编码方式：常用的有 PDF417、QR Code 和 Code 49 等，其中 QR Code 使用较为广泛。

##### 1. 二维码图区域的结构

定位与校正：包括位置探测图形、定位图形和校正图形，用于二维码图像定位与校正。

格式信息定义：定义二维码的纠错级别，使得二维码具备纠错功能，最多可以纠错约 30%。

版本信息定义：定义二维码的规格，不同规格可以存储的信息量不同。例如，版本 40、纠错级别 L 的二维码最多可以存储 7089 个数字，4296 个字母，1817 个汉字。

#### 1.4.6 图像符号转换成文本

字符识别：或称光学字符识别，是将数字图像中的符号转换为文本信息的技术，应用于银行票据、档案和卷宗处理等，以最终识别率、转换速度、版面理解正确率和版面还原满意度作为识别结果的评测依据。

##### 1. 文本转换过程

- 图像获取：数码相机或手机拍摄的数字照片，扫描设备获取的数字图像。
- 预处理：主要包括二值化、图像增强、干扰信息处理和图像滤波等。
- 特征提取：根据数字、英文和汉字的书写特点，提取相关符号的特征。
- 识别分类：根据提取的特征与样本库进行比对，选择相似度最高的数字、英文或汉字。
- 后处理：由于字体、字号或文档清晰度等因素，识别结果可能存在误差，利用语法和上下文信息等，对识别的结果进行修正。

最新技术：使用人工神经网络和深度学习等技术，提高转换的准确率。

#### 1.4.7 语音识别及文本转换软件

语音识别：语音信号转化为文本信息。

##### 1. 语音输入

在线识别：将语音内容传送到远端服务器，由服务器完成识别工作并返回文本信息，识别准确率较高，但是需要联网。

离线识别：由本地设备完成识别工作，需要下载离线语言识别库文件，由于本地设备计算和存储能力的限制，识别准确率较低。

##### 2. 实时翻译

又称同声传译，将语音识别为文本后，再将文本翻译为其他语言文本，并使用语音合成技术转化为语音。

## 1.6 互联网（+）及物联网

传统互联网强调人与人及人与计算机之间的信息传递和沟通。互联网（+）和物联网使用电子标签和传感器等设备获得物体的当前状态信息，以互联网作为信息传递通道将信息及时传递出去，让人与物、物与物之间的有效通信变为可能。

### 1.6.1 互联网+

“互联网+”就是“互联网+各个传统行业”，但这并不是简单的两者相加，而是利用信息通信技术以及互联网平台，让互联网与传统行业进行深度融合，创造新的发展生态。

#### 1. “互联网+”的基本构成

(1)云：是指大数据和云计算基础服务。通过使用这些服务，复杂的数据处理和计算变成了简单的网络请求和接收处理结果，用户可以将创造力专注于处理结果的应用。

(2)网：包括传统互联网、移动互联网和物联网。

(3)端：用户直接接触的计算机、移动设备、可“穿戴”设备和传感器等。

#### 2. 互联网+X

(1)互联网+金融：创造了在线理财、支付、众筹等多种金融模式；

(2)互联网+通信：创造了移动即时通信模式；

(3)互联网+交通：创造了网约车、共享单车、网上购买火车和飞机票、出行导航和智能交通指挥系统等模式。

### 1.6.2 物联网(Internet of Things)

通过二维码扫描设备、射频识别(RFID)装置、红外感应器、全球定位系统和激光扫描器等信息传感设备，按约定的协议，将物品与互联网相连接，进行信息交换和通信，以实现智能化识别、定位、跟踪、监控和管理的一种网络。物联网以广域网为基础，是广域网扩展应用的产物。

#### 1. 物联网的作用

(1) 通过传感器和互联网结合，可使物品与网络相连，实现物品的远程识别和管理。

(2) 物联网是多网融合的产物，将人之间的沟通扩展到人与物、物与物之间的沟通，智能化、网络化的生活方式使人们工作、生活更加便捷和人性化。

#### 2. 物联网技术

(1) 射频识别：又称射频标签，附着于物品表面或嵌入物品内部，通过无线电信号采用非接触方式读写数据，最远读写距离可达数百米，可以用于食品溯源、物流跟踪、高速路口收费和汽车无钥匙开门等领域。

射频标签根据有无电源，分为无源被动型和有源主动型。无源被动型本身无电源，体积小、使用年限长和感应距离短；有源主动型距离远、稳定性好和读取速度快，主要应用于智能停车场、智能交通、智慧城市和物联网等领域。

(2) 传感器：是用于探测物体的温度、湿度、压力或速度等物理现象的一种设备，并能将测试到的信息数字化。

(3) M2M技术：是机器之间、机器与人之间的通信技术，是以机器终端设备智能交互为核心的，网络化的应用与服务，通过在机器内部嵌入通信模块将机器接入网络。

(4) 两化融合：是信息化和工业化的高层次的深度结合，是物联网的主要推动力之一。以信息化带动工业化，以工业化促进信息化，在国家“物联网十二五规划”中描述为“企业信息化，信息条码化”。

#### 3. 物联网的应用

- (1) **智能交通**：使用物联网技术监测道路拥堵或突发事件，系统自动切换红绿灯，充分利用道路资源，向驾驶员推荐最佳行驶路线等。
- (2) **智能建筑**：通过 GPS 或北斗系统，在电子地图上准确、及时反映出建筑物空间、地理位置信息。通过物联网技术获得当前天气情况，建筑物内照明灯自动调节光亮度，实现节能环保。
- (3) **生态保护**：通过物联网技术采集古树状态信息，结合当前的温度和湿度，及时作出数据分析和保护措施。

### 1.6.3 电子商务

**电子商务**是使用**计算机**和**网络通讯技术**进行的商务活动。电子商务不等同于商务电子化，是以电子化和网络化为基础开创新的商业模式。电子商务不仅包括在线购物，还包括商品推荐、物流配送、售后评价和网上支付等内容。

### 1.6.4 电子政务

**电子政务**是指**政务活动**中全面应用**现代信息技术、网络技术**以及**办公自动化技术**等进行办公、管理和为社会提供公共服务的一种全新管理模式。

- (1) **政务公开**：加强政府的信息服务，在网上设有政府网站，向公众提供可能的信息服务。
- (2) **网上办公**：建立网上服务体系，使公众通过网络办理事务，实现网络无纸化办公。
- (3) **政府采购**：将电子商务用于政府，实现政府采购电子化。

## 1.7 大数据、数据挖掘及其应用

**大数据**：在一定时间内**无法**用传统方法对其内容进行**抓取、管理**和**处理**的数据集合。

### 1. 大数据的特点

- (1) **数据体量巨大**：仅某搜索引擎每天提供的数据量就超过 1.5PB（1PB=1024TB）。
- (2) **数据类型多样**：包含文本、图像、音频、视频和地理位置信息等不同类型的数据。
- (3) **处理速度快**：需要从各种类型的数据中快速获得高价值信息，信息价值具有时效性。
- (4) **价值密度低**：以违章视频监控为例，24 小时的视频数据，违章信息可能只有一两秒钟。

### 2. 数据挖掘

**数据挖掘**：也称知识发现或数据分析，是指从**已有的数据**中**发现并抽取**出**有价值的信息和知识**，寻找数据间**潜在的关联**，为未来趋势**预测**和行为**决策**提供参考的技术。

**数据挖掘对象**：通常是**海量的**、**不完全的**、**有噪声的**、**模糊的**和**随机**的数据。数据可以是**结构化**的，如关系数据库中的数据，也可以是**非结构化**的，如文本、图像和视频等数据。

数据挖掘主要方法：

- (1) **分类**：对数据中某些**特征值**进行比较，按照**已定义分类标准**将数据映射到不同类别。。
- (2) **聚类**：将数据按照**相似性**或**差异性**分为不同类别，目的是使得属于同一类别的数据之间相似度尽可能大，不同类别的数据之间相似度尽可能小。
- (3) **关联规则**：**描述**数据项之间所存在的**关系**，**挖掘**隐藏在数据间的关联或相互**关系**。
- (4) **特征分析**：从一组数据中**提取**出关于这些数据的**特征**式，用于表示数据集的总体特征。

### 3. 大数据应用

以基于**大数据**和**数据挖掘**的推荐系统为例，根据用户的历史行为资料，例如，查阅、购买和评价过的物品等，向其推荐同类物品；对历史行为资料进行聚类，同类用户之间相互推荐。



## 1.8 人工智能及其应用

人工智能：是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的科学。

研究内容：图像识别、自然语言理解、专家系统和机器人等。

模拟人脑思维的两个途径：

(1) 复制原型的内部结构以获得原型的类似功能，称之为结构模拟，仿生学采取的主要方法就是结构模拟法。

(2) 避开原型的内部结构，直接模拟原型的某些功能，称之为功能模拟。对于那些结构一时认识不清的原型，一般只能侧重于功能模拟。

### 1. 人工智能分类

(1) 弱人工智能：也称限制领域人工智能或应用型人工智能，只能解决特定领域的问题。弱人工智能表面看起来具有智能行为，但并不具备真正的思维能力和自我意识。

(2) 强人工智能：又称通用人工智能或完全人工智能，是可以胜任人类所有工作的人工智能。强人工智能需要具备以下能力：对不确定因素进行推理，制定决策，通过规则解决问题，知识表示，规划和学习，自主实现既定目标。

### 2. 人工智能的应用案例

用人工神经网络模拟划分平面的最优方程，拟合输入和输出之间的目标函数。

## 1.9 云计算的基本概念

云计算：是综合分布计算、并行处理和网络计算发展而来的，是一种新兴的计算模式。

“云”：是存在于互联网服务器集群上的资源，包括硬件和软件资源。本地计算机通过互联网向“云”发送请求，“云”将结果返给本地计算机。云计算以互联网为纽带，将调用功能的软硬件与实现功能的软硬件分开。

### 1. 云计算分类——按共享类型

(1) 公有云：是由第三方（供应商）提供的云服务，由云提供商完全承载和管理，可通过Internet访问，成本比较低廉。

(2) 私有云：是企业内部提供的云服务，在公司防火墙之内，由企业管理。

(3) 混合云：是公有云和私有云的混合，一般由企业创建，管理职责由企业和公有云提供商共同承担。

### 云计算分类——按服务类型

(1) 基础设施即服务：是将多台服务器组成的““云端”基础设施作为服务的模式，可以弹性分配计算资源。

(2) 平台即服务：是将软件研发平台作为服务的模式，将应用程序的基础结构视为服务，主要目的是支持应用程序的运行。

(3) 软件即服务：服务供应商将应用软件统一部署在服务器上，用户根据需求，通过互联网向服务供应商订购应用软件服务，服务供应商向客户提供软件，并根据用户所定软件的数量以及时间长短等因素收费。

(4) 数据即服务：将大数据中潜在的价值发掘出来，并根据用户需求提供服务。

## 2. 云计算特点

- (1) **虚拟性**：指对**计算资源**进行抽象，对上层应用或用户**隐藏**了计算资源的**底层属性**。既包括将**单个的资源**（例如一个服务器，一个操作系统，一个应用程序，一个存储设备）**划分成多个虚拟资源**，也包括将**多个资源**（例如存储设备或服务器）**整合成一个虚拟资源**。虚拟化技术根据对象可以分成**存储虚拟化**、**计算虚拟化**和**网络虚拟化**等。
- (2) **用户端负载和成本降低**：将应用开发与基础设施维护分离，**不需要为一次性任务或不常见的负载状况**准备大量设备。例如，学生选课系统在选课期间需要使用大量的 CPU、内存和网络带宽资源，可以使用云计算服务进行配置临时升级，提高服务器性能。

## 第二章 计算机系统基础知识

### 2.2 计算机系统及其工作的基本原理

#### 2.2.1 计算机系统构成

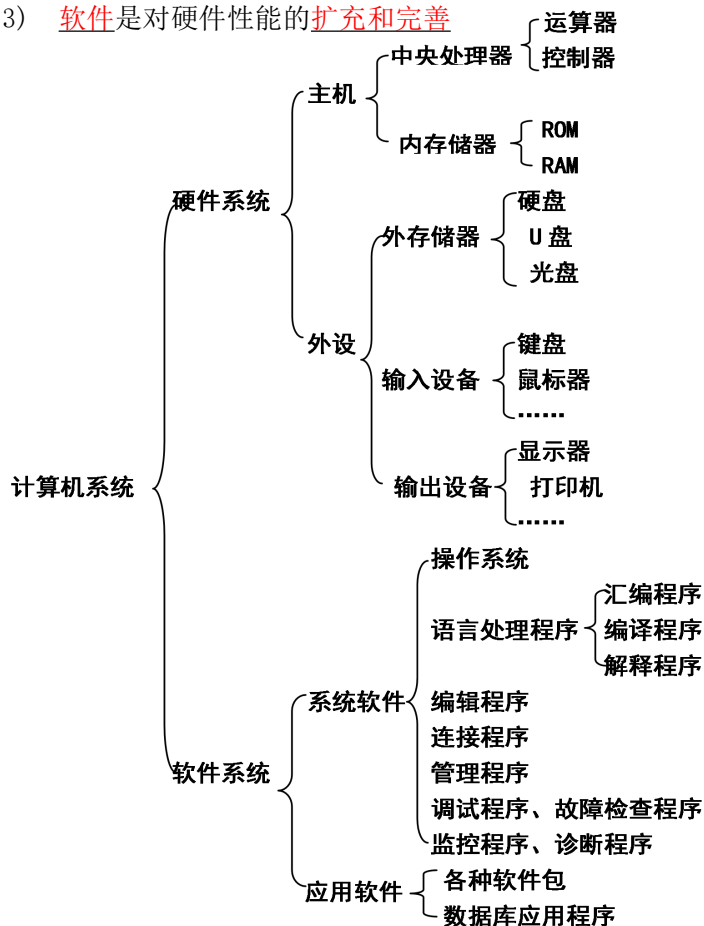
一个完整的计算机系统由**硬件系统**和**软件系统**两部分组成。

##### 计算机硬件

- 1) 是指计算机系统中看得见、摸得着的**物理实体**。
- 2) 其基本功能是在计算机程序的控制下，完成数据的**输入、存储、运算和输出**等一系列操作。
- 3) 通常将仅含有硬件的计算机系统称为硬件系统，也称为**裸机**。

##### 软件系统

- 1) 是指为运行、管理和维护计算机系统而用计算机语言设计的**各种程序**、程序操作所需要的**数据**和**相关文档**的总称。
- 2) **硬件**是运行计算机系统的**物质基础**
- 3) **软件**是对硬件性能的**扩充和完善**





