注：测试数据用黄色 结果和回答问题用绿色 源代码命名格式为实验号-题目号.cpp

实验六题目二三未做 实验八题目二未做

## 3.1 实验一 数据表示与程序控制

### 题目3 简单应用—求圆柱体的体积

输入下面程序代码，编译运行。



**[**测试数据**]**

运行时输入：

5 8

-2 1

结果：

radius: 5 high:8

The volume is: 628.318

radius: -2 high: 1

The volume is: 12.56636

**[**思考与扩展**]**

1、#define PI 3.14159 是符号常量，用const 常量定义PI。const double PI=3.14159

2、如果运行时输入 3.8 5 程序的输出结果是什么，正确吗？将程序中变量radius, high 改为double 类型，然后输入上面的测试数据，结果怎么样？说明原因。

结果：radius:3 high:5 The volume is: 141.37155

改为double后 结果：radius:3.8 high:5 The volume is: 226.822798

原先的radius是int型当输入3.8时radius的值被给3

3、注意输入提示和输出提示语句，尝试修改输入数出提示语句。

4、程序欣赏

#include<iostream.h>

class container

{

protected:

double radius;

public:

container(double radius)

{

container::radius=radius;

}

virtual double surface\_area()=0;

virtual double volume()=0;

};

class cube:public container

{

public:

cube(double radius):container(radius){};

double surface\_area()

{

return radius\*radius\*6;

}

double volume()

{

return radius\*radius\*radius;

}

};

class sphere:public container

{

public:

sphere(double radius):container(radius){};

double surface\_area()

{

return 4\*3.1416\*radius\*radius;

}

double volume()

{

return 3.1416\*radius\*radius\*radius\*4/3;

}

};

class cylinder:public container

{

double height;

public:

cylinder(double radius,double height):container(radius)

{

cylinder::height=height;

}

double surface\_area()

{

return 2\*3.1416\*radius\*(height+radius);

}

double volume()

{

return 3.1416\*radius\*radius\*height;

}

};

void main()

{

container \*p;

cube obj1(10);

sphere obj2(6);

cylinder obj3(4,5);

p=&obj1;

cout<<"输出结果："<<endl;

cout<<"正方体表面积："<<p->surface\_area()<<endl;

cout<<"正方体体积："<<p->volume()<<endl;

p=&obj2;

cout<<"球体表面积："<<p->surface\_area()<<endl;

cout<<"球体体积："<<p->volume()<<endl;p=&obj1;

p=&obj3;

cout<<"圆柱体表面积："<<p->surface\_area()<<endl;

cout<<"圆柱体体积："<<p->volume()<<endl;

}

### 题目4 计算公司员工周工资

编写一个程序，输入某雇员的每周工作时间（以小时计）和每小时的工资数，计算并输出他的工资。若雇员周工作小时超过40 小时，则超过部分按原工资的1.5 倍的加班工资来计算。若雇员每周工作小时超过60 小时，则超过60 的部分按原工资的3 倍的加班工资来计算，而40 到60 小时的工资仍按照原工资的1.5 倍的加班工资来计算。

1、要求有输入提示和输出提示，如要输入雇员的工作时间和每小时的工资值时，可以

提示：

“Please input employee’s work time and wage\_per\_hour:”

输出时，提示：

“The employee’s wage :”。

2、给程序加上注释，并注意排版格式，使得其清晰可读。

**[**实验提示**]**

1、从本题目开始，将涉及到一些简单的算法，请务必按实验要求养成良好编程的习惯，再次强调：

代码要写整齐，注意排板规范，请再次仔细研读附录〈〈编程规范〉〉。

给变量起个好名字，不要什么变量都用 a,b,c…等，变量最好用英文单词或单词的缩写命名，变量名字要体现变量在程序的含义和作用。

要有必要的注释！！，养成这一良好的习惯很重要，请记住，“**代码只被书写一次，而它**

**被修改和阅读的次数可能是无限的”**

**[**测试数据**]**

输入：30 4

输入：45 4.5

输入：60 5

结果：

The employee's wage :120

The employee's wage :213.75

The employee's wage :350

源代码：

#include <iostream>

using namespace std;

int main() {

double time, wage\_per\_hour, wage;

int flag;

cout << "Please input employee's work time and wage\_per\_hour:";

cin >> time >> wage\_per\_hour;

while (time < 0 || time > 168) {

cout << "输入时间非法，请重新输入:";

cin >> time;

}

if (time < 40)

flag = 0;

else if (time <= 60)

flag = 1;

else if (time > 60)

flag = 2;

switch (flag) {

case 0:

wage = time \* wage\_per\_hour;

break;

case 1:

wage = ((time - 40) \* 1.5 + 40) \* wage\_per\_hour;

break;

case 2:

wage = ((time - 60) \* 3 + (60 - 20) \* 1.5 + 40) \* wage\_per\_hour;

break;

}

cout << "The employee's wage :" << wage << endl;

}

### 题目5 设计一个计算+、-、\*、/的程序

程序循环从标准输入读入表达式：

输入： 4+2 回车

则输出：4+2=6

输入： 4\*2 回车

则输出：4\*2=8

………

当用户输入：０ ０ ０ 回车时程序退出

实现要求：

1、有输入提示和输出提示，如要输入表达式时，可以提示：

输出时，提示：

" 4+2=6"。

计算在用户输入的表达式为0 0 0 时程序结束。

2、为程序加上注释，使得其清晰可读。

3、尝试利用调试程序来修改程序的逻辑错误。

**[**测试数据**]**

自己设计一组测试数据，要求覆盖switch 选择结构的中所有路径即每条case 语句都被

执行一次。

结果：

pealse intput arithmetic expression:2+1

2+1=3

pealse intput arithmetic expression:2-1

2-1=1

pealse intput arithmetic expression:2\*1

2\*1=2

pealse intput arithmetic expression:2/1

2/1=2

pealse intput arithmetic expression:2%1

2%1=0

源代码：

/\*

\* 1-5.cpp

\*

\* Created on: 2018年4月24日

\* Author: asus

\*/

#include <iostream>

using namespace std;

int main(){

double a,b,result;

char op;

do {

cout << "pealse intput arithmetic expression: ";

cin >> a >> op >> b;

switch (op) {

case '+':

result = a + b;

cout << a << op << b << "=" << result << endl; break;

case '-':

result = a - b;

cout << a << op << b << "=" << result << endl; break;

case '\*':

result = a \* b;

cout << a << op << b << "=" << result << endl; break;

case '/':

result = a / b;

cout << a << op << b << "=" << result << endl; break;

case '%':

result = (int) a % (int) b;

cout << a << op << b << "=" << result << endl; break;

}

} while (a != 0 && b != 0 && op != 0);

}

**[**思考与扩展**]**

1、为什么要在while 循环体中输入三个数据（表达式）？表达式有三项

2、使用switch 语句中的break 起什么作用？在此处打断 不然此代码后面的case也会被执行

### 题目6 计算生日是星期几

编写一个程序，只要输入年月日，就能回答那天是星期几。

**[**实验提示**]**

1、已知公元1 年1 月1 日是星期一

2、算法提示：如果输入的是2006 年3 月12 日，计算方法是

(1) 先计算从公元0001 年初（0001-1-1）到2005 年末(2005-12-31)共多少天，注意其中

有闰年，闰年是366 天，平年是365 天。闰年的判断条件满足：

(( year%4==0 && year%100!=0 )!!(year%400==0))

(2) 然后再计算2006 年1 月1 日到2006-3-12 是多少天。注意同样判断本年度是否闰年，

即二月份是28 天还是29 天。

(3) 上述两部分相加即从公元0001 年初（0001-1-1）到2006 年末(2006-3-12)的天数，

用该天数模7（total%7）即得到星期数。

２、用户多次循环输入数据并计算可采用 while 语句实现:

//首先输入三个数据：

cout<< "please enter year month day: "

cin>>year>> month>>day; //首先输入三个数： 年 月 日

while(year!= 0 && month!=0 && month!= 0)

{

…… //具体计算语句

//本次循环结束前再次输入三个数据（年月 日），为下次循环计算做准备

cout<< "please enter year month day: "

cin>>year>> month>>day;

}

**[**测试数据**]**

1、1 年1 月1 日是星期一

2、2 年1 月1 日是星期二

3、用上机日期 2018年4月24号是星期2

4、2000 年1 月1 日是星期六

源代码：

/\*

\* 1-6.cpp

\*

\* Created on: 2018年4月24日

\* Author: asus

\*/

#include <iostream>

using namespace std;

//公元1年1月1日是星期一,求输入的年月日是星期几

void mdays(int year); //获取当前年份的每月天数

int m\_day[12];

int main() {

int year, month, day, days = 0;

cout << "请输入年:";

cin >> year;

mdays(year);

do {

cout << "请输入月:";

cin >> month;

} while (month <= 0 || month > 12);

do {

cout << "请输入日:";

cin >> day;

} while (day <= 0 || day > m\_day[month]);

for (int i = 1; i < year; i++) { //获取1~year-1年的天数

mdays(i);

for (int j = 0; j < 12; j++)

days += m\_day[j];

}

for (int i = 1; i < month; i++) //获取year年前month个月的天数

days += m\_day[i - 1];

days += day;

if (days % 7 == 0)

cout << "今天是星期日";

else

cout << "今天是星期" << days % 7;

}

void mdays(int year) {

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) { //算每个月多少天

int day[12] = { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

for (int i = 0; i < 12; i++)

m\_day[i] = day[i];

} else {

int day[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

for (int i = 0; i < 12; i++)

m\_day[i] = day[i];

