**ThreeBand**

**//head.h**

#include<iostream>

#include<string>

#include<fstream>

#include <cstdlib>

#include <time.h>

#include <conio.h>

#include <Windows.h>

#include <mmsystem.h>

#include <assert.h>

#include <stdlib.h>

#pragma comment(lib,"WinMM.Lib")

using namespace std;

void play(string);

class AddressBook;

class contact

{

public:

contact()

{

Name = "null";

Company = "null";

PhoneNumber = "null";

Email = "null";

Classification = "null";

}

void setNumber(int n) { Number = n; }

void print();

void print(int);

string getName() { return Name; }

string getPhoneNumber() { return PhoneNumber; }

void Edit();

void add(int);

void Delete();

void Read(string, int, int);

void Save(ofstream&);

contact& operator=(contact&);

private:

int Number;

string Name;

string Company;

string PhoneNumber;

string Email;

string Classification;

};

class AddressBook

{

public:

AddressBook();

~AddressBook();

void Add();

void Edit(int);

void Delete(int);

bool print(int);

bool printAll();

bool empty() { return count; }

string getName(int i) { return Contact[i - 1].getName(); }

string getPhoneNumber(int i) { return Contact[i - 1].getPhoneNumber(); }

private:

friend class SmartPhone;

contact Contact[200];

int count;

};

class PerInfo

{

public:

PerInfo();

~PerInfo();

void Set();

void print();

void setHeartRate(double i) { HeartRate = i; }

bool STR(string& );

private:

string name;

int age;

double height;

double weight;

double BMI;

double HeartRate;

};

class ThreeBand

{

public:

ThreeBand() :StepNumber(0) {}

void printPerInfo() { P.print(); }

void setPerInfo() { P.Set(); }

void setHertRate();

void printTime();

void setStepNumber();

void Answer(SmartPhone&);

void call(SmartPhone&);

void call(SmartPhone&, string, string);

void addressBook(SmartPhone&);

private:

PerInfo P;

int StepNumber;

};

class SmartPhone

{

public:

SmartPhone(){}

bool PhoneRinging()

{

srand((unsigned)time(NULL));

return rand() % (5) + 0;

}

bool call(string);

AddressBook& getBook() { return book; }

private:

AddressBook book;

};

**//ThreeBand\_definition.cpp**

#include"head.h"

PerInfo::PerInfo()

{

string str;

ifstream read("data1.txt");

read >> str;

if (str.length() == 0)

{

cout << " 初次使用，请输入你的个人信息！\n" << endl;

Set();

read.close();

ofstream Save("data1.txt");

Save << '\*' << name << '\*' << age << '\*' << height << '\*' << weight << '\*';

Save.close();

cout << " 记录成功！\n";

cout << " ";

system("pause");

system("cls");

return;

}

int p1 = 0, p2, p3, p4, p5;

p2 = str.find('\*', 1);

p3 = str.find('\*', p2 + 1);

p4 = str.find('\*', p3 + 1);

p5 = str.find('\*', p4 + 1);

string str2, str3, str4;

name.assign(str, p1 + 1, p2 - p1 - 1);

str2.assign(str, p2 + 1, p3 - p2 - 1);

str3.assign(str, p3 + 1, p4 - p3 - 1);

str4.assign(str, p4 + 1, p5 - p4 - 1);

age = atof(str2.c\_str());

height = atof(str3.c\_str());

weight = atof(str4.c\_str());

BMI = weight / (height\*height);

}

PerInfo::~PerInfo()

{

ofstream Save("data1.txt");

Save << '\*' << name << '\*' << age << '\*' << height << '\*' << weight << '\*';

Save.close();

}

void PerInfo::Set()

{

string str;

cout << " 姓名："; cin >> name;

while (1)

{

cout << " 年龄："; cin >> str;

if (!STR(str))

{

age = atof(str.c\_str()); break;

}

else

{

cout << " 请输入整数！\n ";

system("pause");

system("cls");

}

}

while (1)

{

cout << " 身高(m)："; cin >> str;

if (!STR(str))

{

height = atof(str.c\_str()); break;

}

else

{

cout << " 请输入整数！\n ";

system("pause");

system("cls");