### 2.2.2 计算机硬件系统构成

- 1) 计算机硬件系统主要由控制器、运算器、存储器、输入和输出设备五大部件组成。
- 2) 控制器和运算器是计算机硬件系统的两个核心部件，通常将控制器和运算器统称为中央处理器（CPU）。在微型计算机中，将这两个部件集成在一个芯片中，称之为微处理器。
- 3) 存储器主要指内存储器，也称之为主存储器，简称内存或主存。

### 2.2.3 计算机软件系统

软件是计算机硬件完成一定任务所需的程序、数据和资料，即为运行、管理和维护计算机所编制的各种程序和文档的总和。

软件可分为系统软件和应用软件两大类

#### （一）系统软件

- 是指用于计算机系统内部管理、维护、控制和运行，以及计算机程序编辑、翻译和装入的软件。
  - 它为应用软件提供运行平台，为开发应用系统提供工具。
  - 系统软件包括操作系统、语言处理系统、数据库管理系统和服务程序四大类。
- 1) 操作系统（OS）：
    - 是为了合理方便地利用计算机系统而对其资源进行分配和管理的软件。
    - 计算机软件系统的核心
    - 是软件之间、硬件之间、软件与硬件之间以及用户与计算机之间的接口目前微型计算机上运行的操作系统有 Windows、Linux、UNIX
  - 2) 语言处理系统：
    - 通过计算机语言可以编写程序，控制计算机完成预定的任务。
    - 计算机语言分为机器语言、汇编语言和高级语言（如 C 语言、Visual Basic 和 Java 语言等）。
  - 3) 数据库管理系统
    - 是管理数据库的软件，主要面向解决数据处理的非数值计算问题。
    - 目前多用于档案、财务、图书资料及仓库管理等。
    - 数据处理的主要内容为数据存储、查询、修改、排序和分类等。
    - 微型计算机中常用的数据库管理系统有 Access、Oracle、MySQL 和 SQL Server 等。
  - 4) 服务程序
    - 完成一些与管理计算机系统资源及文件有关的任务。
    - 基本可以分为 5 种：诊断程序、反病毒程序、卸载程序、备份程序、文件压缩程序。

#### （二）应用软件

是针对某一应用目的而开发的软件

分为两大类：通用应用软件和专用应用软件

通用应用软件支持最基本的应用，广泛地应用于各个专业领域。

专用应用软件专用于某一个专业领域，例如，医院挂号、铁路售票、学生选课

## 2.3 中央处理器

### 2.3.1 CPU 的主要组成部件

CPU 主要由控制器、运算器组成。

#### （一）控制器

负责从存储器中取出指令、翻译指令、分析指令，向其他部件发出控制信号（指令），

控制、协调计算机各组件自动、连续地执行指令，控制整个计算机有条不紊地工作。

- 通用寄存器组：微处理内部的一些存储单元
- 指令寄存器(IR)：存正执行的指令
- 程序计数器(PC)：存下条指令的内存地址
- 指令译码器(ID)：分析指令→操作控制信号
- 操作控制器(OC)：将操作控制信号发送给相关部件

控制器的主要特点是采用内存程序控制方式进行工作，即要使计算机运行，必须在内存存储器中存有机语言程序（由一系列指令组成），由控制器依次读取指令并执行。

执行一条指令分为取指令、分析指令、执行指令和调整程序计数器的值 4 个步骤

## （二）运算器

是对二进制数据进行运算的部件，在控制器的控制下执行程序中的运算型指令，完成各种算术和逻辑运算。

- 算术逻辑运算单元 ALU：是运算器的核心，执行算术和逻辑运算，以加法器为基础
- 累加器 A：用于存储算术逻辑单元运算的两个操作数之一，并且保存当前运算的结果
- 暂存寄存器：用于存储算术逻辑单元运算的另一个操作数，但不用于保存运算的结果
- 标志寄存器(FR)：自动记载算术逻辑单元运算结果的重要状态。为进一步处理累加器 A 中的数据提供依据

## 2.3.2 CPU 的多核技术

内核是 CPU 最重要的组成部分，其主要功能是进行计算数据、发出指令、处理数据等操作。

多核技术就是将多个 CPU 内核集成在一个处理器中，每个内核都能独立地完成操作，从而可以并行处理数据，这样有效地提高了数据处理能力和运行速度。

虽然多核 CPU 提高了计算机的性能，但并不是核数越多就越好，核心太多，系统不能进行合理分配，运算速度反而会减慢。

## 2.4 存储器及其分类

存储器的主要性能指标是存储容量。

1、存放一位二进制数（0 或 1）称为位（bit，简称为 b），是度量数据的最小单位。

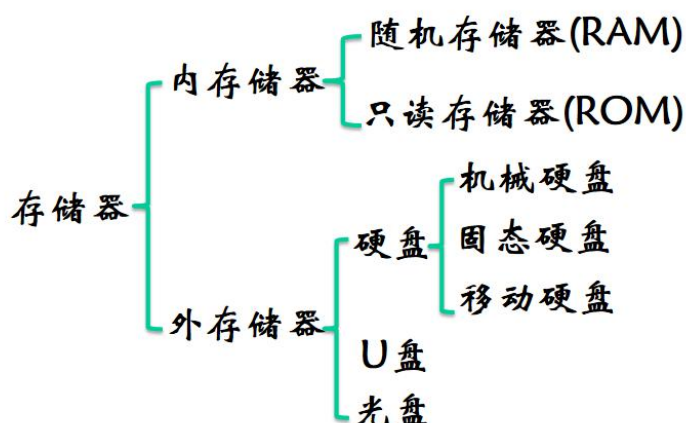
2、8 个二进制位组成一个字节（Byte，简称为 B），是信息组织和存储的基本单位。

$$1\text{ B}=8\text{ b}$$

$$1\text{ KB(千字节)}=2^{10}\text{ B}=1024\text{ B}$$

$$1\text{ MB(兆字节)}=1024\text{ KB}(=2^{20}\text{ B})$$

存储器一般分为内存储器和外存储器两种。



### 2.4.1 内存储器

内存储器用于存放正在执行的程序指令和数据，具有存取速度快、可直接与 CPU 交换信息等特点。

内存储器包含随机存储器 (RAM)和只读存储器 (ROM)两种。

#### (一) 随机存储器 (RAM)

保存计算机正在执行的程序和数据，是临时存储区域，系统断电后信息丢失。

根据存储单元的工作原理不同

静态随机存储器 SRAM (Static RAM): 高速缓存

动态随机存储器 DRAM (Dynamic RAM): 内存条

##### 1) 高速缓存 (Cache)

- 位于 CPU 与主存之间的随机存储器
- 其容量并不大，但信息交换速度很快
- 主要用于解决 CPU 运行速度与主存储器的访问速度不匹配问题，提高 CPU 的工作效率。
- 微型计算机一般都带有两级 Cache。
- Cache 与 CPU 封装在一个芯片中，不能随意选择或更换。
- 启动计算机时，在 CMOS 设置程序中可以看到 Cache 的容量。

##### 2) 内存条

- 内存条是基本内存储器
- 运行用户程序时，主要使用这部分内存储器，通常也称之为物理内存或主存储器。
- 内存容量及性能是影响微型计算机性能的重要因素。

##### 3) CMOS 芯片

- 是系统主板上一个可读写存储器 (RAM) 芯片
- 存储容量较小，靠电池供电
- 用于保存用户通过 BIOS 系统设置程序配置的系统日期及时间、开机密码和某些硬件参数等信息。

#### (二) 只读存储器 (ROM)

在出厂时已写入含有系统初始化程序、操作系统引导程序及多种硬件驱动等程序，并被固化。系统断电后信息不丢失。

BIOS (Basic Input/Output System, 基本输入输出系统)

存储着一组系统程序，当计算机启动时，首先运行 BIOS 中的系统初始化程序对系统进行检测，引导操作系统进入内存，使计算机正式工作

### 2.4.2 外存储器

外存储器中的数据只有先调入内存储器后才能由 CPU 访问和处理，它主要用于存放需要长期保存的信息，其特点是存储容量大、成本低，关闭电源后存储的信息不丢失，但存取速度比内存储器慢。

外存储器主要有机械硬盘、固态硬盘、移动硬盘、U 盘和光盘。

#### (一) 机械硬盘 (HDD)

机械硬盘也就是传统硬盘 (Hard Disk Drive, HDD)，属于磁性材料存储器，是目前各种计算机的主要外存设备。

磁盘存储介质以铝合金或塑料为基体，两面涂有磁性胶体材料。通过电子方法可以控制

磁盘表面磁化，以达到记录信息（0 和 1）的目的。

机械硬盘内部主要由磁盘盘片、读写磁头和主轴组成

盘片在主轴带动下高速旋转，读写磁头在盘片上存放数据的区域快速移动，进行读写数据。一个机械硬盘内可以有多张磁盘盘片，每张盘片上被划分若干磁道和扇区。

- 磁道：是磁头读/写数据的路径。
- 扇区：每个磁道分为若干小区段，一个小区段被称为一个扇区。每个扇区容量 512 个字节。
- 柱面：所有盘片上相同半径的磁道组合在一起，称为一个柱面
- 硬盘总容量=磁头数×柱面数×扇区数×每扇区内记录数据量
- 主轴转速：决定着硬盘内部数据传输率；区别硬盘档次的重要标志。
- 硬盘接口：机械硬盘的接口有数据接口和电源插口两部分。

#### （二）固态硬盘（SSD）

采用固态电子存储芯片阵列，内部没有可以移动的部件，具有抗震性强、噪音低、读取数据快、耗能低、体积小等特点。具有明显的缺点：容量小、价格偏高，其内部硬件一旦损坏，数据难以恢复。SATA 目前是固态硬盘的主流接口。

#### （三）移动硬盘

可以存储大量数据，携带方便。通过 USB 接口 与计算机相连，可以进行热插拔。

#### （四）U 盘

也称为闪存，属于半导体移动存储器，支持热插拔等优点，容量已达 GB 级。通常插入微型计算机的 USB 口 即可使用。

#### （五）光盘

利用光学存储介质存储数据，通过激光原理对数据进行读写。

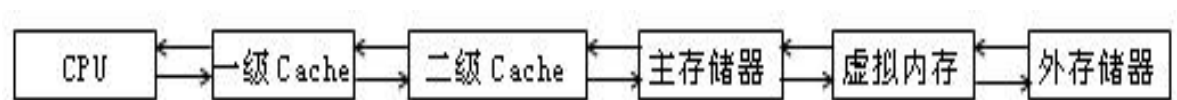
### 2.4.3 存储器间的信息交换

#### I 虚拟内存

通过软件的方法，将物理内存和一部分外存空间构造成一个整体，提供一个比实际物理内存容量大得多的存储空间。

#### II 存储器间的信息交换

微型计算机中，CPU 与各类存储器之间的信息交换关系如图所示。



距离 CPU 最近的存储器，具有较高的信息访问速度，但制作成本较高，存储容量比较小。一级 Cache 的访问速度最快，接近于 CPU 的运行速度。

实现存储器多级管理器的目的是尽量减少 CPU 访问低速存储器的次数，节省 CPU 的时间。

## 2.6 微型计算机系统主板及其作用

### 2.6.1 系统主板

系统主板是一块电路板，是用于连接和插接各个部件的母板。

系统主板通过总线将 CPU、内存条等部件及外部设备有机地连接起来，形成一套完整的硬件系统。

系统主板是微型计算机系统的硬件框架，它决定了整台计算机的硬件性能和档次。

### 2.6.1 常见部件及其作用

#### （一）主板上部件的安装方式

**固化方式：**是将部件焊接在系统主板上，如控制芯片组（南、北桥芯片）、BIOS 芯片等，其缺点是部件与系统主板一体，不容易更换部件。

**插接方式：**是将部件（如 CPU、内存条等）带金属丝（总线）的一边插入到系统主板上的相关插槽中，若部件破损时不会导致更换系统主板或其他部件。

#### （二）常见部件（了解用途即可，南北桥芯片重点）

##### ● CPU 插座

多金属孔（与主板的总线相通）、零插拔力的方形插座，而 CPU 的背面有与插座中插孔数一致的金属针（与 CPU 内部总线相连）。

##### ● 南桥芯片

主要协助 CPU 对键盘控制器、USB（通用串行总线接口）、磁盘驱动器和电源管理控制器等进行管理

##### ● 北桥芯片

主要协助 CPU 对内存储器、显示器、PCI 插槽等进行管理。

南/北桥芯片是系统主板的核心部件。主要用于协调和控制 CPU、内存和各部件间的数据传输。

##### ● 内存插槽

用来插内存条的部件，同时内存插槽也决定了主板所能支持的内存种类和容量。通常主板上会有 2—8 个内存插槽，内存容量可达到 8—128GB。目前的主流内存主要是 DDR3 和 DDR4 两种类型，所使用的插槽类型是 DIMM（双列直插式存储模块）插槽

##### ● IDE 插槽

曾经是普遍使用的存储设备接口，主要用于连接硬盘和光驱的并行接口，即在数据传输时一次可传输 16 位信号。

##### ● SATA 接口

一种基于行业标准的串行硬盘接口，具有更快的传输速度和热插拔功能。由于采用单数据通道，使数据通信线更窄，为机箱腾出更多的空间。SATA 接口有 SATA II 接口、SATA III 接口等标准，其中，SATA II 接口的传输率为 3Gbps，SATA III 接口的传输率为 6Gbps。

##### ● PCI-E 插槽

PCI-Express 简称为 PCI-E

主要优势就是数据传输速率高，支持热拔插。PCI-E 有 PCI-E X1 到 PCI-E X16 等多种规格。PCI-E X1 用于 扩展声卡、网卡等低速设备，而 PCI-E X16 常作为 显卡插槽。PCI-E X1，由南桥提供支持；PCI-E X16，由北桥提供支持。

##### ● PCI 插槽

是传统扩展插槽，主要用于插接一些扩充微型计算机功能的板卡。新式主板很少有此插槽。

### 2.6.3 常见外部接口及其作用

主板上常见的外部接口

#### USB 接口

是通用串行总线接口，支持热插拔。常用的 USB 接口有 Type-A 和 Type-C 两个类型。多口集线器。

#### VGA 接口

VGA 是一种输出和传递模拟信号的电脑显示标准，用于将显卡产生的数字信号转换成

模拟信号并显示输出。VGA 接口通常与显示器或投影机相连接，主要用于老式的电脑输出。

若想将 VGA 接口的电脑与 HDMI 接口的显示设备相连，需使用 VGA 转 HDMI 转换器；若将 HDMI 接口的笔记本电脑连接到 VGA 接口的投影机上，需用 HDMI 转 VGA 转换器。

