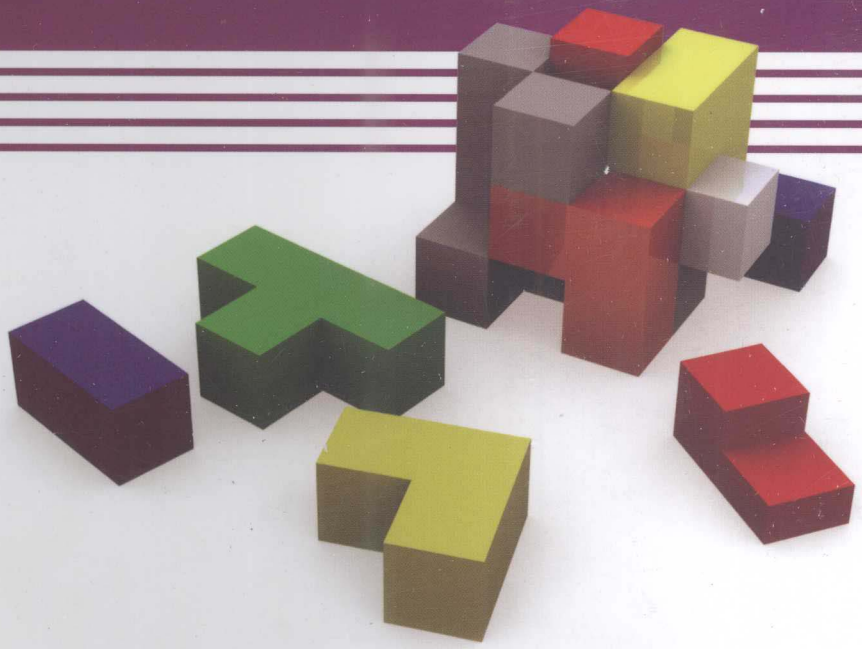
高等学校数据结构课程系列教材

数据结构教程
（C++语言描述）

李春葆　主编



清华大学出版社

高等学校数据结构课程系列教材

数据结构教程(C++语言描述)

李春葆　主编

陈良臣　尹为民　蒋晶珏　喻丹丹　编著

清华大学出版社

北京

内 容 简 介

本书系统地介绍了各种常用的数据结构以及排序、查找的各种算法，阐述了各种数据结构的逻辑关系、存储表示及运算操作，并采用C++语言描述数据组织和算法实现。

全书既注重原理又注重实践，配有大量的图表、示例和习题，内容丰富，概念讲解清楚，表达严谨，逻辑性强，语言精练，可读性好。

本书既便于教师课堂讲授，又便于自学者阅读，既可作为高等院校计算机相关专业本科生、专科生的教材，也可供广大从事计算机应用的科技人员参考。

图书在版编目(CIP)数据

数据结构教程：C++语言描述/李春葆主编. --北京：清华大学出版社，2014
高等学校数据结构课程系列教材
ISBN 978-7-302-35121-4

Ⅰ. ①数… Ⅱ. ①李… Ⅲ. ①数据结构－高等学校－教材 ②C语言－程序设计－高等学校－教材 Ⅳ. ①TP311.12 ②TP312

中国版本图书馆CIP数据核字(2014)第012452号

责任编辑：魏江江　王冰飞
封面设计：杨　兮
责任校对：时翠兰
责任印制：何　芊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62795954

印 刷 者：三河市君旺印务有限公司
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：26.75　　**字　　数**：651千字
版　　次：2014年8月第1版　　**印　　次**：2014年8月第1次印刷
印　　数：1～2000
定　　价：49.00元

产品编号：057038-01

FOREWORD

前言

数据结构课程总结和归纳了软件开发中常用的数据结构，并从数据逻辑结构、存储结构和基本运算算法设计3个层面来讨论问题的求解方法，以提高学生基本的数据组织和处理能力，为后续课程(如操作系统、数据库原理和编译原理等课程)的学习打下基础。

本书是编者根据多年的教学经验，参考近年来国内外出版的多种数据结构以及C++面向对象程序设计教材，考虑教与学的特点，合理地进行内容的取舍，精心组织编写而成的。本书采用C++语言面向对象方法描述数据结构和算法。全书由11章构成，各章内容如下：

