

**程序设计课程设计**

**指导书**

软件学院 软件工程系

2018年6月1日

太原理工大学课程设计任务书

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学生姓名 |  | 专业班级 | 软件 | |
| 课程名称 | **程序设计课程设计（Programming Curriculum Design）** | | | |
| 设计名称 | 螺旋矩阵；消除类游戏；文本文件单词统计等 | | 设计周数 | 2 |
| 设计  任务  主要  设计  参数 | **1.基本要求**  掌握C或C++语言（或自学的其他语言）、结构化程序和面向对象程序设计方法、数据结构和离散数学理论知识，熟悉C或C++程序的开发环境及调试过程，巩固和加深对理论课中知识的理解，提高学生对所学知识的综合运用能力。  **2.培养学生以下技能**  ①培养学生查阅参考资料、手册的自学能力，通过独立思考深入钻研问题，学会自己分析、解决问题。  ②通过对所选题目分析，找出解决方法，设计算法，编制程序与调试程序。  ③能熟练调试程序，在教师的指导下，完成课题任务。  ④按课程设计报告的要求撰写设计报告。 | | | |
| 设计内容  设计要求 | 1.设计内容①螺旋矩阵；②消除类游戏；③文本文件单词统计；④哈夫曼编码；⑤构造可以使n个城市连接的最小生成树；⑥学生信息管理系统。2.设计要求 ①至少完成上述设计内容中的4个设计题目；②对每个题目要给出设计方案、功能模块划分、算法思想；③选择使用的数据结构；④给出题目的程序实现；⑤按要求撰写设计报告。 | | | |
| 主要参考  资 料 | 1.《程序设计课程设计》指导书；  2.《程序设计技术》、《数据结构》等课程教材；  3. 其他自选的相关资料。 | | | |
| 学生提交  归档文件 | 课程设计报告封面应给出专业、班级、姓名、学号、指导教师和完成日期。每个设计题目的内容包括以下几项：设计题目、问题描述、问题分析、功能实现、测试实例及运行结果、源程序清单。 | | | |

注意：

1.课程设计完成后，学生提交的归档文件应按照：封面—任务书—说明书—图纸的顺序进行装订上交（大张图纸不必装订）。

2.可根据实际内容需要续表，但应保持原格式不变。

**指导教师签名**： **日期**：2018.6.18

前 言

《程序设计课程设计》是 软件工程专业的重要实践性课程。目的在于培养学生分析问题和解决问题的能力，为学生提供了一个既动手又动脑，独立实践的机会。将数据结构、算法设计与分析、离散数学和C（C++）语言等课本上的理论知识和实际应用问题进行有机结合，提高学生问题分析、程序设计、程序调试及项目开发能力。为后续课程： 操作系统、软件工程，Web系列课程，,方向课等课程的学习奠定必要的实践基础。

本课程设计是利用数据结构、算法设计与分析、离散数学、C语言理论知识和实验课中学到的编程知识和编程技巧，通过布置具有一定难度、一定编程量的课程设计题目，利用C（C++）语言作为开发工具，使学生通过课程设计掌握高级编程语言的知识和编程技术，掌握程序设计的思想和方法，初步具备利用计算机求解实际问题的能力。

通过《程序设计课程设计》课程的学习，能够帮助学生加深理解数据结构、离散数学、C语言基本概念，达到培养学生良好程序设计的习惯和运用 C 语言编写程序解决实际问题的能力。使学生学会把书本知识用于解决实际问题，起到深化理解和灵活掌握教学内容的目的。同时使学生在程序设计方法及上机操作等基本技能和科学作风方面受到比较系统和严格的训练。

通过该课程设计，学生应该掌握C或C++语言程序设计、结构化程序和面向对象程序设计方法、数据结构和离散数学理论知识，熟悉C或C++程序的开发环境及C或C++程序的调试过程，巩固和加深对理论课中知识的理解，提高学生对所学知识的综合运用能力。

通过本课程设计训练，学生应该具备如下基本技能：

①培养学生查阅参考资料、手册的自学能力，通过独立思考深入钻研问题，学会自己分析、解决问题。

②通过对所选题目方案分析比较，确立方案，编制程序与调试程序。

③能熟练调试程序，在教师的指导下，完成课题任务。

④根据个人的设计调试过程，按课程设计报告的要求撰写设计报告。

选用教材及主要参考书：

1 教材

[1] 呼克佑. C语言程序设计.电子工业出版社,2013.