}

}

**[**思考与扩展**]**

1、用 switch 语句实现程序。比较if 和switch 语句的区别。

2、对输入日期和理性检验，例如月份不能大于12，非闰年不得有2 月29 日，小月不得有4 月31 日

## 3.2 实验2 函数与数组

### 题目1 分析程序运行结果

输入下列程序，运行它，分析得到的结果。



要求：

1、阅读分析程序，给程序加上必要的注释

2、分析何以得出运行结果的原因

3、区分程序中那些是全局变量、局部变量、局部静态变量？这些变量的差别是什么？作用域和撤销时机不一样 全局变量作用在全局但可被局部变量屏蔽 局部变量作用在该局部，执行完后即被撤销 局部静态变量和全局变量作用范围一样，只不过所以程序共享一份只在进程执行完后才被撤销

4、说明函数调用时实参和形参对应关系及值传递方式？值复制一份来传递

**[**测试数据**]**

改变main( )函数中a 的值令a = 10，改变func()函数中b 的值令static int b=10;程序运行结果？

结果：

local a=2

local b=11

parameter x=11

local a=10

local b=13

global n=1

local a=2

local b=12

parameter x=12

local a=11

local b=14

global n=2

local a=11

local b=14

global n=2

local a=2

local b=13

parameter x=11

**[**思考与扩展**]**

1将函数 func（）中的变量static int b=10;改为 int b，程序运行结果有什么变化。有变化

local a=2

local b=11

parameter x=11

local a=10

local b=13

global n=1

local a=2

local b=11

parameter x=12

local a=11

local b=13

global n=2

local a=11

local b=13

global n=2

local a=2

local b=11

parameter x=11

2、如果将 main（）函数中定义变量的语句 int a，b；移到程序第三行？程序运行结果有没有变化，为什么。

没有变化，原先是一般变量，变为全局变量后对本程序执行结果没什么影响

### 题目2 函数的递归调用和多文件结构

编写一个函数，求从 n 个不同的数中取r 个数的所有选择的个数。其个数值为：

其中： n! = n \* (n-1) \* (n-2) \* ... \* 1。

源代码：

//主函数main

/\*

\* 2-2.cpp

\* Created on: 2018年4月29日

\* Author: asus

\*/

#include <iostream>

using namespace std;

#include "fun.h"

int main(){

for(int i=0;i<16;i++)

cout<<i<<"!= "<<fun(i)<<endl;

cout<<func();

}

/\*

\* fun.h

\* Created on: 2018年4月29日

\* Author: asus

\*/

#ifndef LPLIB\_H\_

#define LPLIB\_H\_

Unsigned long long fun(int);

#endif /\* LPLIB\_H\_ \*/

/\*

\* fun.cpp

\*

\* Created on: 2018年4月24日

\* Author: asus

\*/

unsigned long long fun(int i){

if(i==0)

return 1;

else

return i\*fun(i-1);

}

**[**思考与扩展**]**

1、设置断点观察函数的递归调用和嵌套调用的执行流程。

2、递归函数的书写要点是什么？递归条件，递归出口

3、在递归版本中，当n 大到什么值时系统将发生栈溢出报错？你觉得递归和非递归函

数哪种好些？，非递归好

4、多文件结构中头文件的作用是什么？声明

5、将程序划分为多个文件有什么好处？逻辑性好便于管理和维护

### 题目3 排序查找

用选择法对 15 个学生的成绩按从大到小的顺序排序，15 个学生的成绩整数用cin 输入，如果输入的成绩不在[0，100]之间，则提示“输入错误”。排序完成后，输入一个成绩，要求用折半查找法找出该成绩是该组中第几个元素的值（即第几名）。如果该成绩不在数组中，则输出“无此成绩”。

要求：

1、把排序算法写成函数形式，在主函数中输入15 个数据，然后调用排序函数排序。

2、在排序过程中尽量减少数据的交换和移动。

3、把查找算法写成函数形式，在主函数中输入1 个数据，然后调用查找函数排序。

**[**测试数据**]**

自己选择15 数据输入。查找某一成绩时，分别选用在15 个中的成绩和没有在15 成绩

中的数据测试。

源代码：

/\*

\* 2-3.cpp

\*

\* Created on: 2018年4月24日

\* Author: asus

\*/

#include <iostream>

using namespace std;

void sorts(double []);

int bin(double a[],double d);

int main()

{

int r=0;

double d=24,a[15];//={1,3,2,4,19,24,54,67,89,34,25,46,45,68,87};

for(int i=0;i<15;i++){

cin>>a[i];

if(a[i]>100||a[i]<0){

cout<<"输入错误"<<endl;

i--;

}

}

sorts(a);

for(int i=0;i<15;i++)

cout<<a[i]<<" ";

cout<<endl;

r=bin(a,d);

if(r==0)

cout<<"无此成绩"<<endl;

else

cout<<"排名为："<<r<<endl;

}

void sorts(double a[]){//选择排序

for(int i=0;i<15;i++)

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

if(a[i]<a[j]){

a[i]=a[i]+a[j];

a[j]=a[i]-a[j];

a[i]=a[i]-a[j];

}

}

int bin(double a[],double n){//折半查找

int l=0,h=14,m=(l+h)/2;

while(l<=h){

if(n==a[m])

return m+1;

else if(n>a[m])

l=m+1;

else

h=m-1;

m=(l+h)/2;

}

return -1;

}

**[**思考与扩展**]**

1、将选择法排序与起泡法比较，分析它们的特点和效率。选择排序是每次找一个最优的 冒泡是每次只要是比当前元素优就交换

2、如果采用顺序查找的办法，即从第一个元素开始对比一直到找到与之相等的元素，和折半查找法相比，那个效率高？折半查找

3、数组名作为函数参数，形参和实参的关系？传的是数组的首地址对形参操作会改变实参

### 题目4 矩阵相乘

将两个给定的距阵(3＊3)相乘得到另一个距阵并将其打印出来。提示：

要求：

1、从键盘输入两个矩阵，并输出结果矩阵。矩阵的输入输出要求行列对齐。

2、把矩阵相承写成函数形式，在主函数中完成矩阵的输入输出。

**[**测试数据**]**

自己选择一组数据输入。

源代码：

/\*

\* 2-4.cpp

\*

\* Created on: 2018年4月24日

\* Author: asus

\*/

#include <iostream>

using namespace std;

int main(){

int r1,c1,r2,c2;

do{

cout<<"请输入矩阵一的行数和列数：";

cin>>r1>>c1;

cout<<"请输入矩阵二的行数和列数：";

cin>>r2>>c2;

}while(c1!=r2);

int a[r1][c1],b[r2][c2],c[r1][c2]={0};

for(int i=0;i<r1;i++){//初始化矩阵一

for(int j=0;j<c1;j++)

cin>>a[i][j];

}

for(int i=0;i<r2;i++){//初始化矩阵二

for(int j=0;j<c2;j++)

cin>>b[i][j];

}

cout<<endl;

for(int i=0;i<r1;i++){//计算

for(int j=0;j<c2;j++)

for(int k=0;k<c1;k++)

c[i][j]+=a[i][k]\*b[k][j];

}

for(int i=0;i<r1;i++){//输出计算结果

for(int j=0;j<c2;j++)

cout<<c[i][j]<<" ";

cout<<endl;

}

}

**[**思考与扩展**]**

1、如果是一个4\*3 矩阵和一个3\*2 矩阵相乘，程序该如何改写？已改

2、多维数组名作为函数参数，形参和实参的关系？传的是数组首地址

### 题目5 字符串连接

编一程序，将两个字符串连接起来。结果取代第一个字符串。要求

1、用字符数组，不用strcat 函数；

2、用C 标准中的strcat 函数；

3、用string 方法定义字符串变量；

4、对这三种方法进行比较。

源代码：

/\*

\* 2-5.cpp

\*

\* Created on: 2018年4月24日

\* Author: asus

\*/

#include <iostream>

#include <cstring>

#include <string>

using namespace std;

#define max 100

void f1();

void f2();

void f3();

int main() {

f1();

f2();

f3();

}

void f1() { //用数组串起来

char str[max], dst[max];

cin >> str >> dst;

int i = 0, j = 0;

while (str[i] != '\0') {

i++;

}

while (dst[j] != '\0') {

str[i++] = dst[j++];

}

cout << str << endl;

}

void f2() { //strcat函数串起来

char str[max], dst[max];

cin >> str >> dst;

strcat(str, dst);

cout << str << endl;

}

void f3() { //对string类型直接用'+'连接

string str, dst;

cin >> str >> dst;

str = str + dst; //str.append(dst);

cout << str << endl;

}

**[**测试数据**]**

自己选择一组数据输入。

**[**思考与扩展**]**

字符数组和字符串数组有什么联系或区别？字符串数组是由字符串为元素构建的数组，每个元素又是一个数组，字符串数组相当于一个二维数组

## 3.3 实验三 指针与引用

### 题目1 程序分析

输入下列程序，运行它，分析得到的结果。





要求：

1、给本题代码加上适当的注释,编译调试，分析说明程序运行结果。

2、如果将swap 函数改为如下形式，程序运行结果是什么？分析说明。结果不变 temp也为指针变量，与p1，p2类型相同

void swappionter(int \*p1,int \*p2)

{

int \* temp;

temp = p1;

p1 = p2;

p2 = temp;

}

3、如果将swap 函数改为如下形式，主函数中函数调用改为：swap( iv1, iv2 );程序运行结果是什么？分析说明。 结果不变，传的参数是引用

void swap(int &p1,int &p2)

{

int temp;

temp = p1;

p1 = p2;

p2 = temp;