第1章绪论，介绍数据结构的基本概念、采用C++语言描述算法的方法和特点、算法分析方法,以及怎样设计好算法等。

第2章线性表，介绍线性表的定义、线性表的两种主要存储结构和各种基本运算算法设计，最后通过示例讨论线性表的应用。

第3章栈和队列，介绍栈的定义、栈的存储结构、栈的各种基本运算算法设计和栈的应用,以及队列的定义、队列的存储结构、队列的各种基本运算算法设计和队列的应用。

第4章串，介绍串的基本概念、串的存储结构、串的各种基本运算算法设计和串的模式匹配算法。

第5章数组和广义表，介绍数组的基本概念、几种特殊矩阵的压缩存储方式、稀疏矩阵的压缩存储及相关算法设计，递归的定义和递归算法设计方法，广义表的定义、广义表的存储结构及相关算法设计方法。

第6章树和二叉树，介绍树的定义、树的逻辑结构表示方法、树的性质、树的基本运算和树的存储结构，二叉树的定义、二叉树的性质、二叉树的存储结构、二叉树的基本运算算法设计、二叉树的递归和非递归遍历算法、二叉树的构造、线索二叉树和哈夫曼树等。

第7章图，介绍图的定义、图的存储结构、图的基本运算算法设计、图的两种遍历算法以及图的应用(包括图的最小生成树、最短路径、拓扑排序和关键路径等)。

第8章查找，介绍查找的基本概念、线性表上的各种查找方法、树表上的

各种查找方法以及哈希表查找方法等。

第9章内排序,介绍排序的基本概念、插入排序方法、交换排序方法、选择排序方法、归并排序方法和基数排序方法,并对各种排序方法进行了比较。

第10章外排序,介绍外排序的概念、外排序的基本步骤,重点讨论了磁盘排序中的各种算法。

第11章数据结构和STL,介绍STL的概念和使用STL设计数据结构程序的基本方法。

另外,附录A给出书中部分算法清单,附录B给出各章练习题中单项选择题部分的答案。

本书主要特点如下:

☑ 结构清晰,内容丰富,文字叙述简洁明了,可读性强。

☑ 图文并茂,全书用270多幅图来表述和讲解数据的组织结构和算法设计思想。

☑ 力求归纳各类算法设计的规律,如单链表算法中很多是基于建表算法的,二叉树算法中很多是基于遍历算法的,图算法中很多是基于深度优先遍历的,如果读者掌握了建表算法、二叉树的遍历算法和图遍历算法,那么设计相关算法就会驾轻就熟了。

☑ 深入讨论递归算法设计方法。递归算法设计是数据结构课程中的难点之一,编者从递归模型入手,介绍了从求解问题中提取递归模型的通用方法,讲解了从递归模型到递归算法设计的基本规律。

☑ 书中含有大量的示例,所有算法设计示例均在Visual C++ 6.0环境中调试通过。

本书的编写工作得到湖北省教育厅和武汉大学教学研究项目《计算机科学与技术专业课程体系改革》的大力支持,并且清华大学出版社的魏江江主任全力支持本书的编写工作,编者在此一并表示衷心的感谢。为了便于教学,对于本书的课件和附录A的源程序等教学资源,读者可从清华大学出版社网站免费下载。

武汉大学计算机学院数据结构课程组的老师参加了本书的编写工作,中国劳动关系学院的陈良臣老师编写了部分章节。尽管编者不遗余力,由于水平所限,本书仍存在错误和不足之处,敬请广大教师和同学们批评指正,欢迎读者通过邮箱licb1964@126.com与编者联系,在此表示衷心的感谢。

编　者

2014年3月

CONTENTS

目录

CHAPTER 1

第1章 绪 论

数据结构作为一门独立的课程最早在美国的一些大学开设，1968 年，美国的 Donald E. Knuth 教授开创了数据结构的最初体系，他所著的《计算机程序设计技巧》系统地阐述了数据的逻辑结构和存储结构及其操作，是数据结构的经典之作。在 20 世纪 60 年代末出现了大型程序，结构程序设计成为程序设计方法学的主要内容，人们越来越重视数据结构，认为程序设计的实质是对确定的问题选择一种好的结构，加上设计一种好的算法，即程序＝数据结构＋算法。从 20 世纪 70 年代开始，数据结构得到了迅速的发展，编译程序、操作系统和数据库管理系统等均涉及数据元素的组织以及在存储器中的分配，数据结构技术成为设计和实现大型系统软件和应用软件的关键技术。