[2] 严蔚敏. 数据结构(C语言版).清华大学出版社,2012.

[3] 宋春花,吕进来.C++程序设计 (第二版).人民邮电出版社, 2017.8.

# 设计题目（6选4）

# 螺旋矩阵

1.1【问题描述】

输出下面的n（3≤n≤19）阶螺旋矩阵。输入n（=10），输出如下左图所示的螺旋矩阵。输入-n（=-10），输出如下右图所示的螺旋矩阵。

|  |  |  |
| --- | --- | --- |
| 1 2 3 4 5 6 7 8 9 10  36 37 38 39 40 41 42 43 44 11  35 64 65 66 67 68 69 70 45 12  34 63 84 85 86 87 88 71 46 13  33 62 83 96 97 98 89 72 47 14  32 61 82 95 100 99 90 73 48 15  31 60 81 94 93 92 91 74 49 16  30 59 80 79 78 77 76 75 50 17  29 58 57 56 55 54 53 52 51 18  28 27 26 25 24 23 22 21 20 19 |  | 1 36 35 34 33 32 31 30 29 28  2 37 64 63 62 61 60 59 58 27  3 38 65 84 83 82 81 80 57 26  4 39 66 85 96 95 94 79 56 25  5 40 67 86 97 100 93 78 55 24  6 41 68 87 98 99 92 77 54 23  7 42 69 88 89 90 91 76 53 22  8 43 70 71 72 73 74 75 52 21  9 44 45 46 47 48 49 50 51 20  10 11 12 13 14 15 16 17 18 19 |

**输入格式**

输入一个整数n或（-n），n表示要输出螺旋矩阵的阶数。

**输出格式**

输出右旋或左旋n阶螺旋矩阵。

1.2【测试数据】

|  |  |
| --- | --- |
| **输入：** | **输出：** |
| 6 | 1 2 3 4 5 6  20 21 22 23 24 7  19 32 33 34 25 8  18 31 36 35 26 9  17 30 29 28 27 10  16 15 14 13 12 11 |
| -6 | 1 20 19 18 17 16  2 21 32 31 30 15  3 22 33 36 29 14  4 23 34 35 28 13  5 24 25 26 27 12  6 7 8 9 10 11 |

# 消除类游戏

2.1【问题描述】

消除类游戏是深受大众欢迎的一种游戏，游戏在一个包含有n行m列的游戏棋盘上进行，棋盘的每一行每一列的方格上放着一个有颜色的棋子，当一行或一列上有连续三个或更多的相同颜色的棋子时，这些棋子都被消除。当有多处可以被消除时，这些地方的棋子将同时被消除。

2.2【基本要求】

现在给你一个n行m列的棋盘(1≤n,m≤30)，棋盘中的每一个方格上有一个棋子，请给出经过一次消除后的棋盘。

请注意：一个棋子可能在某一行和某一列同时被消除。

**输入数据格式：**

输入的第一行包含两个整数n, m，用空格分隔，分别表示棋盘的行数和列数。接下来n行，每行m个整数，用空格分隔，分别表示每一个方格中的棋子的颜色。颜色使用1至9编号。

**输出数据格式：**

输出n行，每行m个整数，相邻的整数之间使用一个空格分隔，表示经过一次消除后的棋盘。如果一个方格中的棋子被消除，则对应的方格输出0，否则输出棋子的颜色编号。

2.3【测试数据】

为方便调试程序，可将输入数据先写入一个文本文件，然后从文件读取数据处理，这样可避免每次运行程序时都要从键盘输入数据。

**测试数据一**

|  |  |
| --- | --- |
| **输入：** | **输出：** |
| 4 5  2 2 3 1 2  3 4 5 1 4  2 3 2 1 3  2 2 2 4 4 | 2 2 3 0 2  3 4 5 0 4  2 3 2 0 3  0 0 0 4 4 |