}

结果：

---------------------

iv1=888

\*pointer\_1=888

&iv1=0x78fef4

pointer\_1=0x78fef4

---------------------

iv2=666

\*pointer\_2=666

&iv2=0x78fef0

pointer\_2=0x78fef0

---------------------

iv1=666

iv2=888

\*pointer\_1=666

\*pointer\_2=888

pointer\_1=0x78fef4

pointer\_2=0x78fef0

**[**测试数据**]**

自己修改程序中部分代码，调试运行。

**[**思考与扩展**]**

1 归纳说明普通变量作形参、指针作形参、引用作函数形参得区别。

普通变量作形参是值传递 指针作形参传的是指针值，对形参操作相当于对实参操作 引用作形参是传的变量的引用，与普通变量类似，只不过实参不再复制一份，兼容了值传递和指针传递的好处

### 题目2 班级课程成绩计算

一个班有 5 名学生，每个学生修了五门课，1）求每个学生的平均成绩，并输出每个学生的学号，每门课程的成绩及平均值。2）求某门课程的平均分；

实现要求：

1、分别编写2 个函数实现以上2 个要求。

2、第1 个函数用数组名作参数。第2 个函数用指针用参数，并在函数体内用指针对数组操作。

源代码：

/\*

\* 3-2.cpp

\*

\* Created on: 2018年4月24日

\* Author: asus

\*/

#include <iostream>

using namespace std;

void average(int a[][5],int s[]);

void aversco(int a[][5],int s[]);

int main(){

int num[5]={0};

int sco[5][5]={10,20,30,40,50,10,20,30,40,50,10,20,30,40,50,10,20,30,40,50,10,20,30,40,50};

int asco[5] = { 0 };

average(sco,asco);

aversco(sco,num);

return 0;

}

void average(int a[5][5],int s[5]){

for(int i=0;i<5;i++){

for(int j=0;j<5;j++)

s[i]+=a[i][j];

s[i]/=5;

cout<<s[i]<<endl;

}

}

void aversco(int a[5][5],int s[5]){

for(int i=0;i<5;i++){

for(int j=0;j<5;j++)

s[i]+=a[j][i];

s[i]/=5;

cout<<s[i]<<endl;

}

}

**[**测试数据**]**

自己选择一组数据输入。

**[**思考与扩展**]**

1、二维数组作为函数形参int avercourse( int（\*psco）[5], n )中为什么要指定第二维的

大小？规定

2、在函数体中用指针操作二维数组譬如：\*（\*（psco +i）+j）就代表数组第i 行j 列

的元素即psco[i][j]。

### 题目3 排序查找

用选择法对 15 个整数按从大到小的顺序排序。15 个整数用cin 输入。排序完成后输入一个数，要求用折半查找法找出该数是数组中第几个元素的值。如果该数不在数组中，则输出“无此数”。

要求：

1、把排序算法写成函数形式，用指针变量作为排序函数的参数，在排序算法中用指针法访问数组元素。

2、把查找算法写成函数形式，用指针变量作为查找函数的参数，在查找算法中用指针法访问数组元素。

源代码：

/\*

\* 3-3.cpp

\*

\* Created on: 2018年4月24日

\* Author: asus

\*/

#include <iostream>

using namespace std;

void sorts(double []);

int fin(double a[],double d);

int main()

{

int r=0;

double d=24,a[15];//={1,3,2,4,19,24,40,67,89,34,25,46,45,68,87};

for(int i=0;i<15;i++){

cin>>a[i];

if(a[i]>100||a[i]<0){

cout<<"输入错误"<<endl;

i--;

}

}

sorts(a);

for(int i=0;i<15;i++)

cout<<a[i]<<" ";

cout<<endl;

r=fin(a,d);

if(r==0)

cout<<"无此成绩"<<endl;

else

cout<<"排名为："<<r<<endl;

}

void sorts(double a[]){

int k=0;

for(int i=0;i<15;i++){

k=i;

for(int j=i+1;j<15;j++){

if(a[j]<a[k])

k=j;

}

if(k>i){

a[i]=a[k]+a[i];

a[k]=a[i]-a[k];

a[i]=a[i]-a[k];

}

}

}

int fin(double a[],double n){

int l=0,h=14,m=(l+h)/2;

while(l<=h){

if(n==a[m])

return m+1;

else if(n>a[m])

l=m+1;

else

h=m-1;

m=(l+h)/2;

}

return -1;

}

**[**测试数据**]**

自己选择一组数据输入。

**[**思考与扩展**]**

1、在调用函数时实参为什么只需给出数组名？数组名是数组首地址通过首地址即可访问数组全部元素

2、指针函数和函数指针有什么区别？

指针函数是指带指针的函数，本质是一个函数，函数返回类型是某一类型的指针

函数指针是指向函数的指针变量，本质是一个指针变量

## 3.4 实验四 结构体与链表

### 题目1 日期计算

定义一个结构体变量（包括年、月、日），编程序，要求输入年月日，程序能计算并输出该日在本年中第几天。注意闰年问题。

源代码：

/\*

\* 4-1.cpp

\*

\* Created on: 2018年4月24日

\* Author: asus

\*/

#include <iostream>

using namespace std;

struct date{

int year, month, day;

}date;

int m\_day[12];//每月天数

//公元1年1月1日是星期一,求输入的年月日是星期几

void mdays(int year); //获取当前年份的每月天数

int main() {

int days = 0;

cout << "请输入年:";

cin >> date.year;

mdays(date.year);

do {

cout << "请输入月:";

cin >> date.month;

} while (date.month <= 0 || date.month > 12);

do {

cout << "请输入日:";

cin >> date.day;

} while (date.day <= 0 || date.day > m\_day[date.month]);

for (int i = 1; i < date.year; i++) { //获取1~year-1年的天数

mdays(i);

for (int j = 0; j < 12; j++)

days += m\_day[j];

}

for (int i = 1; i < date.month; i++) //获取year年前month个月的天数

days += m\_day[i - 1];

days += date.day;

if (days % 7 == 0)

cout << "今天是星期日";

else

cout << "今天是星期" << days % 7;

}

void mdays(int year) {

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) { //算每个月多少天

int day[12] = { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

for (int i = 0; i < 12; i++)

m\_day[i] = day[i];

} else {

int day[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

for (int i = 0; i < 12; i++)

m\_day[i] = day[i];

}

}

**[**测试数据**]**

1、 1 年1 月1 日是星期一

2、 2 年1 月1 日是星期二

3、用上机日期测试

5、 2000 年1 月1 日是星期六。

**[**思考与扩展**]**

1、修改程序，在程序中用指针方法访问结构体变量的成员。

2、结构体变量占用的存储单元字节数是多少，能不能用sizeof（）求结构体变量字节

数？

能，不过不能直接使用，要针对每一个元素用sizeof才能求结构体变量字节

### 题目2 学生成绩管理

有 5 个学生，每个学生的数据包括学号、班级、姓名、三门课成绩。从键盘输入5 个学生数据，要求打印出每个学生三门课的平均成绩，以及平均分最高分学生数据（包括学号、班级、姓名、三门课成绩，平均分）。

要求：

1、定义学生结构体。

2、用一个函数实现5 个学生数据的输入，用另一个函数负责求每个学生三门课程的平均成绩，再用一个函数求出平均分最高的学生并输出该学生的数据。要求平均分和平均分最高的学生数据都在主函数种输出。

3、用多文件程序结构实现。

源代码：

/\*

\* 4-2.cpp

\*

\* Created on: 2018年4月24日

\* Author: asus

\*/

#include <iostream>

using namespace std;

struct student{

int num;//学号

char name[20];//名字

char classname[20];//班级

float score[3];//三门课成绩

float aver\_score;//平均成绩

};

void average(student &stu);

int max(student stu[]);

int main(){

student stu[5];

cout<<"请输入num,name,class,score1,score2,score3"<<endl;

for (int i = 0; i < 5; i++) {

cin>>stu[i].num>>stu[i].name>>stu[i].classname>>stu[i].score[0]>>stu[i].score[1]>>stu[i].score[2];

}

for(int i=0;i<5;i++){

average(stu[i]);

cout<<"student"<<i<<"'s average score:"<<stu[i].aver\_score<<endl;

}

cout<<"平均分最高的学生为："<<max(stu)<<endl;

}

void average(student &stu){

stu.aver\_score=0;

for(int i=0;i<3;i++){

stu.aver\_score+=stu.score[i];

}

stu.aver\_score/=3;

}

int max(student stu[]){

int k=0;

for(int i=1;i<5;i++){

if(stu[k].aver\_score<stu[i].aver\_score)

k=i;

}

return k;

}

**[**测试数据**]**

自己设计一组测试数据

**[**思考与扩展**]**

1、如果这样定义结构体

typedef struct Student

{

int num； //学号

char name[20]； //姓名注意字节长度

char classname[20]； //班级

float score[3]； //三门课程成绩

float aver\_score; //平均分

} STU，\* pSTU；

说明如下三条语句的含义：

student stu1; 构造一个结构体变量

STU stu2; 构造一个结构体变量

pSTU pstu; 构造一个结构体指针变量

2、结构体中姓名和班级字段如果定义为 string 类型，程序该如何改写？

差不太多，string类型操作还更简单

### 题目3 结构体链表

设计一个单向链表。从标准输入读取若干整数，建立链表，每次读入的数放入链表结尾。

每个节点包含两个值，一个是真正存放的整数值，另一个为指向链表中下一个节点的指针。链表中最后一个节点不指向任何节点，所以指针为空（NULL）。表头指针和表尾指针分别指向链表的头节点和尾节点。

源代码：

/\*

\* 4-3.cpp

\*

\* Created on: 2018年4月25日

\* Author: asus

\*/

#include <iostream>

using namespace std;

struct node{

int elem;

node \*next;