数据结构课程通过介绍一些典型数据结构的特性来讨论基本的数据组织和数据处理方法。通过本课程的学习，学生能够掌握数据结构（逻辑结构和存储结构）的基本概念和必要的基础知识，对定义在数据结构上的基本运算有较强的理解能力，学会分析、研究计算机加工的数据结构的特性，以便为应用涉及的数据选择适当的逻辑结构和存储结构，并能设计出较高质量的算法。

1.1 什么是数据结构

在了解数据结构的重要性之后，开始讨论数据结构的定义。本节先从一个简单的学生表例子入手，继而给出数据结构的严格定义，接着分析数据结构的几种类型，最后给出数据结构和数据类型之间的区别与联系。

1.1.1 数据结构的定义

在用计算机解决一个具体的问题时，大致需要经过以下几个步骤：

① 分析问题，确定数据模型。

② 设计相应的算法。

③ 编写程序，运行并调试程序，直至得到正确的结果。

寻求数学模型的实质是分析问题,从中提取操作的对象,并找出这些操作对象之间的关系,然后用数学语言加以描述。有些问题的数据模型可以用具体的数学方程等来表示,但更多的实际问题是无法用数学方程来表示的,这就需要从数据入手来分析并得到解决问题的方法。

数据是描述客观事物的数、字符以及所有能输入到计算机中并被计算机程序处理的符号的集合。例如,人们在日常生活中使用的各种文字、数字和特定符号都是数据。从计算机的角度看,数据是所有能被输入到计算机中,且能被计算机处理的符号的集合。它是计算机操作的对象的总称,也是计算机所处理信息的某种特定的符号表示形式(例如,A 班学生数据包含了该班全体学生记录)。

通常以**数据元素**作为数据的基本单位(例如,A 班中的每个学生记录都是一个数据元素),也就是说,数据元素是组成数据的、有一定意义的基本单位,在计算机中通常作为整体处理,有些情况下数据元素也称为元素、结点、记录等。有时候,一个数据元素可以由若干个数据项组成。**数据项**是具有独立含义的数据的最小单位,也称为域(例如,A 班中的每个数据元素(即学生记录)是由学号、姓名、性别和班号等数据项组成的)。

数据对象是性质相同的有限个数据元素的集合,它是数据的一个子集,如大写字母数据对象是集合 $C=\{'A','B','C',\cdots,'Z'\}$;1～100 的整数数据对象是集合 $N=\{1,2,\cdots,100\}$。默认情况下,数据结构中的数据指的都是数据对象。

数据结构是指所有数据元素以及数据元素之间的关系,可以看作是相互之间存在特定关系的数据元素的集合,因此,可时把数据结构看成是带结构的数据元素的集合。数据结构包括以下几个方面:

① 数据元素之间的逻辑关系,即数据的逻辑结构,它是数据结构在用户面前呈现的形式。

② 数据元素及其关系在计算机存储器中的存储方式,即数据的存储结构,也称为数据的物理结构。

③ 施加在该数据上的操作,即数据的运算。

数据的逻辑结构是从逻辑关系(主要指数据元素的相邻关系)上描述数据的,它与数据的存储无关,是独立于计算机的。因此,数据的逻辑结构可以看作是从具体问题抽象出来的数学模型。

数据的存储结构是逻辑结构用计算机语言的实现或在计算机中的表示(也称为映像),也就是逻辑结构在计算机中的存储方式,它是依赖于计算机语言的。一般情况下,只在高级语言(例如 C、C++、Java 语言)的层次上讨论存储结构。

数据的运算是定义在数据的逻辑结构之上的,每种逻辑结构都有一组相应的运算。例如,最常用的运算有检索、插入、删除、更新、排序等。数据的运算最终需要在对应的存储结构上用算法实现。