DVI 接口

DVI 是一种国际开放的数字视频接口标准。通过 DVI 接口传输的是数字信号。主要应用于连接液晶显示器、数字投影机等显示设备。

Display Port 接口

是一种高清数字显示接口标准，可以连接显示器和家庭影院。允许音频与视频信号共用一条传输线。

HDMI 接口

是一种数字化视频/音频接口技术，适合影像传输的专用型数字化接口。可同时传送音频和影像信号。

PS/2 接口

用于连接键盘或鼠标，即将被 USB 接口取代

网卡接口

即 RS-45 接口，用于计算机与网络之间的连接。

声卡接口

一般有 5 个接口，用 5 种颜色区别不同的功能。

- 绿色为音频输出端，用来连接音箱或耳机。
- 粉色为麦克风接口，用来连接麦克风。
- 蓝色为音频输入端口，可以连接 MP3、MP4、手机等外置音频。
- 黑色为后置环绕喇叭接口，当使用四声道以上时需要用到。
- 橙色为中置/重低音喇叭接口，当使用六声道以上时需要用到。

## 2.7 计算机的主要性能指标

### 1. 字长

指 CPU 一次能处理二进制数的位数。字长总是 8 的整数倍。

### 2. 运算速度

1) CPU 频率：计算机执行指令的速度主要取决于 CPU 的频率，微型计算机一般采用主频来描述运算速度。

- CPU 主频率：CPU 执行指令时，要将一条指令分解成一系列操作步骤。
- CPU 外频率：是 CPU 与周边部件（如内存存储器和芯片组等）之间传输数据的频率

2) 每秒执行指令条数

### 3. 主存储器

是 CPU 直接访问的存储器，要执行的程序和所要处理的数据都存放在主存中。

### 4. 外存储器

常指硬盘(包括内置硬盘和移动硬盘)，其容量体现了计算机存储信息的能力。

### 5. 硬件扩充能力

指允许配接哪种类型设备，可配接多少设备，是否允许扩充或更换部件等。

## 2.8 数值型数据的存储及其运算

### 2.8.1 机器数

计算机内部信息只有 0 和 1 两种形式。规定最高位为符号位，用 0 表示“+”号；用 1 表



示“-”号，称为数符，其余位表示数值。

将机器内存储的带符号数称为机器数，而由正、负号加绝对值表示的实际数称为真值数。

(-193) D = (-11000001) B

机器内用 16 位时，其机械数为 10000000111000001

标绿符号数/数符 标红数值

### 2.8.2+2.8.3

机器内部小数点是隐含的，位置可以固定，也可以可变动。前者称为定点数，后者称为浮点数。

#### 定点数

##### 1 定点整数

小数点位置固定在数值最低位后面，表示整数

##### 2 定点小数

小数点位置定在数值最高位前面，表示纯小数

当用 M 位二进制数存储数据时

定点整数 N 的取值范围是

$$-(2^{M-1}-1) \leq N \leq (2^{M-1}-1)$$

定点小数 N 的取值范围是

$$-(1-2^{-(M-1)}) \leq N \leq (1-2^{-(M-1)})$$

无符号整数

省略符号位的正整数被称为无符号整数。

在计算机中存储无符号整数时，不留符号位，所有数位都用于存储数值。

#### 浮点数

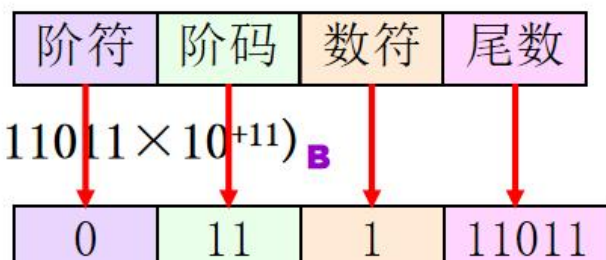
任何R进制数都可以表示成： $N = \pm S \times R^{\pm J}$

数符 ←  $\pm$  尾数 ← S 阶符 ←  $\pm$  阶码 ← J

二进制规范化形式规定：尾数为定点小数且尾数值的最高位为 1

#### 浮点数存放形式

$$(-110.11)_B = (-0.11011 \times 10^{+11})_B$$



### 2.8.7 逻辑运算

#### 1 真 0 假

与运算 “×”、“∧” 或 “and”

或运算 “+”、“∨” 或 “or”

非运算 “ $\bar{A}$ ” 或 “not”

## 第四章 VB 程序设计

本章全部编程示例中<>[]符号实际编程时全部去除

### 4.1 VB 程序设计语言简介

BASIC: **B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode, 结构化程序设计语言, 初学者通用符号指令代码, 称基本 Basic。

**V**isual **B**asic: 可视化程序设计语言, 用于开发 Windows 环境下的各种程序, 处理文本、图像、动画和声音等多媒体数据。

#### 4.1.2 创建应用程序的步骤

1. 创建一个工程文件: 启动 VB, 在新建工程对话框中选择标准 EXE, 单击打开按钮。
2. 在窗体中添加对象: 拖动工具箱中图标到窗体中, 添加对象。
3. 设置属性: 用属性窗口设置对象属性值。
4. 编写事件代码: 为相关控件编写事件代码。
5. 运行调试程序: 单击运行菜单→启动, 或者单击工具栏中的启动按钮, 运行程序。

### 4.3 窗体设计

窗体(Form): 也称窗口, 是应用程序的主要界面。

#### 4.3.1 创建窗体

方法 1: 单击工程菜单→添加窗体, 再双击窗体选项。

方法 2: 从工程资源管理器的右击菜单中选择添加→添加窗体, 双击窗体选项。

#### 4.3.2 属性及其设置

对象的属性: 描述对象的特征, 可以设置和引用属性的值。

修改对象的属性值以改变对象的特征。引用对象的属性值以实现对象间信息交换。

##### I 常用属性

- (1) Caption: 字符串型, 窗体标题栏上的文字内容。
- (2) Enabled: 逻辑型, 对象是否可操作。True 可操作 False 不可操作
- (3) Height: 数值型, 窗口的高度。
- (4) Icon: 字符串型, 控制菜单图标文件名(ICO)。
- (5) Left: 数值型, 左边列坐标。
- (6) MaxButton: 逻辑型, True: 有最大化按钮。
- (7) MinButton: 逻辑型, True: 有最小化按钮。
- (8) Name(名称): 字符串型, 窗体名称。
- (9) StartPosition: 数值型, 窗口开始位置。如: 0 手动; 2 屏幕中心。
- (10) Top: 数值型, 上边行坐标。
- (11) Visible: 逻辑型, 对象是否隐藏。
- (12) Width: 数值型, 窗口宽度。
- (13) WindowState: 数值型, 窗口初始状态。0 正常, 大小由 Height 和 Width 决定; 1 最小化; 2 最大化。

##### II 设置对象的属性

1. 静态设置: 在设计窗体时设置属性的值。

方法: 右击对象→属性窗口, 在属性窗口中设置属性值。

2. 动态设置: 程序运行过程中, 执行语句修改属性值。

#### 4.3.3 事件及其代码

**事件**：对象响应操作的反映机制，是操作与程序代码的接口。设计事件代码，实现对象操作的功能。

##### 1. 编写事件代码

双击对象或从对象选择框选对象名，从事件选择框选择事件名，在开始语句 Private Sub 和结束语句 End Sub 之间写事件代码。

##### 2. 窗体的常用事件

- 1) Activate：窗体变为活动窗口时触发。
  - 2) Click：单击触发。
  - 3) DbClick：双击触发。
  - 4) Deactivate：窗口由活动变为非活动触发
  - 5) GotFocus：获焦点(变活动窗口)触发
  - 6) Load：调用(运行)窗体时触发。
  - 7) LostFocus：对象失去焦点(窗口由活动变非活动)时触发。
  - 8) Resize：窗口大小发生变化时触发。
  - 9) Unload：关闭窗口触发。
- 触发后执行该事件下的程序代码。

#### 4.3.4 方法及其调用

**方法**：是一种依附于对象的子程序，调用方法执行操作。

1. 方法的调用：<对象名>.<方法名> 如 Form1.Hide

##### 2. 常用方法

- 1) Show：装载和显示窗口，成为活动窗口
- 2) Hide：隐藏窗口，即 Visible 设为 False。
- 3) Circle (x,y),<半径>,<颜色>：以(x,y)为圆心画彩色圆。
- 4) Print：在窗口上显示文字。
- 5) CLS：清除窗口上图形或文字。
- 6) SetFocus：使对象得到焦点。

### 4.4 窗体中的控件设计

**控件**：通过工具(基类)添加到窗体中的对象，控件也有属性、事件和方法。

**程序中的对象**：指控件、窗体和菜单等。

#### 4.4.1 向窗体中添加控件：

单击(选定)工具箱中的工具，在窗体上拖动鼠标，画出控件。

拖动控件可改变位置，拖动尺寸控点，可改变大小。

#### 4.4.2 基本控件及其作用（控件作用和属性 事件 方法名字对应的作用重点记）

- ① 标签(Label)：用 Caption 属性显示文字。
- ② 图像(Image)：显示图像。  
picture：图像文件名。  
Stretch：逻辑，True 自动调整图像大小。
- ③ 命令按钮(CommandButton)：实现控制功能。  
Caption 属性：按钮上显示的文字。

Click 事件：单击触发，执行程序代码完成功能。

④ 文本框(TextBox)：显示和输入数据。

常用属性

MultiLine：逻辑，True 多行，False 单行。

Text：字符串，文本框显示和输入的数据。

Alignment：数值，对齐方式。0：左对齐，1：右对齐，2：居中。

常用事件：

Change 事件：文本框中数据变化时触发。

KeyPress 事件：键盘按键时触发，用参数 KeyASCII 得到按键的 ASCII 值。如：空格键为 32；回车键为 13 等。

常用方法：

SetFocus：焦点移至文本框上。

⑤ 列表框(ListBox)：设计列表框。

常用属性

List：List1.list(1)表示列表框 List1 第 2 行的选项内容

ListCount：列表框中列表项的项数。

ListIndex：最后选中的列表项序号。

Text：最后选中的列表项内容。

常用事件

Click：单击触发。

DbClick：双击触发。

常用方法

AddItem：为列表框添加列表项。

RemoveItem：删除列表项。

Clear：清空列表框中的内容。

⑥ 组合框(ComboBox)：设计下拉框。

Style 属性值：0：可在文本框中输入，也可在列表中选；1：只有文本框，没有列表框；2：只可在列表中选。

⑦ 框架(Frame)：起容器作用。先添框架，再向框架内添其他控件，作为一组控件。

Caption 属性：框架上显示的文字。

⑧ 复选框(CheckBox)：可从多个选项中选定多项。

Caption 属性：框上显示的文字。

Value 属性：确定是(1)否(0)选定复选框。

⑨ 选项按钮(OptionButton)：从一组选项选定一项。

组：添加在窗体中的所有选项按钮是一组，添加到框架中的选项按钮单独成组。

Caption 属性：作用与复选框相同。

Value 属性：逻辑，True 选定，False 未选定。

⑩ **计时器(Timer)**：运行时隐藏，用于设计按一定间隔重复执行程序。

Interval 属性：触发 Timer事件的时间间隔(毫秒=1/1000 秒)。Interval 为 0时不触发 Timer 事件。

Timer 事件：按 Interval 设定的时间间隔，重复执行 Timer 事件的程序代码。

#### 4.5 数据表示形式和表达式运算规则

表达式是程序中完成运算的工具和手段。常量、变量、对象的属性和函数是简单表达式，用运算符将表达式正确连接起来，构成更复杂的表达式。

##### 4.5.1 数据类型

数据：程序处理的对象，是程序进行运算的基础。

数据类型：根据运算性质、范围和存储方式，VB 有数值、字符、日期和逻辑等类型。

##### 4.5.2 常量的表示形式

常量是程序运行过程中不变化的量，有常数和符号常量两种形式。

###### 1. 常数

- 1) 整型：由数字 0 ~ 9 和正(+)负(-)号组成。如：100、-5 。
- 2) 长整型：数末尾加&。如：100& 。
- 3) 单精度型：数末尾加!和 E 指数形式的数。如：10!、-5.7!、3.5E-2。
- 4) 双精度型：数末尾加#、D 指数形式或直接写带小数点的数。如：3.14、7#、-5D3 。
- 5) 字符串型：用半角双引号引起字母、数字、汉字及标点符号等。字符串中有双引号时，用连续两个双引号。如："计算机"、"他说："明天来"""
- 6) 日期型：用#括起日期和时间数据。如： #2012-10-1 #。
- 7) 逻辑型：只有 True 和 False 两个值。

##### 4.5.3 变量及其赋值

变量是程序运行中可随时改变的量，用于存储程序处理的原始数据或中间结果。VB 中有内存和对象属性两种变量形式。

###### 1. 变量命名

- 1) 必须以字母或汉字开头，由英文字母、汉字、数字或下划线组成，字母不区分大小写，最大长度为 255 个字符。如：N1、学号
- 2) 不能用 VB 的保留字(系统已用的单词，如：String、Caption、True 等)作变量名。

###### 2. 变量声明

为变量命名及说明数据类型。

语句格式

Dim <变量名 1> As <类型> .....[, <变量名 n> As <类型>]

###### 3. 变量赋值

存储或修改变量中的数据。

格式：<变量名> = <表达式>

<对象名>.<属性名> = <表达式>

数据类型不一致时，已声明(Dim 语句)的变量或对象属性，按变量或属性的类型转换表达式的值；未声明的变量，按表达式值的数据类型确定变量数据类型。

注：一行写多条语句时，语句间用半角冒号(;)分隔。如：a = Text1.Text : b = Text2.Text

#### 4. 数组

在内存中用表存储一组数据。用一个名定义多个同类型变量，每个变量称一个数组元素。

Dim Sc(7) as Integer ' 数组 Sc 含 8 个整型元素

Dim NA(3,2) as Sting ' 数组 NA，含 4×3 个字符型元素

SC(4) = 5

Na(0,1) = "计算机"