**输出说明：**

棋盘中第4列的1和第4行的2可以被消除，其他的方格中的棋子均保留。

**测试数据二**

|  |  |
| --- | --- |
| **输入：** | **输出：** |
| 4 5  2 2 3 1 2  3 1 1 1 1  2 3 2 1 3  2 2 3 3 3 | 2 2 3 0 2  3 0 0 0 0  2 3 2 0 3  2 2 0 0 0 |

**输出说明：**

棋盘中所有的1以及最后一行的3可以被同时消除，其他的方格中的棋子均保留。

# 文本文件单词统计

3.1【问题描述】

**假设有如下的英文文本文档**：(此处为太原理工大学学校简介英文版)

**TAIYUAN UNIVERSITY OF TECHNOLOGY**

Taiyuan University of Technology (TUT) has its history traced all the way back to the Western Learning School of Shanxi Grand Academy (1902), which was one of the three earliest national universities in China. With the tradition and development of over 100 years, TUT is now a general university with engineering as the major, sciences and technology integrated and coordinate development of multiple disciplines. It is a university that is included in the “Project 211”-- the national higher education promotion program for 100 top universities in China.

……

Recollecting the centennial history, generations of TUT have created its mission and glory of a century with responsibility and confidence; expecting the promising tomorrow, over 30,000 TUT students and faculty are producing splendor and perspectives by their wisdom and diligence. In the new era, Taiyuan University of Technology, following the Conception of Scientific Development, is determined to further the reformation on education, to reinforce the teaching management so as to upgrade its teaching and researching levels. Taiyuan University of Technology will be turning itself into a research-based university.

设计C或C++程序，统计在这样的英文文本文件中，出现了多少个单词（不区分大小写），每个单词出现了几次。连续的英文字符(不包括数字)都认为是单词，单词之间用空格或标点符号分隔。

3.2【设计需求及分析】

要统计英文文本文件中出现了哪些单词，就要从文件中读取字符，读取出来的连续英文字符认为是一个单词，遇空格或标点符号单词结束。

使用线性表记录单词以及每个单词出现的次数。线性表中的单词按字典顺序存储。

**线性表的顺序存储结构如下：（必须使用如下定义的存储结构，否则无效）**

#define LIST\_INIT\_SIZE 100 //线性表存储空间的初始分配量

#define LISTINCREMENT 10 //线性表存储空间的分配增量

typedef struct{

char word[21] //存储单词，不超过20个字符

int count; //单词出现的次数

} ElemType; //顺序表元素类型

typedef struct{

ElemType \*elem; //存储空间基址

int length; //当前长度

int listsize; //当前分配的存储容量

} Seqlist; //顺序表类型

3.3【功能设计】

3.3.1 实现顺序表的基本操作（必须使用下面给定的函数名和参数表，否则无效）

⑴顺序表的初始化：InitList(SqList &L)

⑵顺序表上查找指定的单词：LocateElem(SqList &L,char \*s)

若找到，单词的出现次数增1，返回0，否则返回该单词的插入位置。

⑶在顺序表上插入新的单词：InsertList(SqList &L,int i，char \*s)

要求按字典顺序有序。新单词的出现次数为1。

⑷输出顺序表上存储的单词统计信息：PrintList(SqList &L)

输出文件中每个单词出现的次数以及文件中总的单词数(可输出到文件中)。

3.3.2 统计单词数

统计过程如下：

（1）输入要统计单词的文本文件名，打开相应的文件；

（2）初始化顺序表；

（3）从文本文件中读取字符，直到文件结束。具体描述如下：

while (读文件没有结束结束)