因此,数据结构是一门讨论"描述现实世界实体的数学模型(通常为非数值计算)及其之上的运算在计算机中如何表示和实现"的学科。

1.1.2 数据的逻辑结构

讨论数据结构的目的是为了用计算机求解问题,分析并弄清数据的逻辑结构是求解问

题的基础，也是求解问题的第一步。

数据的逻辑结构是用户根据需要建立起来的数据组织形式，它反映数据元素之间的逻辑关系而不是物理关系，是独立于计算机的。

数据中的数据元素之间可以有不同的逻辑关系，下面通过几个示例加以说明。

【例 1.1】 一个学生的高等数学成绩单如表 1.1 所示。这个表中的数据元素是学生成绩记录，每个数据元素由 3 个数据项（即学号、姓名和分数）组成，讨论其逻辑结构特性。

表 1.1 高等数学成绩单

学　号	姓　名	分　数
2011001	王华	90
2011010	刘丽	62
2011006	陈明	54
2011009	张强	95
2011007	许兵	76
2011012	李萍	88
2011005	李英	82

解：该表中的每一行称为一个记录，其逻辑结构特性是，只有一个开始记录（即姓名为王华的记录）和一个终端记录（也称为尾记录，即姓名为李英的记录），其余每个记录有一个前趋记录和一个后继记录，也就是说，记录之间存在一对一的关系，将具有这种逻辑特性的逻辑结构称为**线性结构**。

【例 1.2】 某高校组织结构示意图如图 1.1 所示。高校下设若干个学院和若干个处，每个学院下设若干个系，每个处下设若干个科或办公室，讨论其逻辑结构特性。

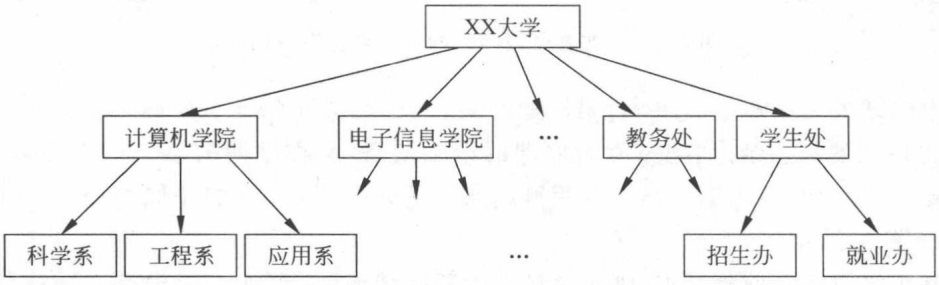


图 1.1 某高校组织结构示意图

解：该图中的每个长方形框表示一个结点，其逻辑结构特性是，只有一个开始结点（即大学名字结点），有若干个终端结点（如科学系等），每个结点对应零个或多个结点（终端结点对应零个结点），也就是说，结点之间存在一对多的关系，将具有这种逻辑特性的逻辑结构称为**树形结构**。

【例 1.3】 全国部分城市交通线路图如图 1.2 所示，讨论其逻辑结构特性。

解：该图中的每个城市表示一个结点，其逻辑结构特性是，每个结点和一个或多个结点相连，也就是说，结点之间存在多对多的关系，将具有这种逻辑特性的逻辑结构称为**图形结构**。

从上面几个示例可以看出，数据的逻辑结构主要是从数据元素之间的相邻关系来考虑

的,数据结构课程中仅讨论这种相邻关系,在实际应用中很容易将其推广到其他关系。

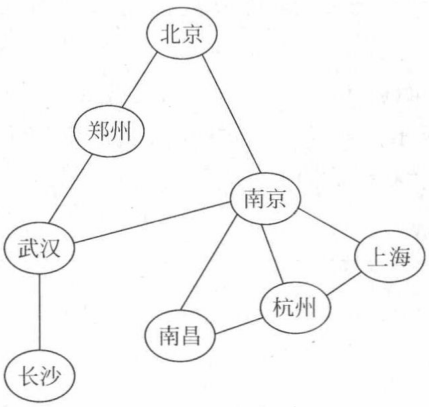


图 1.2　全国部分城市交通线路图

归纳起来,数据的逻辑结构有 4 种,除了前面介绍的线性结构、树形结构和图形结构以外,还有一种集合,集合中的数据元素没有任何相邻关系。

前面的表 1.1、图 1.1 和图 1.2 分别是高等数学成绩单、某高校组织结构和全国部分城市交通线路的逻辑结构表示,也就是说,数据的逻辑结构可以用多种方式来表示,假设用学号表示一个成绩记录,高等数学成绩单的逻辑结构也可以用图 1.3 表示。