#### 4.5.4 表达式

常用基本运算符号

算术运算：+(加)、-(减)、\*(乘)、/(除)、Mod(求余数)和^(乘方)等，优先级遵循数学规定。

字符运算：&或+，两个字符串接成一个。

日期运算：

<日期 1> - <日期 2>：日期相减，得相隔天数。

<日期> ± <整数 n>：n 天后或前的日期。

关系运算：运算结果为逻辑型。运算符：=(等于)、<>(不等)、<(小于)、>(大于)、<=(小于或等于)、>=(大于或等于)。

逻辑运算：AND(并且)、OR(或者)、NOT(否定)。优先级 NOT 最高，OR 最低。

标准函数：语言提供的内部函数。

运算优先级由高到低：算术、日期、字符、关系、逻辑。同级别从左到右运算，圆括号级别最高。多层圆括号，里层优。

### 程序设计

语句的执行顺序与操作的数据和程序的控制结构有关。程序控制结构有顺序、分支(选择)和循环(重复)3 种。顺序结构即执行顺序与程序编写顺序一致。

#### 4.6 分支程序设计

##### 4.6.1 简单分支结构

分支结构：根据条件(逻辑值表达式)确定程序执行取向，也称选择结构。

##### 1. 简单分支语句

语句格式 1

If <表达式> Then <语句序列 1> [ Else <语句序列 2> ]

表达式值为 0 为 True，执行序列 1；非 0 为 False，执行序列 2，语句必须在一行上，若写多条语句用半角冒号：分隔

语句格式 2

If <表达式> Then

<语句序列 1>

[Else

<语句序列 2>]

End If

##### 2. 分支函数

IIf(<表达式>, <表达式 1>, <表达式 2>)

如 Print IIf(X>=0, SQR(X), "不能开平方")

若表达式的值为 True，则值为表达式 1的值，否则，值为表达式 2的值。

##### 3. MsgBox 对话框函数



Msgbox(<字符式 1>[,<数值式>[,<字符式 2>]])  
如 I=MsgBox("要解方程吗 ? ", 256 + 4 + 32, "询问")

#### 4.6.2 多分支结构

##### 1. 多分支 If 语句格式

```
If <表达式 1> Then  
    <语句序列 1>  
Elseif <表达式 2> Then  
    <语句序列 2>  
.....  
Elseif <表达式 n> Then  
    <语句序列 n>  
[ Else  
    <语句序列 n+1> ]  
End If
```

#### 4.7 循环程序设计

循环结构又称重复结构，由条件确定是否重复执行某段程序代码。循环体即重复执行的程序段。重复执行次数由循环控制条件决定。可简化程序，适用有规律重复计算或操作。有当型和直到型两种循环结构。

##### 4.7.1 For 循环结构

为当型循环

```
For <循环变量>=<初值> To <终值> [Step <步长>]  
    <循环体代码>
```

```
Next [<循环变量>]
```

开始时，循环变量值为初值；执行一次循环体(到 Next)，循环变量值加步长，转开始语句；循环变量值超过终值结束循环。Exit For 提前结束循环。

##### 4.7.2 Do While 循环

为当型循环

```
Do While <循环条件>  
    <循环体代码>
```

```
Loop
```

若条件成立，则执行循环体；否则，执行 Loop 后面语句。Exit Do 提前结束循环。

Do While与 Loop必须成对。

循环体中要有执行到的、条件趋于 False的语句，避免永远执行循环体，即死循环。

##### 4.7.2 Do 循环结构

Do 循环是一种直到型循环结构，至少执行一次循环体。

```
Do
```

```
    <循环体代码>
```

```
Loop Until <表达式>
```

先执行一次循环体，判断条件，条件为 False再执行循环体；条件为 True则终止循环。

应有能执行到的 Exit Do 或使表达式值趋于 True的语句，避免出现死循环。

## 4.8 子程序设计

功能比较复杂的程序一般由若干个功能独立的、相关联的程序模块组成。程序模块分为子程序、过程或函数。VB 中分为事件过程、Sub 子程序和 Function 函数 3 种程序模块。没有返回值的程序模块称过程；有返回值的程序模块称函数或用户定义函数。

设计函数格式

Function <函数名> [(参数列表)] [As <数据类型>]

<函数体>

<函数名> = <表达式>

End Function

定义函数时的参数称为形式参数，简称形参，用于接收调用程序传来的数据。

## 4.9 菜单程序设计生成可执行程序

### 4.9.1 菜单的组成

通常是下拉式菜单，由条形(顶层)菜单和弹出式菜单组成。

菜单项：菜单由标题文字和热键组成，弹出式菜单中还包含分隔线和快捷键。

### 4.9.2 菜单设计器的使用

#### 1. 创建菜单

在窗口设计器中，从窗口的右击菜单中选择菜单编辑器或单击“工具”→菜单编辑器选项。

标题：菜单项的文字。

热键：&<字母>。

快捷键：从组合框中选。

分隔线：半角减号—。

名称：菜单项内部名，即对象名。

#### 2. 编写菜单项事件代码

单击确定按钮，关闭菜单编辑器。

单击窗口设计器中的菜单项。

在菜单项的 Click 事件编写代码。

### 4.9.3 生成可执行程序

单击文件→生成<文件名>.exe选项项。

## 第五章 计算机网络技术

计算机网络系统是分布在不同位置且独立功能的计算机系统及辅助设备，通过通信设备和传输线路连接起来，由功能完善的网络软件（网络协议、信息交换方式、控制程序和网络操作系统）实现资源共享及信息通信的系统。

### 5.2 数据通信基础知识

计算机网络是计算机技术和通信技术密切结合的产物。数据通信技术是计算机网络的基础，通信网络为计算机之间的数据传输提供了必要手段。

#### 5.2.1 数据的传输方式

1) 通信系统组成：源站（发送者）、目的站（接收者）、传输的数据和通信线路。

2) 通信线路任务：将源站发送的数据迅速、可靠、准确地传输到目的站。

### （一）数据传输方式

① **并行传输**：**同时**传输**多位**二进制数，**每位**需要**一条信道**。速率高，适用近距离传输。如**微机内各类总线**。

② **串行传输**：逐位（比特）传输，只需要一条信道。速率低，适用远距离传输。现在网络上都是以**串行方式**传输信息，在此种方式下，又分**基带**和**宽带****两种**传输方式。

(1) 基带传输：将数字**1**或**0**用两种不同**电压**在线路上**传输**。**优点**是速率高和误码率低，**缺点**是传输距离短，常用于局域网。

(2) 宽带传输：经过**调制**的**模拟信号**在线路上传输，接收端经**解调**恢复成基带脉冲，常用于广域网。

### （二）传输速率与带宽

#### 传输速率

**数字信号**：用**每秒比特**表示，即每秒传送二进制位的数量。

**模拟信号**：**每秒**传送**脉冲数**，单位为波特。

#### 带宽

指**单位时间内**传输线路上**最大**传输**比特数**，是线路最大传输能力。

## 5.2.2 通信协议

在网络系统中，为了确保**通信双方**能正确并自动地**交换信息**，需要为特定的通信过程制定双方都能遵循的**规则**，这种规则就是**通信协议**。

**通信协议**也是计算机之间进行通信的语言，由于各种通信过程的规则有些差异，因此，在一个网络系统中有**许多种通信协议**。

例如，**NetBIOS**、**NetBEUI**、**IPX/SPX**和**TCP/IP**等都是 Windows 中的通信协议。

### （一）通信协议三要素

① **语法**：用于规定协议中所含**元素及顺序**，是协议内容的数据结构。如**数据报文协议格式**：

SOH	标题	STX	正文	ETX	校验码
-----	----	-----	----	-----	-----

② **语义**：是对协议中各个**元素**的**含义**说明或**规定**。例如：**SOH**表示报文的开始控制符，**STX**和**ETX**分别表示正文（数据）的开始和结束控制符号等，这些都是语义。

③ **时序**：指处理各种**事件**的**先后顺序**。

### （二）Windows 中的常用协议

① **NetBIOS**与**NetBEUI**：**NetBIOS**是网络基本输入 / 输出系统协议；**NetBEUI**是**NetBIOS**的扩展用户接口协议。

② **IPX / SPX**：是互连网络分组交换 / 顺序交换协议，是面向局域网、可路由的高性能协议。

③ **TCP/IP**：是**传输控制协议 / 网际协议**（Transmission Control Protocol/Internet Protocol），已经成为广域网和**Internet**的标准协议。

#### TCP/IP 协议设置

在 Windows 中，从桌面“**网络**”的右击菜单中选“**属性**”，在“**网络和共享中心**”对话框中，再从“**本地连接**”的对话框中选择“**属性**”，在“**本地连接**”对话框中可以**安装**和**设置**有关协议及其参数。

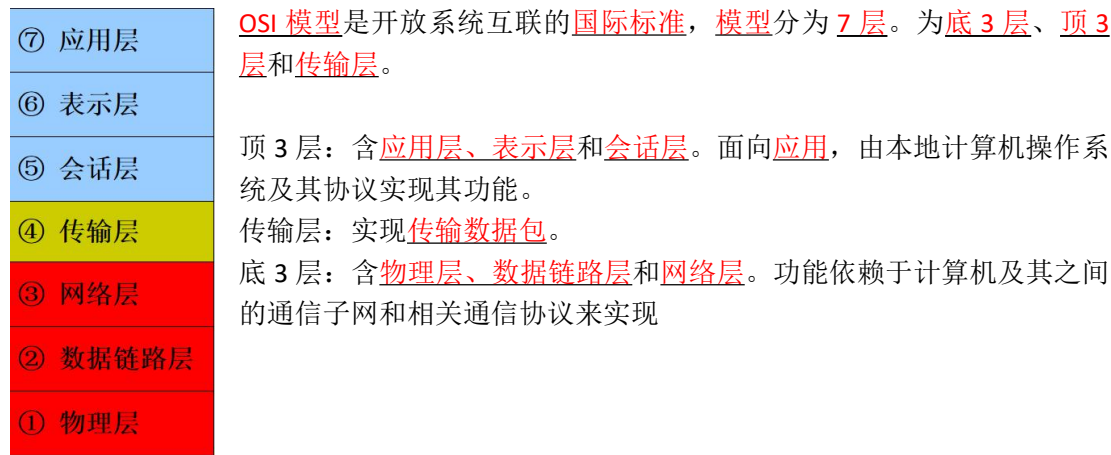
## 5.3 网络体系结构

网络**设计者**常采用把**通信问题**划分为**许多小**问题，然后为**每一个**问题设计**一个通信协议**。

每一个协议的设计、分析、编码和测试都比较容易。将计算机网络中的各层模块及其协议的集合，称为网络体系结构。

5.3.1 OSI 模型的基本思想

OSI 参考模型的结构



5.3.3 TCP/IP 体系结构

TCP/IP 体系结构是网络中的事实标准，适用于大型网络，已成为 Internet 的重要通信协议。



1. 网络接口层

将数据包发送到传输介质，并从传输介质接收包，传给网络层。

2. 网络层

核心是网际协议（IP），主要功能有 IP 寻址、路由选择和数据打包/拆包等。在 IP 协议中每个数据包中都含有目的主机的 IP 地址，可以作为独立的数据包处理。IP 协议是无连接的，开始传输数据之前，没必要先建立一条到达目的主机的通路，各个数据包可以通过不同的通路到达同一个目的主机，到达的顺序也可以与发送的顺序不一致。

3. 传输控制层

提供两个协议：

TCP 协议：提供可靠的面向连接的端到端服务。适合传输大数据量。

用户数据报协议 UDP：提供无连接服务，如小文件传输协议 TFTP。

#### 4. 应用层

提供一组常用应用层协议，与 OSI 类似。包括：文件传输协议 FTP、超文本传输协议 Http、简单邮件传送协议 SMTP 和域名系统 DNS。

#### 5.6 广域网

广域网（Wide Area Network，WAN）：又称远程网，覆盖范围大，从几十公里到数千公里，形成国际性的远程网络。

##### 5.6.2 Internet 基础知识

Internet 是世界上最大的互联网络，是计算机和通信两大现代技术相结合的产物，向全球提供信息服务。基于 TCP/IP 协议，采用客户机 / 服务器工作方式，不属于任何国家，是所有用户可使用和开发的信息资源。

##### 1. 常用协议

###### 1) TCP/IP 协议

规范数据传输格式及传送方法。

###### 2) 文件传送协议 FTP

用于将文件上传到其它机或下载到本地机。

###### 3) 远程登录协议 Telnet

控制远端主机的登录。

###### 4) 超文本传输协议 HTTP

使浏览器有统一的规则和标准。

##### 2. Internet 资源及典型服务

①信息资源：可以在网上找到任何方面的知识。如：天文地理、科学技术和新闻等。

②典型服务：可归纳为专题讨论（Usenet）、信息查询服务（Gopher）、广域信息服务系统（WAIS）、电子公告栏（BBS）、远程登录（Telnet）、电子邮件（E-mail）和文件传输（FTP）等。

##### 5.6.3 IP 地址

主机：连入 Internet 的计算机或设备。

网络系统中，主机之间要准确地相互通信，每台主机都应该有唯一标识。在 Internet 中，将这个唯一标识称为地址，地址有 IP 地址和域名两种表示形式。

###### （一）IPv4 地址

①IP 地址用 32 位二进制编码，每 8 位一组，以 “.” 分隔，实际应用中，用 4 个十进制数表示，单个数取值范围为：0~255，且用圆点分隔。

例如：202.198.091.60

②IP 地址包括两部分内容：一部分为网络标识，另一部分为主机标识。主机标识不可全 0 或全 1。

③IP 地址分为 A~E 类

A 类：前 8 位为网络地址，第一组第一位为 0，128 个网络地址

B 类：前 16 位为网络地址，第一组前两位为 10，有 16 384 个网络地址

C 类：前 24 位（前三组）为网络地址，第一组前三位为 110，有 2 097 152 个网络地址

###### （二）特殊 IPv4 地址

①单播地址：127.0.0.1 是单播地址，用于本机的网络软件测试和进程间的通信。

如：ping 127.0.0.1 用于测试本机安装协议的正确性。

② 指向网络的广播地址：主机地址每一位全 1的 IP 地址。

例如：子网掩码为 255.255.255.0，向 202.198.091.255 发送信息时，将信息传输到地址为 202.198.091 的网络中全部主机。

③ 受限广播地址：32 位全 1的 IP 地址，即向 255.255.255.255 发送数据，实质是向本地网（不跨出路由器）中的所有主机发送数据。

### （三）IP 地址分配

为确保 IP 地址在 Internet 上惟一性。要加入 Internet，必须申请到合法 IP 地址。

动态分配：随机分配一个 IP 地址。如：拨号上网用户使用动态 IP 地址

静态分配：固定 IP 地址。如：信息服务者使用固定 IP 地址。

### （四）IPv6

IPv4 中 IP 地址空间严重不足，为解决 IP 地址资源不足问题，IPv6 对 IPv4 进行了扩展和改进。

①IP 地址空间：由 32 位增加到 128 位，扩大  $2^{96}$  倍。

②报头：用 40B 固定长度的报头，使得处理数据报的速度更快

③优先级：根据数据报的性质为数据报定义优先级别，优先级别高的数据报优先传输。

### 5.7.3 子网掩码

子网掩码用 32 位二进制数编码，表示形式与 IP 地址类似。子网掩码左边连续“1”的个数与网络标识的二进制位数一致。

子网掩码的两个作用：

#### ①获取网络 IP 地址

通过主机 IP地址和子网掩码可以获取主机所在网络 IP地址和主机标识。

具体做法：将主机 IP地址与子网掩码按位进行“与”运算，即可得到网络 IP地址。

IP 地址： 10011111 11100000 00000111 10000001

159.224.7.129（B 类地址）

子网掩码： 11111111 11111111 00000000 00000000

255.255.0.0

网络地址： 10011111 11100000 00000000 00000000

159.224.0.0

这是 B 类 IP 地址和子网掩码，主机所在网络地址为 159.224，主机地址为 7.129。又如 C 类 IP 地址，的子网掩码是 255.255.255.0。