{

过滤单词前的非字母字符；

读取一个单词，以字符串形式存储在一个字符数组中；

在线性表中查找该单词，若找到，单词的出现次数加1，否则返回其插入位置；

上一步中，若没找到，则进行插入操作；

处理下一个单词。

}

（4）关闭文件，输出统计结果。

3.4【测试数据】

将上述给定的英文文档写入文本文件：tyut.txt作为测试数据文件。

# 哈夫曼编码

4.1【问题描述】

假设通讯电文由m（3≤m≤20）个字符组成，每个出现的频率分别为：P1，P2，…，Pn。请设计程序，给出这m个字符的哈夫曼编码。

4.2【问题分析】

把m个字符的出现频率：P1，P2，…，Pm 作为m个权值，构造m个权值的哈夫曼树，按哈夫曼树给出这m个字符的哈夫曼编码。

m个权值的赫夫曼树有2m-1个结点，采用数组存储哈夫曼树中的结点，树中的每个结点结构为：

|  |  |  |  |
| --- | --- | --- | --- |
| w | parent | left | right |

其中：w为结点的权值，parent是结点的双亲在数组中的下标，left和right分别是结点的左右孩子在结点数组中的下标。

另外，与每个叶子结点对应字符的编码存放在一个字符数组中，因为编码是不等长的，即每个编码所占用的空间不等，应此存储编码串的空间需动态分配。

**输入数据格式：**

输入的第一行包含一个整数 m，表示字符的个数。接下来m行，每行两个数据，字符和权值，用空格分隔，表示字符和字符的权值。

**输出数据格式：**

输出m行，每行两个数据，字符和字符的哈夫曼编码，用空格分隔。

4.3【存储结构定义和功能设计】

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define SUCCESS 1

#define FAILED -1

#define CODE\_NUM 20 /\*最多字符数\*/

#define MAX\_WEIGHT 32767 /\*最大权值，代表无穷大\*/

typedef struct node\_st{ /\*定义树结点结构体类型\*/

int w;

int parent,left,right;

} NODE;

typedef struct char\_code\_st{ /\*定义字符、编码结构体类型\*/

char ch; /\*字符\*/

int weight; /\*权值\*/

char \*huffmancode; /\*指向哈夫曼编码的指针\*/

} HCODE;

int CreatHuffmanCode(HCODE node[],int nodenum); /\*求赫夫曼编码函数\*/

void GetTwoNode(NODE huffmantree[],int n,int \*s1,int \*s2); /\*找权值最小的两棵子树函数\*/

int main()

{

int i,nodenum;

HCODE node[CODE\_NUM]; /\*定义数组，存储字符、权值和哈夫曼编码\*/

scanf("%d",&nodenum); /\*输入字符个数\*/

for(i=0;i<nodenum;i++) /\*输入字符和权值\*/

scanf("%\*c%c%\*c%d",&node[i].ch,&node[i].weight);

CreatHuffmanCode(node,nodenum); /\*求哈夫曼编码\*/

for(i=0;i<nodenum;i++) /\*输出字符及其哈夫曼编码\*/

printf("%c %s\n",node[i].ch,node[i].huffmancode);

}

**请设计函数：**

1. int CreatHuffmanCode(HCODE node[],int nodenum);
2. void GetTwoNode(NODE huffmantree[],int n,int \*s1,int \*s2);

4.4【测试数据】

为方便调试程序，可将输入数据先写入一个文本文件，然后从文件读取数据处理，这样可避免每次运行程序时都要从键盘输入数据。

|  |  |
| --- | --- |
| **输入：** | **输出：** |
| 8  a 12  b 23  c 9  d 2  e 27  f 6  g 15  h 5 | a 001  b 01  c 000  d 11010  e 10  f 1100  g 111  h 11011 |

# 构造可以使n个城市连接的最小生成树

5.1【问题描述】

给定一个地区的n个城市间的距离网，用Prim算法或Kruskal算法建立最小生成树，并计算得到的最小生成树的代价。

5.2【基本要求】

1、城市间的距离网采用邻接矩阵表示，邻接矩阵的存储结构定义采用课本中给出的定义，若两个城市之间不存在道路，则将相应边的权值设为自己定义的无穷大值。

2、要求在屏幕上显示得到的最小生成树中包括了哪些城市间的道路，并显示得到的最小生成树的代价。

3、表示城市间距离网的邻接矩阵（要求至少6个城市，10条边）。

5.3【测试数据】

学生自主确定或参考图4-1。

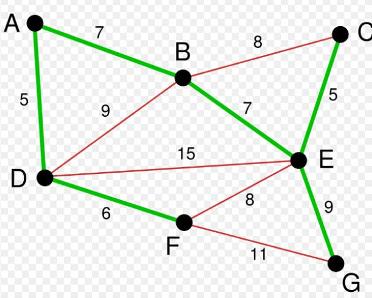


图4-1 一个带权图(网络)