为了更好地描述数据的逻辑结构,通常采用二元组表示数据的逻辑结构,一个二元组如下:

$$B=(D,R)$$

其中,B 是一种数据结构,D 是数据元素的集合,在 D 上数据元素之间可能存在多种关系,R 是所有关系的集合。即:

$$D=\{d_i \mid 1\leqslant i\leqslant n, n\geqslant 0\}$$

$$R=\{r_j \mid 1\leqslant j\leqslant m, m\geqslant 0\}$$

其中,d_i 表示集合 D 中的第 $i(1\leqslant i\leqslant n)$ 个数据元素(或结点),n 为 D 中数据元素的个数,若 $n=0$,则 D 是一个空集,此时 B 也就无结构可言。$r_j(1\leqslant j\leqslant m)$ 表示集合 R 中的第 j 个关系,m 为 R 中关系的个数,若 $m=0$,则 R 是一个空集,表明集合 D 中的数据元素之间不存在任何关系,彼此是独立的,这和数学中集合的概念是一致的。

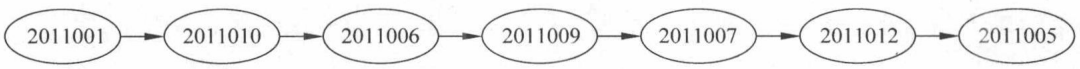


图 1.3　高等数学成绩单的另一种逻辑结构表示

R 中的某个关系 $r_j(1\leqslant j\leqslant m)$ 是序偶的集合,对于 r_j 中的任一序偶 $<x,y>(x,y\in D)$,把 x 称为序偶的第一结点,把 y 称为序偶的第二结点,又称序偶的第一结点为第二结点的**前趋结点**,称第二结点为第一结点的**后继结点**。例如在 $<x,y>$ 的序偶中,x 为 y 的前趋结点,y 为 x 的后继结点。

若某个结点没有前趋结点,则称该结点为**开始结点**;若某个结点没有后继结点,则称该结点为**终端结点**。

对于对称序偶,即满足这样的条件:若 $<x,y>\in r(r\in R)$,则 $<y,x>\in r(x,y\in D)$,可用圆括号代替尖括号,即 $(x,y)\in r$。

对于 D 中的每个数据元素,通常用一个关键字来唯一标识,例如,高等数学成绩表中学生成绩记录的关键字为学号。前面 3 个示例均可以采用二元组来表示其逻辑结构。

【例 1.4】 采用二元组表示前面 3 个例子的逻辑结构。

解:高等数学成绩单(假设学号为关键字)的二元组表示如下。

```
B1 = (D, R)
D = {2011001,2011010,2011006,2011009,2011007,2011012,2011005}
R = {r1}                          //表示只有一种逻辑关系
```

r_1 = {<2011001,2011010>,<2011010,2011006>,<2011006,2011009>,
<2011009,2011007>,<2011007,2011012>,<2011012,2011005>}

某高校组织结构(假设单位名为关键字)的二元组表示如下。

B_2 = (D,R)
D = {XX 大学,计算机学院,电子信息学院,…,教务处,学生处,…,科学系,工程系,应用系,…,
招生办,就业办}
R = {r_1}
r_1 = {< XX 大学,计算机学院>,< XX 大学,电子信息学院>,…, < XX 大学,教务处>,
< XX 大学,学生处>,…,<计算机学院,科学系>,<计算机学院,工程系>,
<计算机学院,应用系>,…,<学生处,招生办>,<学生处,就业办>}

全国部分城市交通线路图(假设城市名为关键字)的二元组表示如下。

B_3 = (D,R)
D = {北京,郑州,武汉,长沙,南京,南昌,杭州,上海}
R = {r_1}
r_1 = {(北京,郑州),(北京,南京),(郑州,武汉),(武汉,南京),(武汉,长沙),(南京,南昌),
(南京,上海),(南京,杭州),(南昌,杭州),(南昌,上海)}