};

int main(){

int i;

cout<<"Please input integers to build the link(0 TO END):";

cin>>i;

if(i==0)

return 0;

else{

node \*head=new node;

node \*s=head;

s->elem=i;

s->next=NULL;

do{

cin>>i;

if(i==0)

break;

else{

node \*p=new node;

p->elem=i;

p->next=NULL;

s->next=p;

s=p;

}

}while(i);

s=head;

cout<<"output:";

while(s){

cout<<s->elem<<" ";

s=s->next;

}

}

}

**[**测试数据**]**

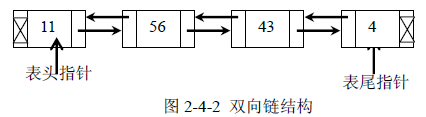
程序输出：Please input integers to build the link(0 TO END):

用户输入：3 4 5 6 7 8 9 0

程序输出：Link elements:3 4 5 6 7 8 9

**[**思考问题**]**

如果是双向链表，程序要做那些改动？双向链表的示意图如下：



双向链表的话，插入一个新的结点是时候要操作4个指针，操作更为复杂，而且有顺序

### 题目4 通信录

设一个通信录由以下几项数据信息构成：

数据项 类型

姓名 字符串

地址 字符串

邮政编码 字符串

电话号码 字符串

电子邮件 字符串

实现一个通讯录程序。实现功能：存取、显示、删除记录、查询记录等功能。

源代码：功能还未完全实现

/\*

\* 4-4.cpp

\*

\* Created on: 2018年4月25日

\* Author: asus

\*

\*

\* 设一个通信录由以下几项数据信息构成：

\* 数据项 类型

\* 姓名 字符串

\* 地址 字符串

\* 邮政编码 字符串

\* 电话号码 字符串

\* 电子邮件 字符串

\* 实现一个通讯录程序。实现功能：存取、显示、删除记录、查询记录等功能

\*

\*/

#include <iostream>

using namespace std;

struct contacts{

char name[20];

char address[20];

char phone[20];

char post[6];

char email[20];

contacts \*next;

};

int count=0;

contacts \*head=NULL,\*s=head;

void add();

void display();

void del(int);

void search(contacts &con);

int main(){

add();

add();

display();

del(1);

display();

return 0;

}

void add(){

count++;

contacts \*con=new contacts;

con->next=NULL;

if(head==NULL){

head=con;

}

else{

s->next=con;

}

cout<<"Please input name,address,phone,post,email"<<endl;

cin>>con->name>>con->address>>con->phone>>con->post>>con->email;

s=con;

}

void display(){

s=head;

cout<<"\noutput name,address,phone,post,email"<<endl;

while(s){

cout<<s->name<<" "<<s->address<<" "<<s->phone<<" "<<s->post<<" "<<s->email<<endl;

s=s->next;

}

}

void del(int n){

if(n<=0||n>count){

cout<<"";

return;

}

s=head;

contacts \*p=s;

while(n--){

p=s;

s=s->next;

}

p=s->next;

//cout<<"\noutput name,address,phone,post,email"<<endl;

//cout<<s->name<<" "<<s->address<<" "<<s->phone<<" "<<s->post<<" "<<s->email<<endl;

}

void search(contacts &con){

}

### 题目5 计算器

实现一个简单的计算器模拟 windows 操作系统中的计算器，系统界面不做强制要求。

实现以下功能：

（1）、能进行实数的加减乘除运算；

（2）、能进行高次方、高次开方运算；

（3）、能进行简单三角运算；

源代码：

/\*

\* 4-5.cpp

\*

\* Created on: 2018年4月25日

\* Author: asus

\*/

#include <iostream>

#include <cmath>

using namespace std;

#define PI 3.1415927

//double addf(double,double);

//double subf(double,double);

//double mulf(double,double);

//double divf(double,double);

double powf(double,int);

typedef double f(double,double);

f addf,subf,mulf,divf;

f \*ff;

int main(){

cout<<sin(90/180.0\*PI)<<endl;

ff=addf;

cout<<ff(2.0,3)<<endl;

ff=subf;

cout<<ff(2.0,3)<<endl;

ff=mulf;

cout<<ff(2.0,3)<<endl;

ff=divf;

cout<<ff(2.0,3)<<endl;

return 0;

}

double addf(double a,double b){

return a+b;

}

double subf(double a,double b){

return a-b;

}

double mulf(double a,double b){

return a\*b;

}

double divf(double a,double b){

return a/b;

}

double powf(double a,int b){

double c=1;

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

c\*=a;

return c;

}

## 3.5 实验五 类和对象

### 题目1 图形类设计

设计三个图形类：Circle（圆）、Rectangle（矩形）、Triangle（三角形）；

1、Cirlce 类基本信息：圆心坐标、半径；

Rectangle 类基本信息：长、宽；

Triangle 类基本信息：三个顶点坐标；

其中：成员变量为 private 属性，成员函数为public 属性；

2、每个图形类有多个构造函数：缺省构造函数、带参数的构造函数；

3、每个图形类有计算图形的面积GetArea()，显示图形的基本信息函数Show()，修改基本信息的函数Set(形参)。以Circle 类为例：通过GetArea()计算圆的面积，Show()函数中显示圆心坐标、直径、周长、面积等基本信息；Set(int x,int y, int r)函数可以修改圆心坐标和半径。

源代码：

/\*

\* 5-1.cpp

\*

\* Created on: 2018年5月3日

\* Author: asus

\*/

#include <iostream>

using namespace std;

#include "Shape.h"

int main(){

Circle c;

Rectangle r;

c.set(20,30,5);

c.show();

c.set(-20,40,30);

c.show();

c.set(40,-10,-5);

c.show();

r.set(20,50);

r.show();

r.set(40,5);

r.show();

r.set(-9,10);

r.show();

r.set(43,-8);

r.show();

Triangle t(20,30,40,50,40,20);

//t.set();

t.show();

Triangle t1(10,10,50,10,35,10);

t1.show();

}

头文件：

/\*

\* Shape.h

\*

\* Created on: 2018年5月3日

\* Author: asus

\*/

#ifndef A405\_SHAPE\_H\_

#define A405\_SHAPE\_H\_

struct dot{

double x;

double y;

};

class Circle{

private:

dot d;

double radius;

public:

Circle();

Circle(double x,double y,double r);

void set(double x,double y,double r);

double getarea();

double getgirth();

void show();

};

class Rectangle{

public:

Rectangle();

Rectangle(double x,double y);

void set(double x,double y);

double getgirth();

double getarea();

void show();

private:

double m\_x,m\_y;

};

class Triangle{

public:

Triangle();

Triangle(double x1,double y1,double x2,double y2,double x3,double y3);

//void set(double x1,double y1,double x2,double y2,double x3,double y3){

void set();

bool f(dot a,dot b,dot c);

double edge(dot a,dot b);

double getgirth();

void show();

private:

dot a,b,c;

};

#endif /\* A405\_SHAPE\_H\_ \*/

Shape.h

/\*

\* Shape.cpp

\*

\* Created on: 2018年5月3日

\* Author: asus

\*/

#include <iostream>

#include <cmath>

using namespace std;

#include "Shape.h"

const double PI = 3.14159;

Circle::Circle() { //缺省的构造方法

d.x = 0;

d.y = 0;

radius = 1;

}

Circle::Circle(double x, double y, double r) { //带参的构造方法

d.x = x;

d.y = y;

while (r < 0) {

cout << "半径r=" << r << "半径小于零,不是一个圆，请重新输入半径:";

cin >> r;

}

radius = r;

}

void Circle::set(double x, double y, double r) {

d.x = x;

d.y = y;

while (r < 0) {

cout << "半径r=" << r << "半径小于零,不是一个圆，请重新输入半径:";

cin >> r;

}

radius = r;

}

double Circle::getarea() {

return PI \* radius \* radius;

}

double Circle::getgirth() {

return 2 \* PI \* radius;

}

void Circle::show() {

if (radius == 0)

cout << "此时，图形为一个点，坐标是：(" << d.x << "," << d.y << ")" << endl << endl;

else {

cout << "圆心为：(" << d.x << "," << d.y << ") , 半径为：" << radius << endl;

cout << "周长为：" << getgirth() << "面积为：" << getarea() << endl << endl;

}

}

Rectangle::Rectangle() { //缺省的构造方法

m\_x = 1;

m\_y = 1;

}

Rectangle::Rectangle(double x, double y) { //带参的构造方法

while (x <= 0) {

cout << "长度x=" << x << "长度不能小于0，请重新输入长度:";

cin >> x;

}

while (x <= 0) {

cout << "宽度y=" << y << "宽度不能小于0，请重新输入宽度:";

cin >> x;

}

m\_x = x;

m\_y = y;

}

void Rectangle::set(double x, double y) {

while (x <= 0) {

cout << "长度x=" << x << "长度不能小于0，请重新输入长度:";

cin >> x;

}

while (x <= 0) {

cout << "宽度y=" << y << "宽度不能小于0，请重新输入宽度:";

cin >> x;

}

m\_x = x;

m\_y = y;

}

double Rectangle::getgirth() {

return 2 \* (m\_x + m\_y);

}

double Rectangle::getarea() {

return m\_x \* m\_y;

}

void Rectangle::show() {

cout << "矩形的长为：" << m\_x << "宽为：" << m\_y << endl;

cout << "周长为：" << getgirth() << "面积为：" << getarea() << endl << endl;

}

Triangle::Triangle() {

a.x = 0;

a.y = 1;

b.x = 0;

b.y = 0;

c.x = 1;

c.y = 0;

}

Triangle::Triangle(double x1, double y1, double x2, double y2, double x3,

double y3) {

a.x = x1;

a.y = y1;

b.x = x2;

b.y = y2;

c.x = x3;

c.y = y3;

if (f(a, b, c) == false) {

cout << "该数据无法构成三角形" << endl;

set();