# 学生信息管理系统

6.1【问题描述】

大学里有各种类型的学生，校方需要对这些学生的信息进行计算机管理。所开发的软件应包括各类学生的添加、修改、删除和查找等功能。考虑到软件的可重用性、可扩展性和可维护性，校方决定采用面向对象的程序设计方法来开发系统。学生信息需要以文件方式保存到计算机硬盘中。另外，系统的用户界面应该尽可能友好，方便用户使用。

6.2【设计需求及分析】

1. 使用C++语言开发，充分利用面向对象程序设计的类、对象、继承、封装和多态性等
2. 概念来设计和实现该管理系统。
3. 设计一个Person（人员）类，考虑到通用性，只抽象出所有类型人员都具有的属性：name（姓名）, id（身份证号），gender（性别），birthday（出生日期）等等。其中“出生日期”为内嵌子对象，是一个Date（日期）类型，Date类具有属性: year(年)，month（月），day（日）。用成员函数实现对人员信息的录入和显示等必要功能操作。
4. 从Person类派生出Student（学生）类，添加属性： studentNo（学号），schoolName（学校），classIn (班级)。从Person类派生出Teacher（教师）类，添加属性：teacherNo（教师编号），schoolName（学校），department（部门）。
5. 从Student类中派生出UnderGraduate（本科生）类，添加属性：major（专业）。从Student类中派生出Graduate（研究生）类，添加属性：direction（研究方向），adviserName（导师姓名）。
6. 从Graduate类和Teacher类派生出TＡ（助教博士生）类。
7. 写程序测试上述各类，看能否正常运行。
8. 构建必要的辅助类，实现对本科生、研究生和助教博士生的添加、修改、删除、查询管理。
9. 根据需要定义类的构造函数、析构函数、拷贝构造函数、成员函数。必要时重载函数。
10. 要求将Person类设置为虚基类，以消除其派生类成员访问的二义性问题（注意在虚基类各级派生类的构造函数实现时调用虚基类的构造函数）。
11. 要求在Person类中定义虚函数displayDetails（），用于显示当前对象的信息；同时定义虚函数inputData( )，用于从键盘获取当前对象的信息。Person类所有派生类也要定义同名虚函数，使程序可以实现动态多态性。
12. 用菜单方式设计主控模块程序。
13. 对程序源代码要给出各部分的详细注释，这也是该题目的考核重点之一。
14. 用UML语言描述系统用到的类及其关系。

6.3【设计功能的实现】（用C或C++语言描述）

//说明：此内容由学生自己设计完成。

//以下代码仅供参考。

程序框架:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (C), 2010, Tyut

File name: main.cpp

Author: gaobaolu Version: 1.0 Date: 2010.6.28

Description: 应用程序主函数

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <cstdlib>

#include <iostream>

#include "date.h"

#include "person.h"

#include "student.h"

#include "teacher.h"

#include "undergraduate.h"

#include "graduate.h"

#include "ta.h"

#include "undergraduateManager.h"

using namespace std;

int main(int argc, char \*argv[])

{ int choiceN;

UndergraduateManager unMan;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\*|\*| |\*|\*"<<endl;

cout<<"\*|\*| 欢迎您使用学生管理系统 |\*|\*"<<endl;

cout<<"\*|\*| |\*|\*"<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

do{

cout << "<---------------------------------------------------->" << endl;

cout << " \n \t\t 1:本科生管理 ";

cout << " \n \t\t 2:研究生管理 ";

cout << " \n \t\t 3.助教博士生管理 ";

cout << " \n \t\t 0:离开 ";

cout << endl;

cout << "<---------------------------------------------------->" << endl;

cout << "请选择：" << endl;

cin >> choiceN;

switch(choiceN)

{

case 1:

unMan.dataManage();

break;

case 2:

//

break;

case 3:

//

break;

default:

break;