1.1.3 数据的存储结构

问题求解最终是用计算机求解,在弄清数据的逻辑结构后,便可以借助计算机语言(本书采用 C++语言)实现其存储结构(或物理结构)。存储实现的基本目标是建立数据的机内表示,其包括两个部分,即数据元素的存储和数据元素之间关系的存储。

数据的存储结构应正确地反映数据元素之间的逻辑关系,也就是说,在设计某种逻辑结构对应的存储结构时,不仅要存储所有的数据元素,还要存储数据元素之间的关系。所以,将数据的存储结构称为逻辑结构的映像,将设计数据的存储结构称为从逻辑结构到存储器的映射,如图 1.4 所示。

归纳起来,数据的逻辑结构是面向问题的,而存储结构是面向计算机的,其基本目标是将数据及其逻辑关系存储到计算机的内存中。

图 1.4 存储结构是逻辑结构在内存中的映像

下面通过一个示例说明数据的存储结构的设计过程。

【例 1.5】 对于表 1.1 所示的高等数学成绩单,设计多种存储结构,并讨论各种存储结构的特性。

解:这里设计高等数学成绩单的两种存储结构。

存储结构 1:用 C++语言中的数组来存储高等数学成绩单,设计存放学生成绩记录的数组类型。

```
struct Stud1                    //学生成绩记录类型
{    int no;                    //存放学号
     char name[10];             //存放姓名
     double score;              //存放分数
};
```

定义一个 Stud1 结构体数组 st,用于存放高等数学成绩单:

```
Stud1 st[MaxSize];                          //存放记录的数组
st[0].no = 2011001; strcpy(st[0].name,"王华"); st[0].score = 90;
st[1].no = 2011010; strcpy(st[1].name,"刘丽"); st[1].score = 62;
st[2].no = 2011006; strcpy(st[2].name,"陈明"); st[2].score = 54;
st[3].no = 2011009; strcpy(st[3].name,"张强"); st[3].score = 95;
st[4].no = 2011007; strcpy(st[4].name,"许兵"); st[4].score = 76;
st[5].no = 2011012; strcpy(st[5].name,"李萍"); st[5].score = 88;
st[6].no = 2011005; strcpy(st[6].name,"李英"); st[6].score = 82;
n = 7;                                      //记录个数
```

st 数组便是高等数学成绩单的存储结构,其中,st[i](0≤i≤6)存放高等数学成绩单中逻辑序号为 i+1 的数据元素(逻辑序号从 1 开始),如图 1.5 所示。那么如何存放逻辑关系呢? 由于 st 数组中的数据元素存放在地址连续的存储单元中,即 st 中的各元素在内存中顺序存放,st[i](0≤i≤5)存放在 st[i+1]之前,而 st[i+1]存放在 st[i]之后,这样物理次序与逻辑次序完全相同,不再需要其他方式专门存储数据元素之间的逻辑关系。这种存储结构的特性是数据元素的逻辑关系和物理关系是一致的,称之为**顺序存储结构**,顺序存储结构是逻辑结构到存储结构的直接映射。

图 1.5 高等数学成绩单的顺序存储结构

存储结构 2:用 C++语言中的单链表来存储高等数学成绩单,设计存放学生成绩记录的结点类型。

```
struct Stud2                                //学生成绩单链表结点类型
{   int no;                                 //存放学号
    char name[10];                          //存放姓名
    double score;                           //存放分数
    Stud2 * next;                           //存放下一个结点指针
}
```

建立一个用于存放高等数学成绩单的单链表(开始结点地址为 head):

```
Stud2 * head;                               //学生单链表的开始结点
Stud2 * p1, * p2, * p3, * p4, * p5, * p6, * p7;
p1 = new Stud2();
p1 -> no = 2011001; strcpy(p1 -> name,"王华"); p1 -> score = 90;
p2 = new Stud2();
p2 -> no = 2011010; strcpy(p2 -> name,"刘丽"); p2 -> score = 62;
p3 = new Stud2();
p3 -> no = 2011006; strcpy(p3 -> name,"陈明"); p3 -> score = 54;
p4 = new Stud2();
p4 -> no = 2011009; strcpy(p4 -> name,"张强"); p4 -> score = 95;
p5 = new Stud2();
p5 -> no = 2011007; strcpy(p5 -> name,"许兵"); p5 -> score = 76;
```