}

}

//void set(double x1,double y1,double x2,double y2,double x3,double y3){

void Triangle::set() {

double x1, x2, x3, y1, y2, y3;

cout << "请输入三角形的三个顶点:";

cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3;

a.x = x1;

a.y = y1;

b.x = x2;

b.y = y2;

c.x = x3;

c.y = y3;

if (f(a, b, c) == false) {

cout << "该数据无法构成三角形" << endl;

set();

}

}

bool Triangle::f(dot a, dot b, dot c) {

if (edge(a, b) == 0 || edge(a, c) == 0 || edge(a, c) == 0)

return false;

if (edge(a, b) + edge(a, c) <= edge(b, c))

return false;

if (edge(a, b) + edge(b, c) <= edge(a, c))

return false;

if (edge(c, b) + edge(a, c) <= edge(b, a))

return false;

return true;

}

double Triangle::edge(dot a, dot b) {

return sqrt((a.x - b.x) \* (a.x - b.x) + (a.y - b.y) \* (a.y - b.y));

}

double Triangle::getgirth() {

return edge(a, b) + edge(a, c) + edge(b, c);

}

void Triangle::show() {

cout << "三角形的三个顶点为：(" << a.x << "," << a.y << "),(" << b.x << "," << b.y

<< "),(" << c.x << "," << c.y << ")" << endl;

cout << "周长为：" << getgirth() << endl << endl;

}

**[**测试数据**]**

1、Circle 类测试数据：

圆心：（20，30） 半径：5

（-20，40） 30

（40，-10） -5

2、Rectangle 类测试数据：

顶点坐标：20，50

40，5

-9，10

43，-8

3、Triangle 类测试数据：

顶点：（20，30）、（40，50）、（40，20）

（10，10）、（50，10）、（35，10）

结果：

圆心为：(20,30) , 半径为：5

周长为：31.4159面积为：78.5397

圆心为：(-20,40) , 半径为：30

周长为：188.495面积为：2827.43

半径r=-5半径小于零,不是一个圆，请重新输入半径:3

圆心为：(40,-10) , 半径为：3

周长为：18.8495面积为：28.2743

矩形的长为：20宽为：50

周长为：140面积为：1000

矩形的长为：40宽为：5

周长为：90面积为：200

长度x=-9长度不能小于0，请重新输入长度:2

矩形的长为：2宽为：10

周长为：24面积为：20

矩形的长为：43宽为：-8

周长为：70面积为：-344

三角形的三个顶点为：(20,30),(40,50),(40,20)

周长为：80.645

该数据无法构成三角形

请输入三角形的三个顶点:3 4 5 7 1 4

三角形的三个顶点为：(3,4),(5,7),(1,4)

周长为：10.6056

### 题目2 学生类设计

设计一个 Studnet(学生)类

1、基本信息：学号、姓名、性别、出生日期、年级、班级、院系、专业；

其中：基本信息为 private 属性，成员函数为public 属性；

2、Student 类有多个构造函数：缺省构造函数、带参数的构造函数、带默认参数的构造

函数

3、基本功能：

1）使用对象数组保存学生对象；

2）可以从键盘输入学生的基本信息；

3）SetInfo(形参表)函数可以修改学生的基本信息函数，例如：班级、专业等；

4）Show()函数显示学生信息；

源代码：

主函数

/\*

\* 5-2.cpp

\*

\* Created on: 2018年5月3日

\* Author: asus

\*/

#include <iostream>

using namespace std;

#include "student.h"

int main(){

information info;

birthday date;

student stu;

stu.SetInfo(info,date);//调用set方法

stu.Show();

}

声明文件student.h

/\*

\* student.h

\*

\* Created on: 2018年5月3日

\* Author: asus

\*/

#ifndef A405\_STUDENT\_H\_

#define A405\_STUDENT\_H\_

struct information{

string num;//学号

string name;//姓名

string sex;//性别

string college;

string grade;//年级

string mclass;//班级

string major;//专业

};

struct birthday{

int year,month,day;

};

class student{

public:

student();

student(information sinfo,birthday sdate);

void SetInfo(information sinfo,birthday sdate);

void Show();

private:

struct information info;

struct birthday date;

};

#endif /\* A405\_STUDENT\_H\_ \*/

student.cpp

/\*

\* student.cpp

\*

\* Created on: 2018年5月3日

\* Author: asus

\*/

#include <iostream>

using namespace std;

#include "student.h"

student::student(){

info.name="李攀";

info.num="320160939611";

info.sex="M";

info.college="兰州大学信息科学与工程";

info.grade="2016";

info.major="计算机科学与技术";

info.mclass="计算机一班";

date.year=1998;

date.month=12;

date.day=17;

}

student::student(information sinfo,birthday sdate){

info=sinfo;

date=sdate;

}

void student::SetInfo(information sinfo,birthday sdate){

cout<<"请输入姓名：";

cin>>sinfo.name;

info=sinfo;

date=sdate;

}

void student::Show(){

cout<<"name,num,sex,college,grade,major,mcalss,year,month,day"<<endl;

cout<<info.college<<"学院"<<info.grade<<"级"<<info.major<<"专业"<<info.mclass<<endl;

cout<<"姓名:"<<info.name<<" "<<" 学号:"<<info.num<<" 性别:"<<info.sex;

cout<<" 出生日期:"<<date.year<<"年"<<date.month<<"月"<<date.day<<"日"<<endl;

}

**[**测试数据**]**

1、学号：2005001 姓名：zhangesan 出生日期：85 年11 月　年级：05 级　班级：jy01　 院系：computer 专业：application

2、学号：2005002 姓名：lihuan 出生日期：84 年11 月　年级：05级　班级：tx02 院系：math 专业：application

3、学号：2005003 姓名：wang 出生日期：85 年7 月　　年级：05 级 班级：jy01 院系：computer 专业：software

4、学号：2005004 姓名：zhan 出生日期：85 年11 月　　年级：05 级 班级：jy02 院系：SCI

5、学号：2005005 姓名：song 出生日期：85 年11 月　　年级：05级　班级：tx02 院系：math 专业：application

**[**思考与扩展**]**

采用什么方式可以让学号自动按输入顺序生成？ 计数器计数方式

### 题目3 人员管理

设计某小型公司的 employee（人员）类

1、类

1）employee 类：

基本信息：编号、姓名、性别、出生日期、职位等；

出生日期使用自定义的 Date（日期）类；

其中：基本信息为 private 属性，成员函数为public 属性；

多个构造函数：缺省构造函数、带参数的构造函数、带默认参数的构造函数；可以从外部访问类成员的友员函数；

2）Date 类：

成员变量：年、月、日

成员函数：SetYear(int year)、SetMonth(int month)、SetDay(int day)

GetYear()、GetMonth()、GetDay()

2、基本功能：

1）职工信息的录入；

2）职工信息的显示；

3）用对象数组保存已输入的职工对象；

4）可以修改人员的基本信息，如：姓名、职位等；

5）可以通过编号或姓名进行人员查询；

源代码：

主函数：

/\*

\* 5-3.cpp

\*

\* Created on: 2018年5月4日

\* Author: asus

\*/

#include <iostream>

using namespace std;

#include "employee.h"

int main(){

//employee e("lipan","12332","M","studentss");

//e.display();

employee ee[10];

ee[0].setinfo("li","111","f","普通",1998,1,2);

ee[0].display();

return 0;

}

头文件

/\*

\* employee.h

\*

\* Created on: 2018年5月4日

\* Author: asus

\*/

#ifndef A405\_EMPLOYEE\_H\_

#define A405\_EMPLOYEE\_H\_

class date{

private:

int year, month, day;

public:

date(int y=2018,int m=5,int d=4);

void setYear(int year);

void setMonth(int month);

void setDay(int day);

int getYear();

int getMonth();

int getDay();

};

class employee{

private:

string name;

string num;

string sex;

string job;

date dd;

public:

employee();

employee(string name,string num,string sex,string job,int year,int month,int day);

//employee(string name,string num="10000",string sex="男",string job="经理",int year=1998,int month=12,int day=17);

void setinfo(string name,string num,string sex,string job,int year,int month,int day);

void display();

};

employee.cpp

/\*

\* employee.cpp

\*

\* Created on: 2018年5月4日

\* Author: asus

\*/

#include <iostream>

using namespace std;

#include "employee.h"

date::date(int y,int m,int d){

year=y;

month=m;

day=y;

}

void date::setYear(int y){

year=y;

}

void date::setMonth(int m){

month=m;

}

void date::setDay(int d){

day=d;

}

int date::getYear(){

return year;

}

int date::getMonth(){

return month;

}

int date::getDay(){

return day;

}

employee::employee(){

name="jack";

num="10001";

sex="男";

job="普通";

dd.setYear(2018);

dd.setMonth(5);

dd.setDay(4);

}

employee::employee(string na,string nu,string se,string jo,int y,int m,int d):dd(y,m,d){

name=na;

num=nu;

sex=se;

job=jo;

}

void employee::setinfo(string na,string nu,string se,string jo,int y,int m,int d){

name=na;

num=nu;

sex=se;

job=jo;

dd.setYear(y);

dd.setMonth(m);

dd.setDay(d);

}

void employee::display(){

cout<<"编号:"<<num<<" 姓名:"<<name<<" 性别:"<<sex<<" 出生日期:"

<<dd.getYear()<<"年"<<dd.getMonth()<<"月"<<dd.getDay()<<"日"

<<" 职位:"<<job<<endl;