}

}while(choiceN!=0);

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

cout<<"\*|\*| 感谢使用学生管理系统 |\*|\*"<<endl;

cout<<" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\a"<<endl;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (C), 2010, Tyut

File name: undergraduateManager.h

Author: gaobaolu Version: 1.0 Date: 2010.6.28

Description: 本科生管理类

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef UNDERGRADUATE\_MANAGER\_H

#define UNDERGRADUATE\_MANAGER\_H

#include <iostream>

#include <string>

#include <fstream>

#include "undergraduate.h"

using namespace std;

/\* Define a Class : UndergraduateManager 本科生管理类\*/

class UndergraduateManager

{

private:

int top; //记录指针

Undergraduate undergraduates[100]; //本科生记录

public:

UndergraduateManager();//构造函数,将Undergraduate.txt读到undergraduates[]中

int queryByNo(string sno);//按本科生号查找 //找到：返回数组下标//没找到：返回-1

void clearStudent(); //删除所有本科生信息

int addStudent(Undergraduate s); //添加本科生,需要先查找是否存在

int modifyStudent(string sno); //修改学生信息 ,需要先查找是否存在

int deleteStudent(string sno);//删除本科生，删除前先查找其是否存在

int queryStudent(string sno);//查找本科生,查到则显示,否则提示未查到

void displayAll();//输出所有本科生信息

void dataManage(); //本科生库维护

void dataSave();

void dataRead();

~UndergraduateManager();//析构函数,将undergraduates[]写入Undergraduate.txt文件中

};

//构造函数,将Undergraduate.txt读到undergraduates[]中

UndergraduateManager::UndergraduateManager()

{

dataRead();

}

//按本科生号查找

//找到：返回数组下标

//没找到：返回-1

int UndergraduateManager::queryByNo(string sno)

{

for(int i=0;i<=top;i++)

if (undergraduates[i].getStudentNo()==sno)

return i;

return -1;

}

//删除所有本科生信息

void UndergraduateManager::clearStudent()

{

top=-1;

}

//添加本科生,需要先查找是否存在

int UndergraduateManager::addStudent(Undergraduate s)

{

int p=queryByNo(s.getStudentNo());

if (p==-1)

{

top++;

undergraduates[top]= s;

dataSave();//保存

return 1;

}

else

{

cout<<"--------->此学生已经存在 !<----------"<<endl<<endl;

return 0;

}

}

//修改科生，删除前先查找其是否存在

int UndergraduateManager::modifyStudent(string sno)

{

int p=queryByNo(sno);

if (p==-1)

{

cout<<"--------->此学生不存在 !<----------"<<endl<<endl;

return 0;

}

else

{

cout << "请输入该生的新信息: " << endl<<endl;

undergraduates[p].inputData();

dataSave();//保存

return 1;

}

}

//删除本科生，删除前先查找其是否存在

int UndergraduateManager::deleteStudent(string sno)

{

int p=queryByNo(sno);

if (p==-1)

{

cout<<"--------->此学生不存在 !<----------"<<endl<<endl;

return 0;

}

else

{

for(int i = p; i < top ; i++)

undergraduates[i]=undergraduates[i+1];

top--;

cout << "--------->删除完成!<----------" << endl<<endl;

dataSave();//保存

return 1;

}

}

//查找科生

int UndergraduateManager::queryStudent(string sno)

{

int p=queryByNo(sno);

if (p==-1)

{

cout<<"--------->此学生不存在 !<----------"<<endl<<endl;

return 0;

}

else

{

cout<<"--------->此学生存在:<----------"<<endl<<endl;

undergraduates[p].displayDetails();

return 1;

}

}

//输出所有本科生信息

void UndergraduateManager::displayAll()

{

for (int i=0;i<=top;i++)

{ cout << "--------第"<< i <<"个学生情况----------" << endl<<endl;

undergraduates[i].displayDetails();

}

}

//析构函数,将undergraduates[]写入Undergraduate.txt文件中

UndergraduateManager::~UndergraduateManager()

{

dataSave();

}

void UndergraduateManager::dataManage()

{

int choice=1;

string sstudentNo;