#### ②分割网络

适当地设计子网掩码可以将一个网络 IP地址分割成多个子网的 IP地址，以便解决距离较远的多个子网使用同一个网络 IP地址的问题。

如

某单位申请到一个 C 类网络 IP 地址：212.188.199，要在相距较远位置分别建立 4 个子网，每个子网中不足 60 台主机。

设计方法：把主机标识的前 2 位作为子网 IP地址的一部分，后 6 位作为新的主机标识。作为子网 IP 地址的 2 位可组合成：

00000000、01000000、10000000 和 11000000

即将子网掩码设计为：255.255.255.192

得到：212.188.199

212.188.199.64



212.188.199.128

212.188.199.192

4 个子网 IP 地址。

4 个子网 IP 地址，每个子网中的主机标识范围为：000001 至 111110， $(2^6 - 2) = 62$  个

（去除全 0 或全 1）可供分配的主机 IP 地址。其中：

子网 212.188.199 中主机 IP 地址范围：212.188.199.1~62

子网 212.188.199.64 中主机 IP 地址范围：212.188.199.65~126

子网 212.188.199.128 中主机 IP 地址范围：212.188.199.129~190

子网 212.188.199.192 中主机 IP 地址范围：212.188.199.193~254

### 5.6.6 域 名

在 Internet 中允许为主机起一个有意义且容易记忆的文字名称，将这个名称称为域名 (DN)。例如，www.jlu.edu.cn 中，www 是服务器主机名，jlu.edu.cn 是域名。

IP 地址与域名之间是一对多的关系，即一个 IP 地址可以有多个域名，但一个域名只能对应一个 IP 地址。

域名采用分层次方法命名，每一层都有一个子域名。子域名之间用圆点分隔，自右至左分别为最高层域名、机构名、网络名。

域名系统 (DNS)：解析域名与 IP 地址间对应关系。

例如，域名 jlu.edu.cn，解析为 IP 地址 202.198.16.80。

省略最高层域名，是在美国注册的网站。例如，微软公司的域名为 microsoft.com。

## 第七章 数据结构、算法及程序设计

计算机程序主要对数据进行加工和处理。通常程序中要说明数据的组织形式和存储方式，即数据结构；也要给出操作数据的步骤和方法，即算法。

### 7.1 数据结构基本概念

#### （一）数据结构定义

数据结构是指具有相同特征、相互之间有关联的数据集合。数据也称为数据元素或结点，现实世界中一切客观事物都可以抽象成数据元素。每个数据元素由一个或多个数据项组成  
数据结构是指带有结构特性的数据元素集合，主要研究：

- 1) 数据集合中数据元素之间所固有的关系，即数据逻辑结构；
- 2) 数据处理时数据在计算机中的存储关系，即数据存储结构；
- 3) 对数据所进行的操作，即算法。

#### （二）数据逻辑结构

① 数据逻辑结构为数据结构中数据元素之间所固有的关系描述成前后件（前驱与后继）关系。数据之间前后件关系是它们之间的逻辑关系，与它们在计算机中存储位置无关，因此将这种关系称为数据逻辑结构。

② 一个数据结构可以表示为：

$$S = (D, R)$$

S: 数据结构    D: 数据元素集合    R: D 中数据元素之间前后件关系集合，即数据逻辑结构

两个元素之间前后件关系用一个二元组表示

如季节数据结构可以定义成  $S = (D, R)$  其中：  $D = \{ \text{春, 秋, 冬, 夏} \}$      $R = \{ (\text{春}, \text{夏}), (\text{夏}, \text{秋}), (\text{秋},$

冬}}}

③ 数据之间有集合，线性，树形和图形4种基本逻辑结构。

线性结构：

数据元素之间一对一关系。除第一个结点无前件外，其他结点都只有一个前件。除最后一个结点无后件外，其他结点都只有一个后件。

树型结构：

数据之间一对多关系，一个结点仅有一个前件，可有多个后件。前、后件之间存在层次关系

图形结构：

数据元素之间多对多关系，一个结点可有多个前件和多个后件

集合：

是一种松散结构，数据元素之间的关系是属于一个集合，可用其他结构表示。

④ 根据数据结构中数据元素间前后件关系的复杂程度，将数据逻辑结构分为线性结构和非线性结构。

线性结构：有且只有一个开始结点和一个终端结点，并且每个结点最多只有一个前件和一个后件，线性结构也称为线性表。

非线性结构：有多个开始结点和多个终端结点，每个结点可以有多个前件和多个后件。

### （三）数据物理结构

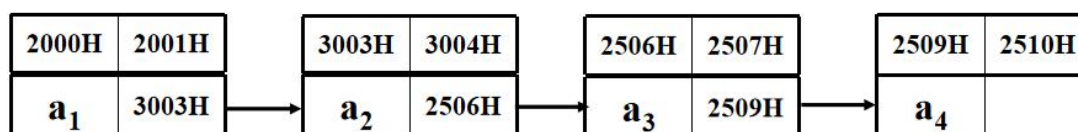
数据在计算机存储器中的存储方式称为数据存储结构（或数据物理结构）。数据结构中数据元素之间在计算机中的位置关系与逻辑关系不一定相同。在数据存储结构中，不仅要存放各个数据元素信息，还要存放数据元素之间前后件关系信息。数据存储结构是逻辑结构在计算机存储器中的表示。

数据元素在计算机中通常有四种存储方式，即顺序、链式、索引和散列，常用顺序存储结构和链式存储结构。

顺序存储结构是在内存中开辟一块连续内存单元用于存放数据，逻辑上相邻的结点在物理位置上也邻接，结点之间的逻辑关系是由存储单元相邻关系来体现。

数据	地址
$A_1$	2000H
$A_2$	2001H
$A_3$	2002H
$A_4$	2003H
.....	

链式存储结构：结点由两部分组成：数据域和指针域。数据域用于存放数据元素值，指针域用于存放前件或后件的存储地址，链式存储结构通过指针反映数据元素之间的逻辑关系。



## 7.2 算法的基本概念

### 7.2.1 算法定义

算法是解决问题的具体方法和步骤的描述，是一组有限运算序列。算法是定义在逻辑结构上的操作，独立于计算机，而算法必须在计算机上执行，因此算法的实现依赖于数据存储结构。

#### ① 算法的特征

可行性：采取的方法和步骤可行，结构另人满意。

确定性：算法中每个步骤结果都必须确定，不能产生歧义。

有穷性：算法必须由有限步组成，并能在有效时间内完成。

输入：执行算法时从外界取得必要的信息。一个算法可以有零或多个输入。

输出：算法得到的结果就是输出，没有输出的算法是没有意义的，一个算法应该有一个或多个输出。

#### ② 用于描述算法的工具很多，通常有流程图、N-S图、自然语言和伪代码等工具。

自然语言：带序号的人类语言描述方法。

特点：易懂却不直观，对复杂算法描述很困难，易产生歧义。

伪代码法：是一种假的代码—不能被计算机执行，可由计算机语言和自然语言构成。接近某种语言编写的程序，便于转换成程序。

流程图法：用一些图框、线条以及文字说明来形象地、直观地描述算法。优点：直观形象

缺点：算法复杂时，篇幅较多，费时且不方便修改。

N-S图：去掉箭头流程线、算法处理步骤写在大矩形框内，框内还可包含一些小矩形框，适于结构化程序设计。

#### ③ 算法评价

某一任务的算法设计得优与劣，将直接影响程序的运行效率、稳定性和可维护性。从4个方面评价算法：

正确性：算法本身没有语法错误，执行时输入正确数据能够得到正确结果。

可读性：算法要容易理解和阅读，容易实现，便于程序维护和完善。

健壮性：算法能够对各种输入数据给予适当处理和提示。

执行效率：算法执行的时间性能和空间性能。

解决同一问题的多个算法，执行时间短的算法时间效率高；占存储空间少的算法空间效率高。

## 7.3 线性表结构

数据逻辑结构分为线性结构和非线性结构。线性结构也称为线性表，栈、队列、数组和字符串等是特殊线性表；非线性结构包括树和图。

### 7.3.1 线性表

#### ① 线性表定义

线性表是一组特征相同数据的有限序列，表示为： $L=(a_1, a_2, a_3, \dots, a_n)$ 。线性表中数据元素个数  $n$  ( $n \geq 0$ ) 称为线性表长度。当  $n=0$  时，称为空表。在非空线性表中，每个数据元素都有一个确定位置，其位置取决于它的序号。

非空线性表的特征：每个数据元素  $a_i$ ，有且只有一个前件  $a_{i-1}$ （除  $a_1$  无前件），有且只有一个后件  $a_{i+1}$ （除  $a_n$  无后件）。

② 线性表通常采用顺序存储结构或链式存储结构。顺序存储的线性表也称为顺序表；链式存储的线性表也称为链表。

#### （一）线性表顺序存储

线性表顺序存储是指用一段连续存储单元存放表中数据元素。数据元素在存储空间中按

逻辑顺序依次存放。线性表的逻辑结构与存储结构相一致。

同一线性表中数据元素类型相同；一个数据元素占用 d 个字节；线性表的首地址 Addr

( $a_1$ ) 为 K, 则数据元素  $a_i$  的首地址  $\text{Addr}(a_i) = K + (i-1) \times d$

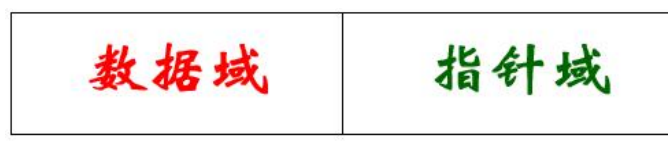
其中  $1 \leq i \leq n$

如：线性结构 {a1, a2, a3}, 其中每个数据元素占 2 个存储空间, 假设存储 a1 的首地址为 2000。那么 a2 地址为 2002。

在线性表中通过元素序号可以很方便地访问某一元素。程序设计中, 通常用数组来表示顺序表。

(二) 线性表链式存储

线性表链式存储是用一组存储单元(可以连续, 也可以不连续)存储线性表中数据元素。数据元素由两部分组成: 数据域(值域)和指针域。



数据逻辑结构和存储结构互相独立, 物理位置上相邻结点逻辑关系上不一定相邻。结点之间逻辑关系由指针域来确定。

1) 单链表

定义: 每个结点只有一个指针域的链表

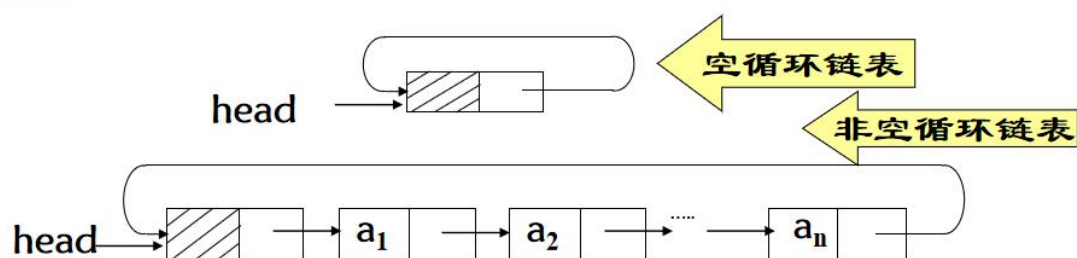
每个单链表都有一个头指针, 存放表中第一个结点的存储地址。

每个结点指针域存放后继结点的存储地址, 最后一个结点无后继结点, 指针域为空, 用 NULL 或 ^ 表示。



2) 循环链表

将单链表最后一个结点的空指针域改为指向该链表的第一个结点, 即首尾相连。循环链表中增设一个表头结点, 其数据域的值可以任意或根据情况设置, 指针域指向第一个结点。



循环链表特点:

- a. 从表中任一结点出发, 均可以找到其它所有结点;
- b. 在任何情况下, 带有表头结点的循环链表中 至少有一个结点存在, 从而使空表和非空表运算统一。

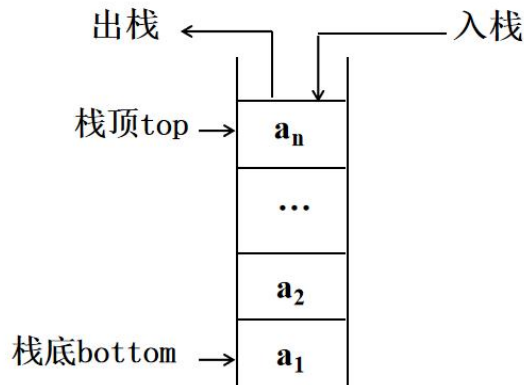
3) 循环链表运算与单链表区别:

对单链表进行操作时, 要判断是否是表尾, 即指针是否为 NULL;

而对循环链表操作时, 要判断是否是头指针。

### 7.3.2 栈

栈是只能在表的一端进行插入和删除运算的线性表。将允许插入和删除运算的一端称为栈顶 (top)，另一端称为栈底 (bottom)。将插入元素操作称为入栈，将删除元素操作称为出栈，栈称为“先进后出”表或“后进先出”表。



1) 栈特点: 后进先出。

2) 栈的基本运算

栈初始化: 构造一个空栈

空栈判断: 判断栈是否为空

入栈: 在栈顶插入一个元素

出栈: 在栈顶删除一个元素

读栈: 仅读取栈顶数据, 并不删除元素

3) 栈的顺序存储及其常用运算

用一块连续存储区域存放栈中元素, 连续区域的低地址一端作栈底, 栈底固定不变, 用 top 表示栈顶位置, n 表示栈中最多能容纳的元素个数。(栈顶位置为运算变动)

① 入栈运算: 在栈顶插入元素

S1: 如果  $\text{top} = n$ , 则栈已满, 提示入栈失败(栈“上溢”错误), 并结束入栈;

S2:  $\text{top} + 1 \Rightarrow \text{top}$ ;