}

#endif /\* A405\_EMPLOYEE\_H\_ \*/

**[**测试数据**]**

1、编号：10001 姓名：jack 出生日期：80 年11 月1 日　　职位：普通

2、编号：10002 姓名：andy 出生日期：75 年1 月9 日　　职位：经理

3、编号：10003 姓名：alex 出生日期：81 年4 月3 日　　职位：秘书

4、编号：10005 姓名：lili 出生日期：82 年10 月1 日　　职位：技师

## 3.6 实验六 继承与派生

### 题目1 图形类

设计一个图形基类：Shape，在此基础上派生Rectangle、Circle、Triangle 类。

1、Cirlce 类基本信息：圆心坐标、半径；

Rectangle 类基本信息：长、宽；

Triangle 类基本信息：三个顶点坐标；

其中：成员变量为 private 属性，成员函数为public 属性；

2、每个图形类有多个构造函数：缺省构造函数、带参数的构造函数；

3、每个图形类有计算图形的面积GetArea()，显示图形的基本信息函数Show()，修改基

本信息的函数Set(形参)。

源代码：

主函数：

/\*

\* 6-1.cpp

\*

\* Created on: 2018年5月17日

\* Author: asus

\*/

#include <iostream>

using namespace std;

#include "cshape.h"

int main(){

Circle c;

c.set(30, 45, 35);

c.show();

c.set(50, 89, 23);

c.show();

Rectangle r;

r.set(30, 53);

r.show();

r.set(28, 24);

r.show();

Triangle t(34, 45, 89, 45, 54, 67);

t.show();

Triangle t1(22, 34, 67, 43, 86, 64);

t1.show();

}

头文件：

/\*

\* cshape.h

\*

\* Created on: 2018年5月17日

\* Author: asus

\*/

#ifndef CSHAPE\_H\_

#define CSHAPE\_H\_

class cshape{

public:

double getarea();

void set();

void show();

};

class Circle:public cshape{

private:

double m\_x,m\_y,m\_r;

public:

Circle();

Circle(double x,double y,double r);

double getarea();

void set(double x,double y,double r);

void show();

};

class Rectangle:public cshape{

private:

double m\_x,m\_y;

public:

Rectangle();

Rectangle(double x,double y);

double getarea();

double getgirth();

void set(double x,double y);

void show();

};

class Triangle:public cshape{

private:

double m\_x1,m\_y1,m\_x2,m\_y2,m\_x3,m\_y3;

public:

Triangle();

Triangle(double x1,double y1,double x2,double y2,double x3,double y3);

double getarea();

double getgirth();

double edge(double x1,double y1,double x2,double y2);

bool f(double x1,double y1,double x2,double y2,double x3,double y3);

void input();

void show();

};

#endif /\* CSHAPE\_H\_ \*/

cshape.cpp

/\*

\* cshape.cpp

\*

\* Created on: 2018年5月17日

\* Author: asus

\*/

#include <iostream>

#include <cmath>

#include "cshape.h"

using namespace std;

const double PI = 3.14159;

Circle::Circle() { //缺省的构造方法

m\_x = 0;

m\_y = 0;

m\_r = 1;

}

Circle::Circle(double x, double y, double r) { //带参的构造方法

m\_x = x;

m\_y = y;

while (r < 0) {

cout << "半径r=" << r << "半径小于零,不是一个圆，请重新输入半径:";

cin >> r;

}

m\_r = r;

}

void Circle::set(double x, double y, double r) {

m\_x = x;

m\_y = y;

while (r < 0) {

cout << "半径r=" << r << "半径小于零,不是一个圆，请重新输入半径:";

cin >> r;

}

m\_r = r;

}

double Circle::getarea() {

return PI \* m\_r \* m\_r;

}

void Circle::show() {

if (m\_r == 0)

cout << "此时，图形为一个点，坐标是：(" << m\_x << "," << m\_y << ")" << endl << endl;

else {

cout << "圆心为：(" << m\_x << "," << m\_y << ") , 半径为：" << m\_r << endl;

cout << "面积为：" << getarea() << endl << endl;

}

}

Rectangle::Rectangle() { //缺省的构造方法

m\_x = 1;

m\_y = 1;

}

Rectangle::Rectangle(double x, double y) { //带参的构造方法

while (x <= 0) {

cout << "长度x=" << x << "长度不能小于0，请重新输入长度:";

cin >> x;

}

while (x <= 0) {

cout << "宽度y=" << y << "宽度不能小于0，请重新输入宽度:";

cin >> x;

}

m\_x = x;

m\_y = y;

}

void Rectangle::set(double x, double y) {

while (x <= 0) {

cout << "长度x=" << x << "长度不能小于0，请重新输入长度:";

cin >> x;

}

while (x <= 0) {

cout << "宽度y=" << y << "宽度不能小于0，请重新输入宽度:";

cin >> x;

}

m\_x = x;

m\_y = y;

}

double Rectangle::getgirth() {

return 2 \* (m\_x + m\_y);

}

double Rectangle::getarea() {

return m\_x \* m\_y;

}

void Rectangle::show() {

cout << "矩形的长为：" << m\_x << "宽为：" << m\_y << endl;

cout << "周长为：" << getgirth() << "面积为：" << getarea() << endl << endl;

}

Triangle::Triangle() {

m\_x1 = 0;

m\_y1 = 0;

m\_x2 = 1;

m\_y2 = 0;

m\_x3 = 0;

m\_y3 = 1;

}

Triangle::Triangle(double x1, double y1, double x2, double y2, double x3,

double y3) {

m\_x1 = x1;

m\_y1 = y1;

m\_x2 = x2;

m\_y2 = y2;

m\_x3 = x3;

m\_y3 = y3;

if (f(x1, y1, x2, y2, x3, y3) == false)

input();

}

void Triangle::input() {

double x1, x2, x3, y1, y2, y3;

do {

cout << "请输入三角形的三个顶点:";

cin >> x1 >> y1 >> x2 >> y2 >> x3 >> y3;

m\_x1 = x1;

m\_y1 = y1;

m\_x2 = x2;

m\_y2 = y2;

m\_x3 = x3;

m\_y3 = y3;

cout << "该数据无法构成三角形" << endl;

} while (f(x1, y1, x2, y2, x3, y3) == false);

}

bool Triangle::f(double x1, double y1, double x2, double y2, double x3,double y3) {

double a, b, c;

a = edge(x1, y1, x2, y2);

b = edge(x1, y1, x3, y3);

c = edge(x2, y2, x3, y3);

if (a == 0 || b == 0 || c == 0)

return false;

if (a + b <= c)

return false;

if (a + c <= b)

return false;

if (b + c <= a)

return false;

return true;

}

double Triangle::edge(double x1,double y1,double x2,double y2) {

return sqrt((x1 - x2) \* (x1 - x2) + (y1 - y2) \* (y1 - y2));

}

double Triangle::getgirth() {

return edge(m\_x1,m\_y1,m\_x2,m\_y2) + edge(m\_x1,m\_y1,m\_x3,m\_y3) + edge(m\_x3,m\_y3,m\_x2,m\_y2);

}

double Triangle::getarea(){

double p, a, b, c;

a = edge(m\_x1, m\_y1, m\_x2, m\_y2);

b = edge(m\_x1, m\_y1, m\_x3, m\_y3);

c = edge(m\_x2, m\_y2, m\_x3, m\_y3);

p = (a + b + c) / 2;

return sqrt(p \* (p - a) \* (p - b) \* (p - c));

}

void Triangle::show() {

cout << "三角形的三个顶点为：(" << m\_x1 << "," << m\_y1 << "),(" << m\_x2 << "," << m\_y2

<< "),(" << m\_x3 << "," << m\_y3 << ")" << endl;

cout << "周长为：" << getgirth() << "面积为：" << getarea() << endl << endl;

}

**[**测试数据**]**

Circle：

圆心（30，45） 半径：35

圆心（50，89） 半径：23

Rectangle：长：30 宽：53

长：28 宽：24

Triangle：

顶点：（34，45）、（89，45）、（54，67）

顶点：（22，34）、（67，43）、（86，64）

结果：

圆心为：(30,45) , 半径为：35

面积为：3848.45

圆心为：(50,89) , 半径为：23

面积为：1661.9

矩形的长为：30宽为：53

周长为：166面积为：1590

矩形的长为：28宽为：24

周长为：104面积为：672

三角形的三个顶点为：(34,45),(89,45),(54,67)

周长为：126.072面积为：605

三角形的三个顶点为：(22,34),(67,43),(86,64)

周长为：144.893面积为：387

### 题目2 教师干部类（未做）

设计两个基类：教师类（Teacher）、干部类（Cadre），采用多重继承派生出新类：

Teacher\_Cadre

1、两个基类都包含：姓名、年龄、性别、地址、电话等；

在 Teacher 类中包含：title（职称），在Cadre 类中包含：post（职务），在Teacher\_Cadre

中包含：wages（工资）；

2、两个基类都包含：display()函数，分别输出本类中的成员信息；

3、在两个基类中的姓名、年龄、性别、地址、电话等数据成员使用相同名字，在派生

类中引用这些数据成员时，采用指定作用域的方式；

4、在派生类Teacher\_Cadre 的成员函数Show（）中调用Teacher 类中的display（）函

数，输出姓名、年龄、性别、地址、电话；

5、使用对象数组保存输入的对象；

**[**实验提示**]**

**[**测试数据**]**

1、姓名：zhang 年龄：20 性别：male 地址：Build 305-503 电话：6081202

职称：tutor（助教） 职务：普通职工

2、姓名：chen 年龄：38 性别：male 地址：Build 3A-302 电话：6088132

职称：associate professor（副教授） 职务：副院长

3、姓名：fan 年龄：30 性别：female 地址：Build 2A-202 电话：6088099

职称：lectuer（讲师） 职务：教研室主任

[思考与扩展]