Undergraduate s;

while (choice!=0)

{

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "\t\t本科生维护\n";

cout << "^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^" << endl;

cout << "\n \t\t 1:新增 ";

cout << "\n \t\t 2:修改 ";

cout << "\n \t\t 3:删除 ";

cout << "\n \t\t 4:查找 ";

cout << "\n \t\t 5:显示 ";

cout << "\n \t\t 6:全部删除 ";

cout << "\n \t\t 0:退出 " ;

cout << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << "请选择：" << endl;

cin >> choice;

switch (choice)

{

case 1:

s.inputData();

addStudent(s);

break;

case 2:

cout << " 请输入学号:";

cin >> sstudentNo;

modifyStudent(sstudentNo);

break;

case 3:

cout << " 请输入学号:";

cin >> sstudentNo;

deleteStudent(sstudentNo);

break;

case 4:

cout << " 请输入学号:";

cin >> sstudentNo;

queryStudent(sstudentNo);

break;

case 5:

displayAll();

break;

case 6:

clearStudent();

break;

default:

break;

}

}

}

void UndergraduateManager::dataSave()//存储资料函数,将read[]写入Undergraduate.txt文件中

{

fstream file("Undergraduate.dat",ios::out);

for (int i=0;i<=top;i++)

file.write((char \*)&undergraduates[i],sizeof(undergraduates[i]));

file.close();

}

void UndergraduateManager::dataRead() //构造函数,将Undergraduate.txt读到read[]中

{

Undergraduate s;

top=-1;

fstream file("Undergraduate.dat",ios::in);

while (1)

{

file.read((char \*)&s,sizeof(s));

if (!file) break;

top++;

undergraduates[top]=s;

}

file.close();

}

#endif //UNDERGRADUATE\_MANAGER\_H

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (C), 2010, Tyut

File name: date.h

Author: gaobaolu Version: 1.0 Date: 2010.6.28

Description: 日期类

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef DATE\_H

#define DATE\_H

#include <iostream>

using namespace std;

/\* Define a Class : Date\*/

/\* with attributes: year, month, and day, and \*/

/\* operations: getYear, getMonth,.... \*/

class Date

{

private:

int year;

int month;

int day;

public:

Date(){year = 0; month = 0; day = 0;}

Date(int yy,int mm,int dd){year = yy; month = mm; day = dd;}

Date(Date& d){year = d.year; month = d.month; day = d.day;}

~Date(){}

void setYear(int yy){ year = yy;}

void setMonth(int mm){ month = mm;}

void setDay(int dd){ day = dd;}

int getYear(){return year;}

int getMonth(){return month;}

int getDay(){return day;}

void inputDate()

{

cout << "年："; cin >> year;

cout << "月: "; cin >> month;

cout << "日: "; cin >> day;

}

void displayDate(){ cout << year << "/" << month << "/" << day << endl; }

};

#endif //DATE\_H

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (C), 2010, Tyut

File name: person.h

Author: gaobaolu Version: 1.0 Date: 2010.6.28

Description: 人员类

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef PERSON\_H

#define PERSON\_H

#include <iostream>

#include <cstring>

#include <string>

using namespace std;

/\* Define a Class : Person \*/

/\* with attributes: name, id,gender, birthday \*/

/\* operations: getName, getId, and setId. \*/

/\* person is the base class \*/

//该类为虚基类

class Person

{ protected:

char name[20];//姓名

char id[18]; //身份证号

char gender[2];//性别

Date birthday;//出生日期

public:

Person();

Person(char\* sname, char\* sid, char\* sgender,int yy,int mm ,int dd);

~Person(){}

void setName(char\* sname){ strcpy(name,sname); }

char\* getName(){ return name; }

void setId(char\* sid){ strcpy(id,sid); }

char\* getId(){ return id;}

void setGender(char\* sgender){ strcpy(gender,sgender); }

char\* getGender(){ return gender; }

void setBirthday(Date d){ birthday = d; }

Date getBirthday() { return birthday; }

virtual void inputData();//输入数据

virtual void displayDetails(); //显示数据

};

Person::Person()