S3: 将新元素放在当前栈顶位置(top)上。

② 出栈运算: 取出栈顶元素

S1: 如果  $\text{top} = 0$ , 则栈为空, 提示出栈失败(栈“下溢”错误), 并结束出栈;

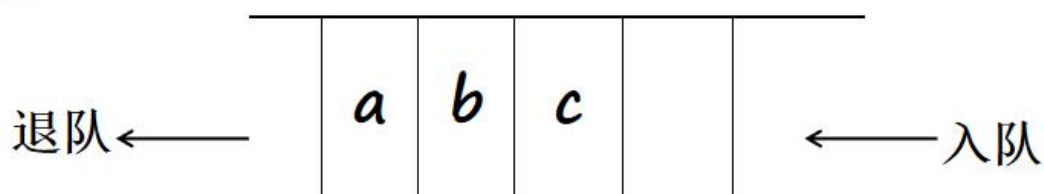
S2: 将当前栈顶(top)元素赋给一个变量;

S3:  $\text{top} - 1 \Rightarrow \text{top}$ 。

### 7.3.3 队 列

#### 1. 队列的定义

允许一端插入、而另一端删除的线性表, 允许插入的一端称为队尾, 允许删除的一端称为队头。



## 2. 队列的基本运算

初始化队列：创建一个空队列。

空队列判断：判断队列是否为空。

入队运算：在队尾插入一个元素。

出队运算：在队头删除一个元素。

读队头元素：读取队头元素赋给一个指定变量，不删除队头元素。

队列长度：求队列中元素个数。

## 3. 队列的顺序存储及其常用运算

通常将顺序存储的队列称为顺序队列。变量(front和rear)分别存放队列头和尾位置。front和rear都是整型变量。front指向队列中第一个元素的前一个单元位置，rear指向队列中最后一个元素位置。设队列中能容纳元素个数为 n。下面为队列的几种常用运算算法。(front 与 rear 位置为运算变化)

### 1) 初始化队列

创建一个空队列，并令  $\text{front} = \text{rear} = -1$

### 2) 入队运算

S1: 如果  $\text{rear} = n - 1$ ，则队列已满，提示入队失败(队列“上溢”错误)，并结束入队；

S2:  $\text{rear} + 1 \Rightarrow \text{rear}$ ;

S3: 将新元素放在当前队列位置(rear)上

### 3) 退队运算

此时虽然队列有空位置，但也不能插入新结点。

S1: 如果  $\text{front} = \text{rear}$ ，则队列已空，提示退队失败(队列“下溢”错误)，并结束退队；

S2:  $\text{front} + 1 \Rightarrow \text{front}$ ;

S3: 取 front 所指元素

## 7.4 树及二叉树

### 7.4.1 树

树是一种常用的非线性结构，树结构中结点之间既有分支关系又有层次关系。树是由  $n$  ( $n \geq 0$ ) 个结点组成的有限集合。

当  $n = 0$  时，称为空树；否则，有且仅有一个根结点。当  $n > 1$  时，其余结点被分成  $m$  ( $m > 0$ ) 个互不相交的子集  $T_1, T_2, \dots, T_m$ ，每个子集又是一棵树。

用图形表示树时，通常表示成一棵倒挂树。

在树的结构中有且只有一个根结点，根结点没有前件，其他结点只有一个前件，每个结点可以有多个(包括 0)后件，将结点的后件称为该结点的子结点，该结点是其子结点的双亲结点，将没有后件的结点称为叶子结点。一个结点所拥有后件个数称为该结点的度。树中所有结点的最大度称为树的度。树中结点的最大层次称为树的深度或高度。

结点的层次从根结点算起，根结点在第一层，根的直接后继结点在第二层，同一层上所有结点的后继结点均在下一层。

### 7.4.2 二叉树的特点及性质

#### ① 二叉树及其特点

二叉树是另一种树形结构，每个结点最多只有两个后件。其特点是：非空二叉树有且只有一个根结点；每个结点最多有两棵子树，且有左右之分。

二叉树有五种基本形态：a.空二叉树 b.只有根结点 c.只有左子树 d.有左右子树 e.只有右子树

#### ② 二叉树基本性质



性质一

在二叉树的第  $i$  层上, 最多有  $2^{i-1}$  个结点( $i \geq 1$ )

性质二

深度为  $k$  的二叉树最多有  $2^k - 1$  个结点( $k \geq 1$ )

性质三

对于任意一棵二叉树, 度为 0 的结点(即叶子结点)总比度为 2 的结点多 1 个

③ 满二叉树

每一层的结点数都达到最大值, 叶子结点都在最下面的同一层上, 拥有  $2^k - 1$  个结点, 深度为  $k$  的二叉树。

④ 完全二叉树及其性质 (性质删了, 背着麻烦)

一棵深度为  $k$  的二叉树, 如果第一层到第  $k-1$  层是一棵满二叉树, 第  $k$  层上的结点数可能没达到最大值  $2^k - 1$ , 但这些结点都满放在该层最左边。

注: 满二叉树是完全二叉树, 但完全二叉树不一定是满二叉树。

### 7.4.3 二叉树的存储

① 二叉树的顺序存储

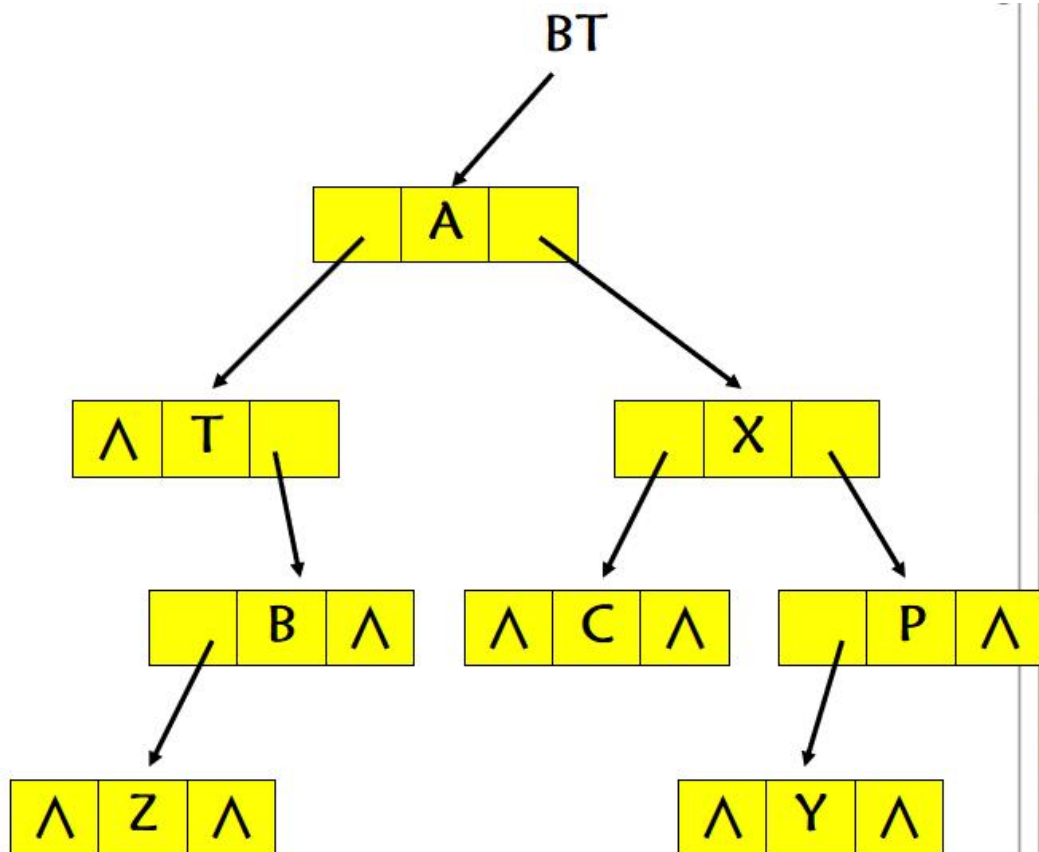
指用一组连续存储单元存储二叉树中的结点。结点存储顺序是按着从上到下、从左到右顺序。按完全二叉树每个结点编号的顺序存放结点, 通过结点的编号准确地反映结点之间的逻辑关系。

顺序存储结构适用于完全二叉树, 对于非完全二叉树, 浪费许多存储空间。

② 二叉树链式存储

二叉树每个结点由数据域和指针域组成。两个指针域一个用于指向左子结点, 一个用于指向右子结点。





#### 7.4.4 二叉树遍历

遍历二叉树就是按照某种顺序访问二叉树中每个结点的过程,每个结点被访问一次且仅一次。非空二叉树可以看成由根结点、左子树和右子树三部分构成。

二叉树的遍历分为先序遍历、中序遍历和后序遍历。

##### ① 先序遍历

- ✓访问根结点
- ✓先序遍历左子树
- ✓先序遍历右子树

##### ② 中序遍历

- ✓中序遍历左子树
- ✓访问根结点
- ✓中序遍历右子树

##### ③ 后序遍历

- ✓后序遍历左子树
- ✓后序遍历右子树
- ✓访问根结点

#### 7.5 数值计算方法及程序设计

数值计算方法是一种研究并解决数学问题的数值近似求解的方法,是用计算机解决数学

问题的方法。

迭代算法和递归算法是典型的数值算法。

#### 7.5.1 迭代算法

迭代算法是一种不断用变量的旧值递推新值的过程，一般从一个初始估计出发寻找一系列近似解来解决问题。

迭代过程的控制通常可分为两种情况：

一种是所需的迭代次数是个确定的值，可以构建一个固定次数的循环来实现对迭代过程的控制，如求  $1+2+3+\cdots+100$  值；

一种是所需的迭代次数无法确定，需要进一步分析出用来结束迭代过程的条件，如求方程的根。

#### 7.5.2 递归算法

一个过程或函数在其定义或说明中又直接或间接调用自身的一种方法，称为递归算法。

一般来说，递归需要有边界条件、递归前进段和递归返回段。当边界条件不满足时，递归前进；当边界条件满足时，递归返回

### 7.6 数据排序算法及程序设计

排序是将一组无序数据按值递增（或递减）进行重新排列。待排序数据序列可以是顺序存储或链式存储结构。下面排序算法中，待排序数据序列均采用顺序存储结构。

交换排序、选择排序和插入排序是 3 类基本排序方法。

#### ● 交换排序法

在排序过程中，通过数据元素之间不断地进行比较与交换，最终达到排序目的。冒泡排序法是典型的交换排序法之一。

冒泡排序法的基本思想是：对所有相邻元素进行比较，若逆顺，则将其交换，最终达到有序化。

### 7.7 数据查找算法及程序设计

查找又称检索，是数据处理中使用最频繁的一种操作。查找是指在数据集合中查找某个数据元素的过程。若存在这样数据元素，则查找成功；否则，查找失败。

#### 7.7.1 顺序查找

顺序查找适用于线性表。其基本方法是：

从线性表中第一个元素开始，依次将线性表中元素与给定值进行比较。若相等，则查找成功；若直到最后一个元素，还没找到与给定值相等的元素，则查找失败。

特点：顺序查找算法简单，但是执行效率较低

在下列两种情况下，只能使用顺序查找法：

- ①线性表链式存储。
- ②线性表顺序存储，但元素无序排列。

#### 7.7.2 二分查找法

又称折半查找，要求被查找的表用顺序存储结构且数据元素按数据值升序或降序排列，即二分查找法只适用于有序表。

基本思想（设顺序表升序排列）

将给定值与中间元素比较，若相等，则找成功；若给定值小于中间元素值，则继续对前半再折半查找；若给定值大于中间元素值，则继续对后半再折半查找。

## 8.1 实例数据库

### 8.1.1 人工表、数据库表及转换

- 1) **人工表**：对事物进行抽象、提取、归纳和总结，通过表格描述客观事物(对象)特征。
- 2) **数据库表**：结构化、有数据类型、存于数据库中的二维表。
- 3) **人工表转为数据库表**：拆分某些数据项，使其意义更明确；为数据项起容易记、便于操作且有意义的名称；对某些数据项进行必要的编码。转换过程。

## 8.2 数据库系统概述

### 8.2.2 数据库系统组成

数据库系统：**DBS**—Data Base System，是存储数据库的计算机系统。

系统组成：**硬件**、**软件**和**相关人员**。

硬 件：提供系统运行和存储数据库**环境**。

软 件：用于管理、控制和分配计算机资源，建立、维护和使用数据库 。

相关人员：实施管理和使用数据库的人。

1. **数据**—Data：描述事物、存储于计算机系统中的符号。可以是数字、文字、图形、图像、声频、视频等。

2. **数据库 DB**—Data Base：动态存储在计算机系统中、有组织、结构化的关联数据集合。以文件方式存储。常见文件扩展名：**MDF**—SQL Server；**MDB**—Access；**DBC**—VFP(Visual FoxPro)

3. **数据库管理系统 DBMS**—Data Base Management System：建立、维护和管理数据库的**软件**。

### 8.2.3 数据库管理系统功能

- 1) **数据定义**：通过数据定义语言 **DDL**， 定义数据库、表、数据一致性、安全性、完整性规则和索引。
- 2) **数据操纵**：通过数据操纵语言 **DML**，数据插入、修改和删除。
- 3) **数据查询**：通过数据查询语言 **DQL**，对数据查询、排序、汇总和表连接等操作。

## 8.4 数据模型

数据模型：是描述实体及其联系的方法，是信息世界中概念模型的数据化，是数据库的逻辑结构和设计基础。

对数据模型的要求

- 1) 直观、易懂，能很容易地被人们使用和理解；
- 2) 具有很强仿真性，能比较逼真地反映现实世界中事物及其联系；
- 3) 在计算机中容易实现存储和操作。

常见**模型**：层次数据模型、网状数据模型、关系数据模型和面向对象数据模型。

### 8.4.3 关系数据模型

二维表描述实体型，表中记录表示实体。同类实体间的联系由同表同字段实现；不同实体型之间联系由相同含义字段实现。

## 8.5 关系数据库基本概念

- 1) 关系：二维表，是实体属性信息（名、类型和宽度等）及属性值（记录）的集合。属性个数（n）称为关系的元或目。

- 2) 属性：表中每一列称为一个属性，也称之为列、字段或数据项。有属性名，也称之为列名或字段名。
- 3) 元组：表中一行数据称为一个元组，也称之为记录，一个元组对应一个实体。
- 4) 关键字：一或多个属性组成，唯一标识元组的最小属性集合。表都有关键字，将关键字也称为候选键或候选码。
- 5) 主关键字：将用户选用的候选码称为主关键字，也简称主键、主码。每表一个。
- 6) 主属性：含在候选码中的属性。
- 7) 外码：表 R 的一组属性 F 不是表 R 的候选码，如果 F 与某表 S 的主键相对应（属性的含义相同），则 F 是表 R 的外码或外键。外码通过其他表的主码建立表间联系。
- 8) 关系模式：是对关系的描述，关系名及其全部属性信息的集合，关系模式描述表结构。  
格式：<关系名>(全部属性)，如 教师 ( 教师号，姓名，性别，出生日期，职称，办公电话，住宅电话，移动电话，在职 )
- 9) 关系子模式：对用户操作数据的结构描述。用户需要的数据可来自于一张表或多张表。