}

}

while (1)

{

cout << " 体重(kg)："; cin >> str;

if (!STR(str))

{

weight = atof(str.c\_str()); break;

}

else

{

cout << " 请输入整数！\n ";

system("pause");

system("cls");

}

}

BMI = weight / (height\*height);

}

void PerInfo::print()

{

cout << "\n 你的个人健康信息如下:\n";

cout << " 姓名：" << name << endl;

cout << " 年龄：" << age << endl;

cout << " 身高(m)：" << height << endl;

cout << " 体重(kg)：" << weight << endl;

cout << " BMI：" << BMI << endl;

cout << " ";

system("pause");

system("cls");

}

bool PerInfo::STR(string& str)

{

int count = 0;

for (int i = 0; i < str.length(); i++)

{

if (((57 < str[i])||(str[i] < 48))&& (str[i] != 46))

count++;

}

return count;

}

void ThreeBand::setHertRate()

{

string S;

time\_t t\_start, t\_end;

t\_start = time(NULL);

cout << " 按任意字符记录心跳(一个字符代表一次)，回车键结束：" << endl;

cout << " "; cin >> S;

t\_end = time(NULL);

double time = difftime(t\_end, t\_start);

double count = S.length();

cout << " 心跳次数：" << count << " 用时：" << time << "秒" << endl;

double HeartRate = 0;

HeartRate = count / (time / 60);

cout << " 心率为：" << HeartRate << endl;

P.setHeartRate(HeartRate);

cout << " ";

system("pause");

system("cls");

}

void ThreeBand::printTime()

{

time\_t t = time(0);

char tmp[32] = { NULL };

strftime(tmp, sizeof(tmp), "%Y-%m-%d %H:%M:%S", localtime(&t));

cout << "时间：" << tmp;

}

void ThreeBand::setStepNumber()

{

int i; string str;

while (1)

{

cout << " 1【记录步数】 0【退出记录】 2【显示步数】：\n";

cout << " "; cin >> i;

if (i == 1)

{

cout << " 按任意字符记录步数(一个字符代表一次)，回车键结束：\n";

cout << " "; cin >> str;

StepNumber += str.length();

cout << " 记录成功！\n";

cout << " ";

system("pause");

system("cls");

}

else if (i == 0)

{

system("cls"); return;

}

else if (i == 2)

{

cout << " 当前步数为：" << StepNumber << endl;

cout << " ";

system("pause"); system("cls");

}

else

{

cout << " 输入有误，请重新输入！\n"; cout << " ";

system("pause");

system("cls");

}

}

}

void ThreeBand::Answer(SmartPhone& S)

{

if (!S.PhoneRinging())

{

cout << " 电话来了！\n";

play("7890.wav");

while (1)

{

cout <<" 输入： 1【接听】 0【挂断】\n";

int i;

cout << " "; cin >> i;

if (i == 1)

{

while (1)

{

cout << " 通话中(0【挂断】)....\n";

cout << " "; cin >> i;

if (i == 0)

{

system("cls");

return;

}

else

{

cout << " 输入有误，请重新输入！\n"; cout << " ";

system("pause");

system("cls");

}

}

}

else if (i == 0)

{

cout << " ";

system("pause");

system("cls");

return;

}

else

{

cout << " 输入有误，请重新输入！\n"; cout << " ";

system("pause");

system("cls");

cout << " 电话来了！\n";

}

}

}

return;

}

void ThreeBand::call(SmartPhone& S)

{

string str;

cout << " 输入电话号码：";

cin >> str;

cout << " 正在播打：" << str << endl;

play("1670.wav");

if (S.call(str))

{

while (1) {

cout << " 通话中....\n 输入0挂断：";

int i;

cin >> i;

if (i == 0)

{

system("cls");

return;

}

else

{

cout << " 输入有误，请重新输入！\n"; cout << " ";

system("pause");

system("cls");

}

}

}

else

{

cout << " 暂时无人接听...\n";

play("356.wav");

}

system("cls");

return;

}

void ThreeBand::call(SmartPhone& S, string name, string PhoneNumber)