### 题目3 小型公司人员管理（未做）

某小型公司有四类人员：总经理、技术人员、销售经理、推销员。设计一个基类employee，

派生出manager（总经理）、technician（技术人员）、salesmanager（销售经理）、saleman（推销员）。销售经理既是经理又是销售人员，兼具两类人员的特点，因此同时继承manager 和salesman 两个类。

1、类定义

1）employee 类：

基本信息：编号、姓名、性别、出生日期、职位、薪水等；

出生日期使用自定义的 Date（日期）类；

其中：基本信息为 private 属性，成员函数为public 属性；

多个构造函数：缺省构造函数、带参数的构造函数、带默认参数的构造函数；

可以从外部访问类成员的友员函数；

2）Date 类：

成员变量：年、月、日

成员函数：SetYear(int year)、SetMonth(int month)、SetDay(int day)

GetYear()、GetMonth()、GetDay()

3）派生类technician：新增属性：工作时间

派生类saleman： 新增属性：销售额、所属部门

2、实现人员信息的录入与显示；

3、计算并显示个人月薪：

月薪计算办法：总经理拿固定月薪8000 元，技术人员按每小时25 元领取月薪；推销员

的月薪按当月销售额的4%提成；销售经理固定月薪5000 元加所管辖部门当月销售总额的5

‰ 。

**[**实验提示**]**

1、在基类中，除了定义构造函数和析构函数，还应统一定义对各类人员信息应有的操

作，规范类族中各派生类的基本行为，但是各类人员的月薪计算方法不同，不能在基类

employee 中统一确定计算方法。各类人员信息的显示内容不同，同样不能在基类employee

中统一确定显示方法。在基类中实现上述功能的函数体应为空，在派生类中根据同名覆盖原

则定义各自的同名函数实现具体功能。

2、将基类employee 分成两个独立文档：employee.h(类声明头文件) 和 employy.cpp（类

实现文件）

3、“职位”的类型设定为int；

**[**测试数据**]**

1、编号：10001 姓名：张可 性别：男 出生日期：75-6-10 职位：总经理

2、编号：10002 姓名：李兵 性别：男 出生日期：79-10-10 职位：销售经理 所属部门：华北地区

3、编号：10003 姓名：王刚 性别：男 出生日期：80-12-10 职位：销售员 所属部门：华北地区

4、编号：10004 姓名：陈浩月 性别：女 出生日期：82-1-10 职位：销售员 所属部门：华北地区

5、编号：10005 姓名：宋书 性别：男 出生日期：79-1-19 职位：技术人员 工作时间：120 小时

6、编号：10006 姓名：付强 性别：男 出生日期：82-4-12 职位：销售员 所属部门：华东地区

## 3.7 实验七 运算符重载

### 题目1 重载<，>， ==

设计字符串类String，用来存放不定长的字符串，重载运算符“= =”，“>”，“<”，用于两个字符串的大于、小于和等于的比较运算。

源代码：

/\*

\* 7-1.cpp

\*

\* Created on: 2018年5月24日

\* Author: asus

\*/

#include <iostream>

#include <string.h>

using namespace std;

class String {

public:

String();

String(const char \*);

friend void operator==(String, String);

friend void operator>(String, String);

friend void operator<(String, String);

protected:

const char \*p;

};

String::String() {

p = NULL;

}

String::String(const char \*a) {

p = a;

}

void operator==(String a, String b) {

if (strcmp(a.p, b.p) == 0)

cout << "\"" << a.p << "\"" << " == " << "\"" << b.p << "\"" << endl;

else

cout << "\"" << a.p << "\"" << " != " << "\"" << b.p << "\"" << endl;

}

void operator<(String a, String b) {

if (strcmp(a.p, b.p) < 0)

cout << "\"" << a.p << "\"" << " < " << "\"" << b.p << "\"" << endl;

else

cout << "\"" << a.p << "\"" << " >= " << "\"" << b.p << "\"" << endl;

}

void operator>(String a, String b) {

if (strcmp(a.p, b.p) > 0)

cout << "\"" << a.p << "\"" << " > " << "\"" << b.p << "\"" << endl;

else

cout << "\"" << a.p << "\"" << " =< " << "\"" << b.p << "\"" << endl;

}

int main() {

char const a[] = "China", b[] = "china";

String s(a), s1(b);

s == s1;

s > s1;

s < s1;

}

**[**测试数据**]**

1、 China china

2、 National Computer

3、 Examination Rank

4、 swust swust

结果：

"China" != "china"

"China" =< "china"

"China" < "china"

### 题目2 重载　　时间加减法

设计 CTime 类

1、成员变量：hour、minute、second；

2、多个构造函数；

显示函数：Display()；

设置函数：SetHour(int hour)、SetMinute(int min)、 SetSecond(int sec)；

3、重载++运算符，模拟钟表走动，每次走一秒，每60 秒进一分钟，秒又从0 开始计算；每60 分进一小时，分钟从0 开始计算。

源代码：

/\*

\* 7-2.cpp

\*

\* Created on: 2018年5月24日

\* Author: asus

\*/

#include <iostream>

using namespace std;

class CTime {

private:

int hour, minute, second;

public:

CTime();

CTime(int, int, int);

CTime operator++();

CTime operator++(int);

void settime();

void display();

void sethour(int hour);

void setmin(int min);

void setsec(int sec);

};

CTime::CTime() {

hour = 0;

minute = 0;

second = 0;

}

CTime::CTime(int h, int m, int s) {

hour = h;

minute = m;

second = s;

}

CTime CTime::operator++() {

++second;

if (second == 60) {

second = 0;

++minute;

if (minute == 60) {

second = 0;

minute = 0;

++hour;

if (hour == 24) {

second = 0;

minute = 0;

hour = 0;

}

}

}

return CTime(hour, minute, second);

}

CTime CTime::operator++(int a) {

second++;

if (second == 60) {

second = 0;

minute++;

if (minute == 60) {

second = 0;

minute = 0;

hour++;

if (hour == 24) {

second = 0;

minute = 0;

hour = 0;

}

}

}

return CTime(hour, minute, second);

}

void CTime::display() {

cout << hour << ":" << minute << ":" << second << endl;

}

void CTime::settime(){

cout<<"please input hour:";

cin>>hour;

sethour(hour);

cout<<"please input minute:";

cin>>minute;

setmin(minute);

cout<<"please input second:";

cin>>second;

setsec(second);

}

void CTime::sethour(int hh) {

while (hh < 0 || hh >= 24) {

cout << "hour error please reset hour:";

cin >> hh;

}

hour = hh;

}

void CTime::setmin(int mm) {

while (mm < 0 || mm >= 60) {

cout << "minute error please reset minute:";

cin >> mm;

}

minute = mm;

}

void CTime::setsec(int ss) {

while (ss < 0 || ss >= 24) {

cout << "second error please reset second:";

cin >> ss;

}

second = ss;

}

int main() {

CTime c;

c.settime();

c++;

++c;

c.display();

}

结果：

please input hour:24

hour error please reset hour:23

please input minute:69

minute error please reset minute:59

please input second:70

second error please reset second:23

23:59:25

### 题目3：重载 类对象作参数

假设某班有N（5）名学生，开设M（3）门课程，依次为高数、英语、计算机，计算班级单科平均成绩。

设计Student 类

class student

{……

char name[10];

int deg1,deg2,deg3;

｝

成员变量：name[10]、int deg1,deg2,deg3;

设置函数：

1. 构造函数 student(){deg1=deg2=deg3=0;}
2. student(char na[],int d1,int d2,int d3)；
3. friend student operator+(student s1,student s2)；
4. friend void avg(student &s,int n)；

要求输出学生成绩清单、每门课的班级平均分。

源代码：

/\*

\* 7-3.cpp

\*

\* Created on: 2018年5月31日

\* Author: asus

\*/

#include <iostream>

#include <string.h>

using namespace std;

class student {

protected:

char name[10];

int deg1, deg2, deg3;

public:

student();

student(char na[], int d1, int d2, int d3);

friend student operator+(student s1, student s2);

friend void avg(student &s, int n);

};

student::student() {

deg1 = 0;

deg2 = 0;

deg3 = 0;

}

student::student(char na[], int d1, int d2, int d3) {

strcpy(name, na);

deg1 = d1;

deg2 = d2;

deg3 = d3;

}

student operator+(student s1, student s2) {

char na[20];

strcpy(na, s1.name);

strcat(na, s2.name);

int d1 = s1.deg1 + s2.deg1;

int d2 = s1.deg2 + s2.deg2;

int d3 = s1.deg3 + s2.deg3;

return student(na, d1, d2, d3);

}

void avg(student &s, int n) {

cout << "name:" << s.name << endl;

cout << "ave:" << (s.deg1 + s.deg2 + s.deg3) / 3.0 << endl;

}

int main() {

char na[10] = "stu1";

student s(na, 60, 67, 78);

avg(s, 3);

}

结果：

name:stu1

ave:68.3333

## 3.8 实验八 多态与虚函数

### 题目1 图形类

用虚函数实现第六实验的第一道题目：不同图形面积的计算。

要求：

使用基类指针，通过基类指针调用确定的派生类对象中的面积计算函数，实现面积的计算。

源代码：

/\*

\* 8-1.cpp

\*

\* Created on: 2018年6月7日

\* Author: asus

\*/

#include <iostream>

#include <cmath>

using namespace std;

#define PI 3.1415926

class shape {

protected:

double m\_x, m\_y;

public:

shape() {

m\_x = 0;

m\_y = 0;

}

shape(double x, double y) {

m\_x = x;

m\_y = y;

}

virtual double area()=0;

};

class circle: public shape {

protected:

double m\_r;

public:

circle() {

m\_r = 1;