## 8.6 数据模型的要素

数据模型构成：数据结构、数据操作和数据完整性约束 3 个要素。

关系数据模型主要用二维表及其联系描述。

### ① 数据结构

主要描述数据类型、内容以及数据间的联系等，即数据存储的静态性。表是关系数据库的核心内容，主要用于存储关系(表)结构和数据记录。

表主要具有下列性质：

- 1) 属性原子性：属性不可再分割。
- 2) 属性名惟一性：同一表中属性不可重名。
- 3) 属性有限性：属性个数有限，至少 1 个。
- 4) 属性无序性：属性前后顺序无关紧要。
- 5) 属性值域同一性：同一列数据必须具有相同的数据类型和取值范围。
- 1) 元组次序无关性：元组的前后顺序无关。
- 2) 元组唯一性：任意两个元组不能完全相同，任何表都有关键字。
- 3) 元组有限性：表中元组(记录)个数有限，可无记录。
- 4) 关键字非空性：主属性对任何元组都不能出现空值。

### ② 数据操作

对数据模型中数据和联系所允许的操作及操作规则，即描述数据模型的动态性。

- 1) 查询数据：检索数据库中数据，同时可以检索一个表或多个表中数据。
- 2) 插入数据：向表中增加记录，一次操作仅向一个表中插入记录。
- 3) 删除数据：先在表中查找（选择）记录，然后再从表中将其删除。一次操作仅从一个表中删除记录。
- 4) 修改数据：修改表中相关记录属性值，一次操作仅修改一个表中数据。可视为查找、删除和插入数据 3 种操作的组合。

对数据库的上述 4 种操作可归纳成记录选择、数据项投影、表连接、记录插入和删除 5 种基本操作。总之，对数据模型的所有操作都是由这 5 种基本操作及其组合完成的。在关系数据库管理系统中，通过专用操作实现记录选择、数据项投影和表连接；通过集合运算实现记录插入和删除。

### ③ 完整性约束

数据语义是对数据含义的规定与解释。语义可确定属性的取值范围(值域)；确定表的关键字；限制更新相关表中数据。

完整性约束是语义施加在数据上的限制。关系数据模型中有域、实体、参照和用户定义4类完整性约束。

- 1) 域完整性约束：限制属性值的数据类型和范围。数据类型和宽度由D B M S实现，而数据范围由用户定义。
- 2) 实体完整性：每个表都必须有关键字，主属性值不能为空，以确保主键能唯一地标识记录。
- 3) 参照完整性：对表之间联系约束，不允许一个表引用另一个表中不存在的记录。
- 4) 用户定义完整性：用户根据具体数据库设置的约束条件。即字段有效性规则。

## 8.7 关系的基本操作

关系(表)的专门操作：数据项投影、记录选择和表连接3种。操作结果仍然是关系(表)。

- 1) 选择操作：从关系中选取符合条件的元组(记录)，指明被操作的元组条件。

比较运算： $<$ 、 $<=(\leq)$ 、 $=$ 、 $<>(\neq)$ 、 $>=(\geq)$ 或 $>$ ；

逻辑运算：Or(或)、And(并且)、Not(否定)。

- 2) 投影操作：从关系中选取若干个属性(列)，用于指定要操作的属性名称。

- 3) 连接操作：对2张表进行连接，根据某种条件生成一张新表。

等值连接：比较运算符都是“=”的连接。

自然连接：不含冗余属性的等值连接。

## 8.8 SQL 语言简介

SQL—Structured Query Language，是结构化查询语言，数据库通用操作语言。

语言构成：数据定义、数据操纵、数据查询和数据控制语言4部分组成。

### 8.8.2 数据定义语言 DDL

#### ① 建立数据库表：

Create Table <表名>

( <字段名 1> <类型描述> [ [Not] Null ] [Primary Key ]

...

[, <字段名 n> <类型描述> [ [Not] Null ] [Primary Key ]

[, Primary Key (<字段名表>)]

...

);

#### ② 修改表结构：

增加字段：

Alter Table <表名> Add <字段名> <类型描述>

修改字段：

Alter Table <表名> Alter <字段名> <类型描述>

删除字段：

Drop <字段名>

#### ③ 删除表：

Drop Table <表名>



### 8.8.3 数据操纵语言 | DML

#### ① SQL 中常数的表示

- 1) 数值型：包括双精度、长整型和整型。
- 2) 文本型：是一段文字，也称字符串。用半角单引号‘或双引号’括起数据。
- 3) 逻辑型：也称是/否型，True(-1)表示真，False(0)表示假。
- 4) 日期型：常数格式为：#月/日/年#或("日期串")，年份可 2 或 4 位，可用 DateValue 转换。

#### ② 增加数据记录

Insert Into <表名> [(<字段名表>)] Values (<表达式表>);

如：Insert Into 课程 (课程号, 课程名, 学分) Values (' 01004', ' 专业英语', 4);

Insert Into 课程 Values (' 01005', ' 英语写作', 4, 80);

#### ③ 修改数据记录

Update <表名> Set <字段名 1> =<表达式 1>

[..., <字段名 1> =<表达式 1>] [Where <条件>];

条件：是逻辑值表达式，可以含比较、逻辑或谓词运算。

##### 1) 区间谓词：

<表达式 1> Between <表达式 2> And <表达式 3>

如：Update 课程 Set 实验学时=10 Where 实验学时=0 And 学分 Between 3 And 5;

##### 2) 属于谓词：<表达式 1> In (<表达式表>)

如：Update 课程 Set 实验学时=10 Where 实验学时=0 And 学分 In (3,4,5);

##### 3) 模糊谓词：<字符式 1> Like <字符式 2>

如：Update 课程 Set 理论学时=72 Where 理论学时<72 And 课程名 Like '\*英语\*'

#### ④ 删除记录

Delete From <表名> [Where <条件>];

如：Delete From 教师 Where 出生日期<=#1955/12/31# ;

### 8.8.4 数据查询语言 | DQL

Select [Distinct][<表名>.\*] |

<表达式 1>[ As <别名 1> ][, ...]

From <表名 1>[, ...]

[Where <条件>]

[Group By <分组字段> [Having <条件>] ]

[Order By <排序关键字> [ASC | DESC]];

## 第九章多媒体技术基础

### 9.2 多媒体技术概论

**媒体**是表示和传播信息的方法。**多媒体**是多种**信息载体**的**表示形式**和**传递方式**，是以计算机技术为基础多种媒体数字信息与相关设备交互处理的**手段**和**方法**。

### 9.3 数据压缩方法

**数据压缩**：指通过某种算法**减少冗余**(重复)或**相关联数据**的**加工**或**运算过程**。

**数据解压缩**：是由压缩数据**恢复**到原数据的**过程**，也成**还原**数据。

数据**压缩/解压缩工具**：分**硬件**(如**声卡**、**视频卡**等)和**软件**两种。

**信源符号**：将数字化的任何(字节)信息(文字、音频、图像和视频)都视为符号，**每种符号**称**信源符号**。

压缩方法分类：根据压缩的数据能否准确还原，可分**无损**和**有损压缩**。

### 9.3.1 无损压缩

无损压缩的压缩 / 解压缩过程可逆，即压缩后的数据可还原，没有丢失。适于不允许丢的数据，如文本和程序等。WinZip、WinRAR 是无损压缩 / 解压缩软件。

压缩比：与压缩算法、被压缩的数据有关。

压缩的主要思想：统计原数据中的信源符号出现的次数，重新编码。

典型的无损压缩方法有哈夫曼编码和行程编码。

#### 1. 行程编码

将原数据中连续出现的信源符号用一个计数值(称行程长度)和该信源符号来代替。

如：BBBBBCCCCAAAAAGGG，占 17 个字节，压缩后的数据为：5B3C6A3G，数字表示其后符号连续出现的次数，占 8 个字节，压缩比例约 2:1。

主要特点：方法简单直观，易压缩/解压缩，速度快。压缩比与原数据有关，行程越长，压缩比越大。适合相邻重复量大的数据。

#### 2. 哈夫曼编码

按信源符号出现的比率进行编码；在解压缩时能准确地还原，使平均编码字长尽量短。  
步骤：

- 1) 将信源符号按出现的比率排序。
- 2) 取 2 个最小比率相加得新比率，将比率大的路径赋为 0，比率小的路径赋为 1。
- 3) 重复第 2)步，直到合并了所有的信源符号，形成合并过程的二叉树。
- 4) 树根到信源符号路径上的 1、0 序列即为符号的编码。

### 9.3.2 有损压缩

优化一些冗余或相关联的数据，有丢失。压缩过程不可逆，即不能还原。适用于图像、音频和视频等信息量大的多媒体数据。

压缩思路：根据信息的物理特性、人感知器官(耳和眼)的误差和专业知识，整理、优化和压缩人们易忽略的细节，换取较大的压缩比例。

应用环境：软件固化在多媒体卡(如声、视频卡)中，数字化时压缩，播放时解压缩。

## 9.4 音频技术

音频(Audio)：包括语音、音乐及各种动物和自然界发出的声音。

音频技术：是利用计算机及音频设备处理音频信息的技术，主要包括录制、数字化、存储和播放等。

### 9.4.2 音频信号的数字化

音频信号：是时间和幅值连续的信号，有无穷多个点，即模拟信号。

数字信息：时间和幅值离散，有限个点。

数字化：将模拟信号转成数字信息，包括音频采样、量化和编码。

#### 1. 采样

每隔一定间隔取一个点，将连续的信号曲线(无穷多个点)变成离散的有限个信号点。

均匀采样：间隔相等的采样。

采样频率：每秒采样的点数。采样频率越高，数字化音频质量越好，但数据量越大。

奈奎斯特理论：若采样频率不低于模拟音频最高频率的 2 倍，则能还原音频。

#### 2. 量化

以数值表示每个采样点的幅度。每个点对应一个数，量化值的范围由振幅决定。

### 3. 编码

对量化值进行整理,用有限位的二进制整数表示音频。

编码字长: 编码所用二进制位数,也称采样解析度。音质与采样频率、编码字长有关。

编码与量化值的关系式:  $\text{编码 } i = \text{编码最大值} \div \text{量化最大值} \times \text{量化值 } i$ , 取四舍五入整数。

### 4. 数字音频的数据量

数据率: 数字化 1 秒(s)音频所需要的数据位数(bit)。数据率(b/s)=

采样频率(Hz)×编码字长(b)×声道数。

### 9.5.1 图像的特点

图像(Image): 是输入设备(扫描仪和数码照相机等)捕捉的静态画面,以位图(像素点阵)形式存储,描述每个像素点的色彩。

#### 1. 图像分辨率

是图像上的像素总数,用水平像素数×垂直像素数表示。分辨率越高,图像细节表现力越强,清晰度越高。

#### 2. 图像颜色模型

1) RGB 模型: R、G 和 B 分别表示红(Red)、绿(Green)和蓝(Blue)色。各颜色强度分 256 级(0~255),组合成  $256^3=16777216$  种颜色。

2) HSB(L)模型: 由色调、饱和度和亮度 3 个要素定义颜色。色调表示颜色色彩; 饱和度表示色彩的强度。

在 Word 中打开颜色对话框的方法: 单击字体颜色→其他颜色→自定义选项卡。

3) CMYK 模型: 设置打印颜色,由青(C)、品红(M)、黄(Y)和黑(K)4 种颜色混合。

一般用 RGB 模型编辑信息, 打印时系统自动转换为 CMYK 模型。

在 Word 中设置 CMYK 模型的方法:

单击文件→打印,单击打印对话框中属性按钮;选定属性对话框中颜色,单击高级按钮;单击高级选项对话框中属性按钮;在颜色调整对话框中设置颜色平衡(CMYK)。

### 9.5.2 图像信息的数字化

1. 采样: 从连续图像中取若干个点,转换成离散点图像。将采样像素点总数称采样分辨率。

2. 量化: 是用数值表示图像中每个采样点色彩的过程。

3. 编码: 用有限位二进制数表示颜色数据。

颜色深度: 指编码所用二进制位数,也称编码字长。一般为 8、16、24 或 32。如: 颜色深度为 16,表示  $2^{16}=65536$  种颜色,1 个点占 2 个字节。

4. 图像的数据量:  $(\text{水平像素数} \times \text{垂直像素数} \times \text{颜色深度}) / 8B$ 。

### 9.5.3 图形技术

图形(Graphics): 是软件绘制的画面,由点、线、矩形和椭圆等组合而成,也称矢量图。

图形文件内容: 是描述像素点位置及色彩的程序。执行程序输出图形,由指令生成像素点及颜色。

图形的优点: 存储空间小;变形(如缩放、旋转、扭曲)时误差小。矢量图可转成位图,但反之困难。

## 9.6 视频与动画技术

视频(Video): 由一系列相关静态图像组成,每幅图像称一帧。12 帧/秒(12fps)以上速率播放,产生运动效果。典型帧速为 24~30fps,效果既连续,又平滑。

动画(Animation): 由一系列相关图形或图像组成,可通过软件生成,播放时有动感效果。

### 9.6.1 视频信息的特点

1. **模拟**视频：**传统影视和录像**等以**模拟方式**记录、存储和传播，存储介质是胶带和录像带。

**NTSC**制式：**30fps, 525**行/帧，美国、加拿大和日本等采用。

**PAL**制式：**25fps, 625**行/帧，中国及欧洲多数国家采用。

2. **数字**视频：由**数码摄像机、监控设备**等录制，以文件形式存储。优点是方便创作和加工处理(加字幕、旁白)、节省制作成本、无损耗复制、长期保存无衰减、可倒序播放和易传播等。