{

cout << " 正在播打：" << name << " " << PhoneNumber << endl;

play("1670.wav");

if (S.call(PhoneNumber))

{

while (1) {

cout << " 通话中....\n 输入0挂断：";

int i;

cin >> i;

if (i == 0)

{

system("cls");

return;

}

else

{

cout << " 输入有误，请重新输入！\n"; cout << " ";

system("pause");

system("cls");

}

}

}

else

{

cout << " 暂时无人接听...\n";

play("356.wav");

}

system("cls");

return;

}

void ThreeBand::addressBook(SmartPhone& S)

{

AddressBook& B = S.getBook();

int flag = -1;

while (1)

{

cout << " 输入 1【新建联系人】 2【显示所有联系人】 0【退出】 \n";

cout << " "; cin >> flag;

if (flag == 0)

{

cout << " ";

system("cls");

return;

}

else if (flag == 1)

{

system("cls");

B.Add();

cout << " 新建成功！\n";

cout << " ";

system("pause");

system("cls");

}

else if (flag == 2)

{

system("cls");

if (!B.printAll())

continue;

while (1)

{

if (!B.empty())

break;

cout << " 输入编号以查看联系人(0【返回上一层】)：\n";

cout << " "; cin >> flag;

if (flag != 0)

{

system("cls");

if (!B.print(flag))

{

B.printAll();

continue;

}

while (1)

{

int f;

cout << " 输入 1【编辑】 2【删除】 3【拨电话】 0【返回】 -1【关闭通讯录】\n";

cout << " "; cin >> f;

if (f == 2)

{

B.Delete(flag);

cout << " ";

system("pause");

system("cls");

B.printAll();

break;

}

else if (f == 1)

{

B.Edit(flag);

cout << " 修改成功！\n";

cout << " ";

system("pause");

system("cls");

B.printAll();

break;

}

else if (f == 3)

{

system("cls"); call(S, B.getName(flag), B.getPhoneNumber(flag));

system("cls");

B.printAll();

break;

}

else if (f == 0)

{

system("cls"); B.printAll(); break;

}

else if (f == -1)

{

system("cls"); return;

}

else

{

cout << " 输入有误，请重新输入！\n"; cout << " ";

system("pause");

system("cls");

B.print(flag);

}

}

}

else

{

flag = -1;

system("cls");

break;

}

}

}

else

{

cout << " 输入有误，请重新输入！\n"; cout << " ";

system("pause");

system("cls");

}

}

}

bool SmartPhone::call(string n)

{

srand((unsigned)time(NULL));

return rand() % (2) + 0;

}

**//AddressBook\_definition.cpp**

#include"head.h"

BOOL MByteToWChar(LPCSTR lpcszStr, LPWSTR lpwszStr, DWORD dwSize)

{

// Get the required size of the buffer that receives the Unicode

// string.

DWORD dwMinSize;

dwMinSize = MultiByteToWideChar(CP\_ACP, 0, lpcszStr, -1, NULL, 0);

assert(dwSize >= dwMinSize);

// Convert headers from ASCII to Unicode.

MultiByteToWideChar(CP\_ACP, 0, lpcszStr, -1, lpwszStr, dwMinSize);

return TRUE;

}

void play(string s)

{

wchar\_t wText[30] = { 0 };

MByteToWChar(s.c\_str(), wText, sizeof(wText) / sizeof(wText[0]));

PlaySound(wText, NULL, SND\_FILENAME | SND\_SYNC);

}

AddressBook::AddressBook()

{

string str;

count = 0;

ifstream read("data.txt");

read >> str;

if (str.length() == 0)

{

read.close();

return;

}

int p1 = 0, p2 = 0, i = 0;

while (p2 != (str.length() - 1))

{

p1 = str.find('{', p2);

p2 = str.find('}', p1 + 1);

Contact[i++].Read(str, p1, p2);

Contact[i - 1].setNumber(i);

}

count = i;

read.close();

}

AddressBook::~AddressBook()

{

ofstream Save("data.txt");

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

{

Save << '{';

Contact[i].Save(Save);

Save << '}';

}

Save.close();

}

void AddressBook::Add()

{

int k;

k = count++;

Contact[k].add(k);