}

circle(double x, double y, double r) {

m\_x = x;

m\_y = y;

m\_r = r;

}

double area() {

return PI \* m\_r \* m\_r;

}

};

class rectangle: public shape {

public:

rectangle() {

m\_x = 1;

m\_y = 1;

}

rectangle(double x, double y) {

m\_x = x;

m\_y = y;

}

double area() {

return m\_x \* m\_y;

}

};

class triangle: public shape {

protected:

int m\_x2, m\_y2, m\_x3, m\_y3;

public:

triangle() {

m\_x = 0;

m\_y = 1;

m\_x2 = 0;

m\_y2 = 0;

m\_x3 = 1;

m\_y3 = 0;

}

triangle(double x1, double y1, double x2, double y2, double x3, double y3) {

m\_x = x1;

m\_y = y1;

m\_x2 = x2;

m\_y2 = y2;

m\_x3 = x3;

m\_y3 = y3;

while (f(m\_x, m\_y, m\_x2, m\_y2, m\_x3, m\_y3) == false) {

cout << "该数据无法构成三角形" << endl;

cout << "请输入三角形的三个顶点:";

cin >> m\_x >> m\_y >> m\_x2 >> m\_y2 >> m\_x3 >> m\_y3;

}

}

bool f(double x1, double y1, double x2, double y2, double x3, double y3) {

double a, b, c;

a = edge(x1, y1, x2, y2);

b = edge(x1, y1, x3, y3);

c = edge(x2, y2, x3, y3);

if (a == 0 || b == 0 || c == 0)

return false;

if (a + b <= c)

return false;

if (a + c <= b)

return false;

if (b + c <= a)

return false;

return true;

}

double edge(double x1, double y1, double x2, double y2) {

return sqrt((x1 - x2) \* (x1 - x2) + (y1 - y2) \* (y1 - y2));

}

double area() {

double a, b, c, p;

a = edge(m\_x, m\_y, m\_x2, m\_y2);

b = edge(m\_x, m\_y, m\_x3, m\_y3);

c = edge(m\_x2, m\_y2, m\_x3, m\_y3);

p = (a + b + c) / 2.0;

return sqrt(p \* (p - a) \* (p - b) \* (p - c));

}

};

int main() {

shape \*p;

circle c(30,45,35);

rectangle r(14,45);

triangle t(34, 45, 89, 45, 54, 67);

p=&c;

cout << "circle area:" << p->area() << endl;

p=&r;

cout << "rectangle area:" << p->area() << endl;

p=&t;

cout << "triangle area:" << p->area() << endl;

}

**[**测试数据**]**

1、Circle：

圆心（30，45） 半径：35

2、Rectangle：

长：14 宽：45

3、Triangle：

顶点：（34，45）、（89，45）、（54，67）

结果：

circle area:3848.45

rectangle area:630

triangle area:605

### 题目2 员工管理(未做)

1、将第二次实验中题目三不同员工的工资计算函数pay()，改用虚函数实现；

2、将每个类中的信息显示函数：display()改用虚函数实现；

3、对每个类增加一个升职函数：promote()；

**[**实验提示**]**

1、employee（）类中的pay()和display()设计为纯虚函数；在主函数中便可以依据赋值兼容规则用基类类型的指针数组来处理不同派生类的对象；

2、由于不同类的对象调用升级函数promote()需要赋于不同的实参，难以对各类对象统一处理，因此，employee()中promote()定义为虚函数，各派生类再定义同名函数；

**[**测试数据**]**

自己设计测试数据，并详细记录测试数据；

## 3.9 实验九 文件操作

### 题目1 二进制文件操作

从键盘输入字符串并保存成文件。

要求：

1、输入字符串中包括：大小写字母、空格、数字等字符，以“#”作为字符串输入的结

束条件；

2、对输入的字符进行加密并保存，以二进制文件方式保存，用户可以输入保存路径，

如：C：\file.txt ；

3、打开文件时，对加密文件进行解密并显示在屏幕上。

源代码：

/\*

\* 9-1.cpp

\*

\* Created on: 2018年6月14日

\* Author: asus

\*/

#include <iostream>

#include <fstream>

using namespace std;

void file(char []);

void decode(char []);

int main() {

char des[20];

cout<<"please input file name:";

cin>>des;

file(des);

decode(des);

}

void file(char des[]) {

char ch;

ofstream out(des, ios::out);

if (!out) {

cerr << des<<" : File could not be open!" << endl;

return;

}

cout << "please input:";

cin.get(ch);

while (ch != '#') {

if (('a' <= ch && ch <= 'z') || ('A' <= ch && ch <= 'Z')) {

switch (ch) {

case 'w':

ch = 'a';break;

case 'W':

ch = 'A';break;

case 'x':

ch = 'b';break;

case 'X':

ch = 'B';break;

case 'y':

ch = 'c';break;

case 'Y':

ch = 'C';break;

case 'z':

ch = 'd';break;

case 'Z':

ch = 'D';break;

default:

ch += 4;break;

}

}

out.put(ch);

cin.get(ch);

}

out.close();

}

void decode(char des[]) {

char ch;

ifstream in(des, ios::in);

if (!in) {

cerr << des<<" : File could not be open!" << endl;

return;

}

cout << "\ndecode:";

while (in.get(ch)) {

if (('a' <= ch && ch <= 'z') || ('A' <= ch && ch <= 'Z')) {

switch (ch) {

case 'a':

ch = 'w';break;

case 'A':

ch = 'W';break;

case 'b':

ch = 'x';break;

case 'B':

ch = 'X';break;

case 'c':

ch = 'y';break;

case 'C':

ch = 'Y';break;

case 'd':

ch = 'z';break;

case 'D':

ch = 'Z';break;

default:

ch -= 4;break;

}

}

cout << ch;

}

in.close();

}

**[**测试数据**]**

结果：

please input file name:file.txt

please input:dsacsdcydvscuavcudvmds

havcagdsvc#

decode:

dsacsdcydvscuavcudvmds

havcagdsvc

file.txt:(存放的密文)

hwegwhgchzwgyezgyhzqhw

lezgekhwzg

### 题目2 人员信息保存

为第六次实验的题目三“人员管理”添加文件读取保存功能

1、将输入人员信息保存成文件；

3、读取已保存的人员信息文件，并显示文件中保存的人员信息；

源代码：输入信息后自动保存到log.txt文件

/\*

\* 9-2.cpp

\*

\* Created on: 2018年6月23日

\* Author: asus

\*/

#include <iostream>

#include <fstream>

using namespace std;

class Employee {

protected:

int id; //代号

string name; //姓名

string sex; //性别

int age; //年龄

double salary; //工资

public:

Employee();

Employee(int i, string na, string s, int a, float sa);

void input(); //输入信息

void display(); //输出信息

void storelog(); //存储记录到文件

void readlog(); //读文件

};

Employee::Employee() {

id = 1000;

name = "";

sex = "";

age = 18;

salary = 0;

}

Employee::Employee(int i, string na, string s, int a, float sa) {

id = i;

name = na;

sex = s;

age = a;

salary = sa;

}

void Employee::input() {

while (true) {

cout << "请输入ID(1000~9999):";

cin >> id;

if (1000 <= id && id <= 9999)

break;

}

cout << "请输入姓名:";

cin >> name;

while (true) {

cout << "请输入性别(f/m):";

cin >> sex;

if (sex == "f" || sex == "F" || sex == "m" || sex == "M")

break;

}

while (true) {

cout << "请输入年龄(18~65):";

cin >> age;

if (18 <= age && age <= 65)

break;

}

storelog(); //存储人员信息

}

void Employee::display() {

cout << id << "\t" << name << "\t" << sex << "\t" << age << endl;

}

void Employee::storelog() {

ofstream outstuf("log.txt", ios::app);

if (!outstuf) {

cerr << "log.txt : File could not be open!" << endl;

return;

}

outstuf << "id\t\tname\tsex\tage" << endl;

outstuf << id << '\t' << name << "\t\t" << sex << '\t' << age;

outstuf.close();

}

void Employee::readlog(){

char ch;

ifstream in("log.txt", ios::in);

if (!in) {

cerr << "log.txt : File could not be open!" << endl;

return;

}

while(in.get(ch)){

cout<<ch;

}

in.close();

}

int main(){

Employee e;

e.input();

cout << endl;

e.display();

e.readlog();

}

结果：

请输入ID(1000~9999):1111

请输入姓名:李攀

请输入性别(f/m):f

请输入年龄(18~65):21

1111 李攀 f 21

id name sex age

1111 李攀 f 21

log.txt文件：

id name sex age

1111 李攀 f 21

### 题目3 文件合并

合并两个文件成为一个文件

要求：

执行可执行程序，将两个文件合并成一个文件；

例如：combine.exe file1.txt file2.txt file3.txt

Combine.exe 为源程序生成的可执行程序，将file1.txt 和 file2.txt 中的内容合并，存放到file3.txt 这个文件中；

源代码：

/\*

\* 9-3.cpp

\*

\* Created on: 2018年6月23日

\* Author: asus

\*/

#include <iostream>

#include <fstream>

using namespace std;

void filecopy(char src[], char des[]);

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

if (argc != 4) {

cout << "输入格式错误！" << endl;

return 0;

}

filecopy(argv[1], argv[3]);

filecopy(argv[2], argv[3]);

}

void filecopy(char src[], char des[]) {

char ch;

ifstream in(src, ios::in);

ofstream out(des, ios::app);

if (!in) {

cerr << src << " : File could not be open!" << endl;

return;

}

if (!out) {

cerr << des << " : File could not be open!" << endl;

return;

}

while (in.get(ch))

out.put(ch);

in.close();

out.close();

}

**[**测试数据**]**

任意建立两个文件进行测试；

结果：