### 9.6.2 视频信息的表示

1. 视频数字化：视频是音频和图像的结合，数字化可用**图像**和**音频数字化**方法**结合实现**。

**录像带**或**激光视盘**数字化：安装**视频卡**，连接录放像机、摄像机、电视机或影碟机，安装**驱动程序**和视频**处理软件**。

2. 数字视频的**数据量**：数据率(数据量/秒)=**帧速×每幅图像数据量+音频数据量**。

### 9.6.3 动画技术

1. 帧动画：构成动画的**基本单位**是**静态图形或图像(帧)**，**相关联帧**连续播放，形成**动画**。

2. 矢量动画：经过计算机**软件生成**的动画，主要表现图形、曲线、文字或图案的变换。

## 需要理解的知识

### (一) 进位计数制

2B 80 10D 16H

(4) R 进制转 10 进制：按权展开式

$$(307.16)_o = 3 \times 8^2 + 0 \times 8^1 + 7 \times 8^0 + 1 \times 8^{-1} + 6 \times 8^{-2} \quad (\text{起始为 } R^{n-1})$$

3、0、7 ( $a_i$ ) 为数码

8 (R) 为基数，可取最大为 R-1

$8^2$  ( $R^i$ ) 为位权

(5) 十进制转 R 进制

整数除 R 取余，由下到上

小数乘 R 取整，由上到下，计算到 n+1 位， $<R/2$  舍去， $\geq R/2$  进位

(6) 二、八、十六进制数间的相互转换

一位八进制数对 3 位二进制数

一位十六进制对 4 位二进制数

### (二)

8421 编码

涂写的选项之和表示要输入的本列数字，涂写[2]和[4]表示 6

涂写的选项之和大于 10 时无效，等于 10 时表示 0。

### (三)

(1)

计算机内部信息只有 0 和 1 两种形式。规定最高位为符号位，用 0 表示“+”号；用 1 表示“-”号，称为数符，其余位表示数值。

将机器内存储的带符号数称为机器数，而由正、负号加绝对值表示的实际数称为真值数。

$$(-193)_D = (-11000001)_B$$

机器内用 16 位时，其机械数为 10000000111000001

标绿符号数/数符 标红数值

(2)

机器内部小数点是隐含的，位置可以固定，也可以可变动。前者称为定点数，后者称为浮点数。

定点数

1 定点整数

小数点位置固定在数值最低位后面，表示整数

2 定点小数

小数点位置定在数值最高位前面，表示纯小数

当用 M 位二进制数存储数据时

定点整数 N 的取值范围是

$$-(2^{M-1}-1) \leq N \leq (2^{M-1}-1)$$

定点小数 N 的取值范围是

$$-(1-2^{-(M-1)}) \leq N \leq (1-2^{-(M-1)})$$

无符号整数

省略符号位的正整数被称为无符号整数。

在计算机中存储无符号整数时，不留符号位，所有数位都用于存储数值。

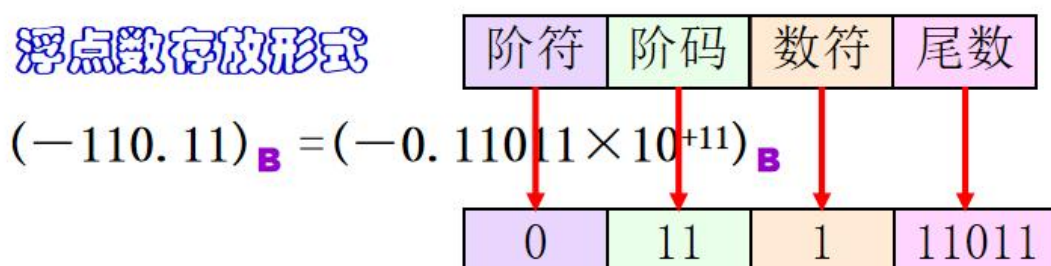
浮点数

任何R进制数都可以表示成： $N = \pm S \times R^{\pm J}$

数符      尾数      阶符      阶码

二进制规范化形式规定：尾数为定点小数且尾数值的最高位为 1

浮点数存放形式



(四) 逻辑运算

1 真 0 假

与运算 “×”、“∧” 或 “and”

或运算 “+”、“∨” 或 “or”

非运算 “ $\overline{A}$ ” 或 “not”

(五) 常数

- 1) 整型：由数字 0 ~ 9 和正(+)负(-)号组成。如：100、-5 。
- 2) 长整型：数末尾加&。如：100& 。
- 3) 单精度型：数末尾加! 和 E 指数形式的数。如：10!、-5.7!、3.5E-2。
- 4) 双精度型：数末尾加#、D 指数形式或直接写带小数点的数。如：3.14、7#、-5D3 。
- 5) 字符串型：用半角双引号引起字母、数字、汉字及标点符号等。字符串中有双引号时，用连续两个双引号。如："计算机"、"他说：""明天来"" "
- 6) 日期型：用#括起日期和时间数据。如： #2012-10-1 #。
- 7) 逻辑型：只有 True 和 False 两个值。

(六) 变量命名

- 1) 必须以字母或汉字开头，由英文字母、汉字、数字或下划线组成，字母不区分大小写，最大长度为 255 个字符。如：N1、学号
- 2) 不能用 VB 的保留字(系统已用的单词，如：String、Caption、True 等)作变量名。

(七)

① 简单分支结构

1) 简单分支语句



语句格式 1

If <表达式> Then <语句序列 1> [ Else <语句序列 2> ]

语句格式 2

If <表达式> Then

<语句序列 1>

[Else

<语句序列 2> ]

End If

2) 分支函数

IIf(<表达式>, <表达式 1>, <表达式 2>)

3) MsgBox 对话框函数

Msgbox(<字符式 1>[,<数值式>[,<字符式 2>]])

## ② 多分支结构

多分支 If 语句格式

If <表达式 1> Then

<语句序列 1>

Elseif <表达式 2> Then

<语句序列 2>

.....

Elseif <表达式 n> Then

<语句序列 n>

[ Else

<语句序列 n+1> ]

End If

## ③ 循环结构

1. For 循环结构

For <循环变量>=<初值> To <终值> [Step <步长>]

<循环体代码>

Next [ <循环变量>]

2. Do While 循环

Do While <循环条件>

<循环体代码>

Loop

3. Do 循环结构

Do

<循环体代码>

Loop Until <表达式>

## (八) 网络分割

### ① IP 地址

IP 地址包括两部分内容：一部分为网络标识，另一部分为主机标识。主机标识不可全 0 或全 1。分为 A~E 类：

A 类：前 8 位为网络地址，第一组第一位为 0

B 类：前 16 位为网络地址，第一组前两位为 10

C 类：前 24 位（前三组）为网络地址，第一组前三位为 110

## ②子网掩码

子网掩码用 32 位二进制数编码，表示形式与 IP 地址类似。子网掩码左边连续“1”的个数与网络标识的二进制位数一致。

### a. 获取网络 IP 地址

通过主机 IP 地址和子网掩码可以获取主机所在网络 IP 地址和主机标识。

具体做法：将主机 IP 地址与子网掩码按位进行“与”运算，即可得到网络 IP 地址。

IP 地址： 10011111 11100000 00000111 10000001

159.224.7.129（B 类地址）

子网掩码： 11111111 11111111 00000000 00000000

255.255.0.0

网络地址： 10011111 11100000 00000000 00000000

159.224.0.0

这是 B 类 IP 地址和子网掩码，主机所在网络地址为 159.224，主机地址为 7.129。又如 C 类 IP 地址，的子网掩码是 255.255.255.0。

### b. 分割网络

例

某单位申请到一个 C 类网络 IP 地址：212.188.199，要在相距较远位置分别建立 4 个子网，每个子网中不足 60 台主机。

设计方法：把主机标识的前 2 位作为子网 IP 地址的一部分，后 6 位作为新的主机标识。作为子网 IP 地址的 2 位可组合成：

00000000、01000000、10000000 和 11000000

即将子网掩码设计为：255.255.255.192

得到：212.188.199

212.188.199.64

212.188.199.128

212.188.199.192

4 个子网 IP 地址。

4 个子网 IP 地址，每个子网中的主机标识范围为：000001 至 111110， $(2^6 - 2) = 62$  个

（去除全 0 或全 1）可供分配的主机 IP 地址。其中：

子网 212.188.199 中主机 IP 地址范围：212.188.199.1~62

子网 212.188.199.64 中主机 IP 地址范围：212.188.199.65~126

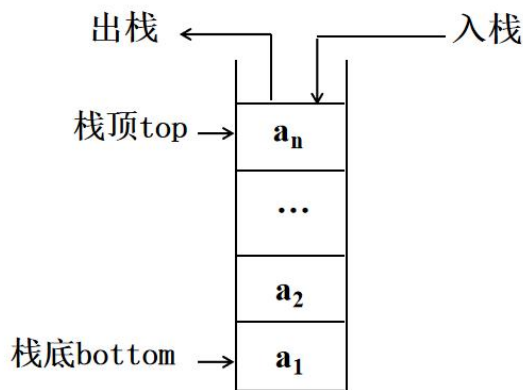
子网 212.188.199.128 中主机 IP 地址范围：212.188.199.129~190

子网 212.188.199.192 中主机 IP 地址范围：212.188.199.193~254

## （九）出栈入栈、出队入队

栈是只能在表的一端进行插入和删除运算的线性表。将允许插入和删除运算的一端称为栈顶（top），另一端称为栈底（bottom）。将插入元素操作称为入栈，将删除元素操作称为

出栈，栈称为“先进后出”表或“后进先出”表。



4) 栈特点：后进先出。

5) 栈的基本运算

栈初始化：构造一个空栈

空栈判断：判断栈是否为空

入栈：在栈顶插入一个元素

出栈：在栈顶删除一个元素

读栈：仅读取栈顶数据，并不删除元素

6) 栈的顺序存储及其常用运算

用一块连续存储区域存放栈中元素，连续区域的低地址一端作栈底，栈底固定不变，用  $top$  表示栈顶位置， $n$  表示栈中最多能容纳的元素个数。（栈顶位置为运算变动）

③ 入栈运算：在栈顶插入元素

S1: 如果  $top=n$ ，则栈已满，提示入栈失败(栈“上溢”错误)，并结束入栈；

S2:  $top + 1 \Rightarrow top$ ;

S3: 将新元素放在当前栈顶位置( $top$ )上。

④ 出栈运算：取出栈顶元素

S1: 如果  $top=0$ ，则栈为空，提示出栈失败(栈“下溢”错误)，并结束出栈；

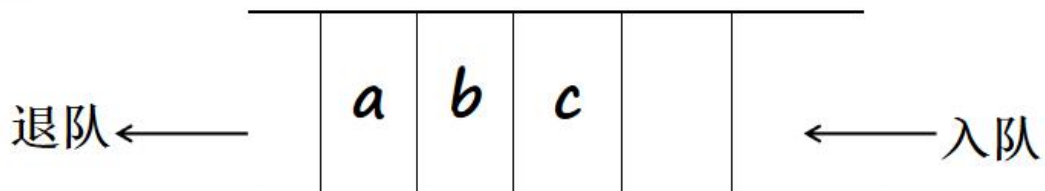
S2: 将当前栈顶( $top$ )元素赋给一个变量；

S3:  $top - 1 \Rightarrow top$ 。

### 7.3.3 队 列

#### 1. 队列的定义

允许一端插入、而另一端删除的线性表，允许插入的一端称为队尾，允许删除的一端称为队头。



#### 2. 队列的基本运算

初始化队列：创建一个空队列。

空队列判断：判断队列是否为空。

入队运算：在队尾插入一个元素。

出队运算：在队头删除一个元素。

读队头元素：读取队头元素赋给一个指定变量，不删除队头元素。

队列长度：求队列中元素个数。

#### 4. 队列的顺序存储及其常用运算

通常将顺序存储的队列称为顺序队列。变量(front和rear)分别存放队列头和尾位置。front和rear都是整型变量。front指向队列中第一个元素的前一个单元位置，rear指向队列中最后一个元素位置。设队列中能容纳元素个数为n。下面为队列的几种常用运算算法。(front与rear位置为运算变化)

##### 4) 初始化队列

创建一个空队列，并令 front= rear= -1

##### 5) 入队运算

S1: 如果 rear=n-1, 则队列已满，提示入队失败(队列“上溢”错误)，并结束入队；

S2: rear+1  $\Rightarrow$  rear;

S3: 将新元素放在当前队列位置(rear)上

##### 6) 退队运算

此时虽然队列有空位置，但也不能插入新结点。

S1: 如果 front=rear, 则队列已空，提示退队失败(队列“下溢”错误)，并结束退队；

S2: front+1  $\Rightarrow$  front;

S3: 取 front 所指元素

#### (十) 二叉树遍历

遍历二叉树就是按照某种顺序访问二叉树中每个结点的过程，每个结点被访问一次且仅一次。非空二叉树可以看成由根结点、左子树和右子树三部分构成。

二叉树的遍历分为先序遍历、中序遍历和后序遍历。

##### ④ 先序遍历

√访问根结点

√先序遍历左子树

√先序遍历右子树

##### ⑤ 中序遍历

√中序遍历左子树

√访问根结点

√中序遍历右子树

##### ⑥ 后序遍历

√后序遍历左子树

√后序遍历右子树

√访问根结点

#### (十一) 排序算法

##### 交换排序法

在排序过程中，通过数据元素之间不断地进行比较与交换，最终达到排序目的。冒泡排序法是典型的交换排序法之一。

冒泡排序法的基本思想是：对所有相邻元素进行比较，若逆顺，则将其交换，最终达到有序化。

## （十二）查找算法

**查找**是指在数据集合中**查找**某个数据元素的过程。若**存在**这样数据元素，则查找**成功**；否则，**查找失败**。

### 7.7.1 顺序查找

从线性表中**第一个**元素开始，依次将线性表中元素与**给定值**进行**比较**。若**相等**，则查找**成功**；若直到**最后**一个元素，还没找到与给定值相等的元素，则查找**失败**。

### 7.7.2 二分查找法

将给定值与中间元素比较，若相等，则找成功；若给定值小于中间元素值，则继续对前半再折半查找；若给定值大于中间元素值，则继续对后半再折半查找。

## （十三）SQL 中常数的表示

- 1) 数值型：包括双精度、长整型和整型。
- 2) 文本型：是一段文字，也称字符串。用半角单引号 ‘或双引号’括起数据。
- 3) 逻辑型：也称是/否型，True(-1)表示真，False(0)表示假。
- 4) 日期型：常数格式为：#月/日/年#或("日期串")，年份可 2 或 4 位，可用 DateValue 转换。

## （十三）哈夫曼编码

按信源符号出现的**比率**进行**编码**；在解压缩时能准确地还原，使**平均编码**字长尽量**短**。  
步骤：

- 1) 将信源符号按出现的比率排序。
- 2) 取 2 个最小比率相加得新比率，将比率大的路径赋为 0，比率小的路径赋为 1。
- 3) 重复第 2)步，直到合并了所有的信源符号，形成合并过程的二叉树。
- 4) 树根到信源符号路径上的 1、0 序列即为符号的编码。