{

strcpy(name,"NA");

strcpy(gender,"男");

strcpy(id,"000");

//注意：这里没有给birthday赋值,系统会调用其默认构造函数

//也可以

//Date d(1980,1,1);

//birthday = d;

}

Person::Person(char\* sname, char\* sid, char\* sgender,int yy,int mm ,int dd):birthday(yy,mm,dd)

{

strcpy(name,sname);

strcpy(id,sid);

strcpy(gender,sgender);

}

void Person::inputData()

{

cout << "姓名：" ;

cin >> name ;

cout << "身份证号：" ;

cin >> id ;

cout << "性别：";

cin >> gender ;

cout << "出生日期：" << endl;

birthday.inputDate();

}

void Person::displayDetails()

{

cout << "姓名：" <<name <<endl;

cout << "身份证号：" <<id <<endl;

cout << "性别：" <<gender<<endl;

cout << "出生日期：";

birthday.displayDate();

}

#endif //PERSON\_H

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (C), 2010, Tyut

File name: teacher.h

Author: gaobaolu Version: 1.0 Date: 2010.6.28

Description: 教师类

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#ifndef TEACHER\_H

#define TEACHER\_H

#include <iostream>

#include <string>

#include <cstring>

#include "person.h"

using namespace std;

/\* Define a Class : Teacher \*/

/\* with attributes: \*/

/\* operations: \*/

class Teacher: virtual public Person

{ protected:

char teacherNo[5]; //教师编号

char schoolName[20]; //学校名称

char department[20]; //部门

public:

Teacher();

Teacher(char\* sname, char\* sid, char\* sgender,int yy,int mm ,int dd

,char\* steacherNo,char\* sschool,char\* sdepartment);

~Teacher(){}

void setTeacherNo(char\* sno){ strcpy(teacherNo,sno); }

char\* getTeacherNo(){ return teacherNo;}

void setSchoolName(char\* sschool){ strcpy(schoolName,sschool); }

char\* getSchoolName(){ return schoolName; }

void setDepartment(char\* sdepartment){ strcpy(department,sdepartment); }

char\* getDepartment(){ return department; }

virtual void inputData();//输入数据

virtual void displayDetails(); //显示数据

};

Teacher::Teacher():Person()

{

strcpy(teacherNo,"001");

strcpy(schoolName,"defaultSchoolName");

strcpy(department,"defaultDepartment");

}

Teacher::Teacher(char\* sname, char\* sid, char\* sgender,int yy,int mm ,int dd

,char\* steacherNo,char\* sschool,char\* sdepartment)

:Person(sname,sid,sgender,yy,mm,dd)

{

strcpy(teacherNo,steacherNo);

strcpy(schoolName,sschool);

strcpy(department,sdepartment);

}

void Teacher::inputData()

{

Person::inputData();

cout << "教师编号：" ;cin >> teacherNo ;

cout << "学校：" ;cin >> schoolName ;

cout << "部门：" ;cin >> department ;

}

void Teacher::displayDetails()

{

Person::displayDetails();

cout << "教师编号：" << teacherNo <<endl;

cout << "学校：" << schoolName <<endl;

cout << "部门：" << department << endl;

}

#endif //TEACHER\_H

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (C), 2010, Tyut

File name: student.h

Author: gaobaolu Version: 1.0 Date: 2010.6.28

Description: 学生类

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (C), 2010, Tyut

File name: graduate.h

Author: gaobaolu Version: 1.0 Date: 2010.6.28

Description: 研究生类

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (C), 2010, Tyut

File name: undergraduate.h

Author: gaobaolu Version: 1.0 Date: 2010.6.28

Description: 本科生类

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Copyright (C), 2010, Tyut

File name: ta.h

Author: gaobaolu Version: 1.0 Date: 2010.6.28

Description: 助教博士生类

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

6.4【实例测试及运行结果】

输入多个本科生、研究生和助教博士生的数据。

//说明：此内容由学生自己设计完成。

6.5【实现提示】

//说明：学生自己补充。



课程设计

课程名称： **程序设计课程设计**

设计名称：

专业班级： 学号：

学生姓名：

指导教师：

2018年 6月 29 日