}

void AddressBook::Edit(int i)

{

Contact[i - 1].Edit();

}

void AddressBook::Delete(int i)

{

Contact[--i].Delete(); count--;

for (i; i < count; i++)

{

contact temp = Contact[i];

Contact[i] = Contact[i + 1];

Contact[i + 1] = temp;

Contact[i].setNumber(i + 1);

}

}

bool AddressBook::print(int i)

{

if (i<1 || i>count)

{

cout << " 查无此人！\n";

cout << " ";

system("pause");

system("cls");

return false;

}

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

Contact[i - 1].print();

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n";

return true;

}

bool AddressBook::printAll()

{

if (count == 0)

{

cout << " 联系人为空，请新建联系人！\n";

cout << " ";

system("pause");

system("cls");

return false;

}

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

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

Contact[i].print(1);

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

return true;

}

void contact::print()

{

cout << " 编号：" << Number << " 姓名：" << Name << " 单位：" << Company

<< " 电话：" << PhoneNumber << " 邮箱：" << Email << " 分类：" << Classification << endl;

}

void contact::print(int i)

{

cout << " 编号：" << Number << " " << "姓名：" << Name<< endl;

}

void contact::Save(ofstream& save)

{

save << '\*' << Name << '\*' << Company << '\*' << PhoneNumber

<< '\*' << Email << '\*' << Classification << '\*';

}

void contact::Edit()

{

cout << " 输入修改后的信息！\n";

cout << " 编号为：" << Number << endl;

cout << " 姓名：";

cin >> Name;

cout << " 单位：";

cin >> Company;

cout << " 电话:";

cin >> PhoneNumber;

cout << " 邮箱:";

cin >> Email;

cout << " 分类:";

cin >> Classification;

}

void contact::add(int n)

{

cout << " 输入新增联系人的信息：\n";

Number = n + 1;

cout << " 编号为：" << Number << endl;

cout << " 姓名：";

cin >> Name;

cout << " 单位：";

cin >> Company;

cout << " 电话:";

cin >> PhoneNumber;

cout << " 邮箱:";

cin >> Email;

cout << " 分类:";

cin >> Classification;

}

void contact::Delete()

{

Number = 0;

Name = "null";

Company = "null";

PhoneNumber = "null";

Email = "null";

Classification = "null";

cout << " 删除成功！\n";

}

void contact::Read(string Str, int k1, int k2)

{

int r1 = k1 + 1, r2 = k1 + 1, i = 0;

while (r2 != (k2 - 1))

{

r1 = Str.find('\*', r2);

r2 = Str.find('\*', r1 + 1);

switch (i++)

{

case 0:Name.assign(Str, r1 + 1, r2 - r1 - 1); break;

case 1:Company.assign(Str, r1 + 1, r2 - r1 - 1); break;

case 2:PhoneNumber.assign(Str, r1 + 1, r2 - r1 - 1); break;

case 3:Email.assign(Str, r1 + 1, r2 - r1 - 1); break;

case 4:Classification.assign(Str, r1 + 1, r2 - r1 - 1); break;

}

}

}

contact& contact::operator=(contact& C)

{

this->Name = C.Name;

this->Company = C.Company;

this->PhoneNumber = C.PhoneNumber;

this->Email = C.Email;

this->Classification = C.Classification;

return \*this;

}

**//test.cpp**

#include"head.h"

int main()

{

ThreeBand B; SmartPhone S; int i;

while (1)

{

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << " \* "; B.printTime(); cout << " \*\n";

cout << " \* \*\n";

cout << " \* 1【测心率】 2【计步】 3【显示健康信息】 4【拨打电话】 5【通讯录】 0【退出】\*\n";

cout << " \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "\n 输入你想执行的操作:"; cin >> i;

switch (i)

{

case 1:B.setHertRate(); break;

case 2:B.setStepNumber(); break;

case 3:B.printPerInfo(); break;

case 4:B.call(S); break;

case 5:B.addressBook(S); break;

case 0:cout << " "; return 0;

default:cout << " 输入有误，请重新输入！\n"; cout << " ";

system("pause");

system("cls");

}

B.Answer(S);

}

return 0;

}