

题目：学生健康管理系统--单链表

学生姓名：郑嘉涛

学号： 20152100015

专业班级：计算机类5班

指导教师：陈红英

学生健康系统--链表

摘 要

目前人们对于健康的关注程度越来越高，而同时学校对于学生的健康也变得更加关注，但是对学生的健康档案管理却没有一个较好的解决办法，基于此，我们完成了用链表实现的学生健康管理系统。使用链表来完成有多种优势，链表的插入删除操作方便，链表是动态的，可以随着学生的数量改变而改变,链表具有多种优点，可以保存许多信息。使用了C++的面向对象特性，简化了开发过程。

**关键词**：C++；面向对象；链表；

前言

**实验目的：**

学会使用C++编写链表数据结构，增强对链表这种数据结构的理解。在开发学生健康系统的过程中，学会模板的使用。并确实开发出一个简单实用的学生健康管理系统，并使用该学生健康管理系统保存学生数据并对学生数据进行管理，最后在将学生数据保存在文件之中，方便下次使用。

**实验意义：**

学会数据结构中的链表，掌握如何编写链表数据结构，帮助掌握链表的各项特性，学会使用模板。开发的学生健康管理系统拥有多种多样的基本功能，各项功能能够满足实用，且在作为健康管理系统的同时也可作为信息管理系统，满足日常需求。

**实验内容概述：**

建立了学生类，学生类包含多种数据类型，同时也包含有多种数据，如学生的Id、姓名、生日、性别和健康状况。

学生类头文件代码如下：

#ifndef \_STUDENT\_H

#define \_STUDENT\_H

#include<fstream>

using namespace std;

class Student{

private:

char Id[13];

char name[16];

char birth[10];

int sex;

int health;

public:

//构造函数

Student(char \*Id = NULL , char \*name = NULL , char \*birth = NULL , int sex = 0 , int health = 0);

Student(Student &t);

~Student();

//输出student对象

void display();

//重载输入、输出函数

friend istream& operator >> (istream &is , Student &t);

friend ostream& operator << (ostream &os , Student &t);

//写入文件，和从文件读取，希望能够使用传入的地址来进行读取写入

bool writeToFile(char \*address = "C:\\student.txt");

bool readFromFile(char \*address = "C:\\student.txt");

//重载 文件读取写入

friend ifstream& operator >> (ifstream &ifs , Student &t);

friend ofstream& operator << (ofstream &ofs , Student &t);

//重载 ==

friend bool operator == (Student &t1 , Student &t2);

//这个类的set函数和get函数

bool setId(char \*id);

bool setName(char \*name);

bool setBirth(char \*birth);

bool setSex(int sex);

bool setHealth(int health);

char\* getId();

char\* getName();

char\* getBirth();

int getSex();

int getHealth();

friend Student\* newStudent();

};

#endif

同时采用了文件来保存文件，通过对链表的操作来保存所输入的学生信息。

学生类的文件操作代码如下：

bool Student::writeToFile(char \*address){

ofstream out(address);

if(!out)return false;

out<<this->getId()<<","<<this->getName()<<","<<this->getBirth()

<<","<<this->getSex()<<","<<this->getHealth()<<".";//类内部函数不用使用get函数

return true;

}

bool Student::readFromFile(char \*address){

ifstream in(address);

if(!in)return false;

char ch;

in.get(ch);

for(int i = 0 ; ch != ',' && i < 12; i++){

Id[i] = ch;

in.get(ch);

}

in.get(ch);

for(int i = 0 ; ch != ',' && i < 16 ; i++){

name[i] = ch;

in.get(ch);

}

in.get(ch);

for(int i = 0 ; ch != ',' && i < 9 ; i++){

birth[i] = ch;

in.get(ch);

}

in.get(ch);

sex = ch;

in.get(ch);in.get(ch);

health = ch;

return true;

}

使用模板类和链表来开发学生健康管理系统，模板类可以满足多种需要，对不同的数据都可使用，而链表具有多种优点，链表可以动态创建空间，不会造成空间浪费，链表的插入删除操作时间效率很高，而且链表存在多种类型，如双向链表、循环链表等，本实验主要采用的链表类型是简单单链表。

实验设计

**采用的关键技术：**

链表：链表是一种物理[存储单元](http://baike.baidu.com/view/1223079.htm" \t "http://baike.baidu.com/_blank)上非连续、非顺序的[存储结构](http://baike.baidu.com/view/2820182.htm" \t "http://baike.baidu.com/_blank)，[数据元素](http://baike.baidu.com/view/38785.htm" \t "http://baike.baidu.com/_blank)的逻辑顺序是通过链表中的[指针](http://baike.baidu.com/view/159417.htm" \t "http://baike.baidu.com/_blank)链接次序实现的。链表由一系列结点（链表中每一个元素称为结点）组成，结点可以在运行时动态生成。每个结点包括两个部分：一个是存储[数据元素](http://baike.baidu.com/view/38785.htm" \t "http://baike.baidu.com/_blank)的数据域，另一个是存储下一个结点地址的[指针](http://baike.baidu.com/view/159417.htm" \t "http://baike.baidu.com/_blank)域。 相比于[线性表](http://baike.baidu.com/view/178622.htm" \t "http://baike.baidu.com/_blank)[顺序结构](http://baike.baidu.com/view/552878.htm" \t "http://baike.baidu.com/_blank)，操作复杂。由于不必须按顺序存储，链表在插入的时候可以达到O(1)的复杂度，比另一种线性表顺序表快得多，但是查找一个结点或者访问特定编号的结点则需要O(n)的时间，而线性表和顺序表相应的时间复杂度分别是O(logn)和O(1)。

类模板：是对一批仅仅成员数据类型不同的类的抽象，程序员只要为这一批类所组成的整个类家族创建一个类模板，给出一套程序代码，就可以用来生成多种具体的类，（这类可以看作是类模板的实例），从而大大提高编程的效率。

**主要设计思想与设计流程：**

设计思想：

建立一个学生类，如果直接创建学生类的对象同样也可操作，但是这样对单独的学生对象操作的意义不大，只有对多个数据操作是才会有较大影响，比如建立一个学生数组，基于这个原因，学生类中的数据即使使用public修饰关系也不大，但是为了程序的封装性，为了更好的利用C++面向对象的特性，还是将学生类的数据成员都设置了private。但是对所有的数据都设置了set函数和get函数，方便了后面的计算。

然后建立链表的结点类，其中包含模板类的数据data，和指向下一结点的指针next，为了后面操作的方便，故在此将结点类的数据都设置成了公有public，结点类的成员方法很简单，仅创建了五个方法，在析构函数中将非空的结点next删除释放空间。

还建立了链表类，链表类里面只包含一个数据成员，即一个指向结点类的指针，通过链表类的成员方法对这个指针进行操作，即可完成多种多样的功能，但对于模板类，一些成员方法是难以实现的，考虑到模板的使用多种多样，所以建立了几个专门针对学生链表的普通函数，用来完成一些基础而特殊的操作，如查找学生，写入文件和向文件读取学生数据。

最后编写了入口函数main函数，这也是程序界面的主要部分，在此处编写了一些功能代码，用户可以通过相应的序号选择功能，是面向用户的最重要的代码。

设计流程：

学生类

建立链表结点类

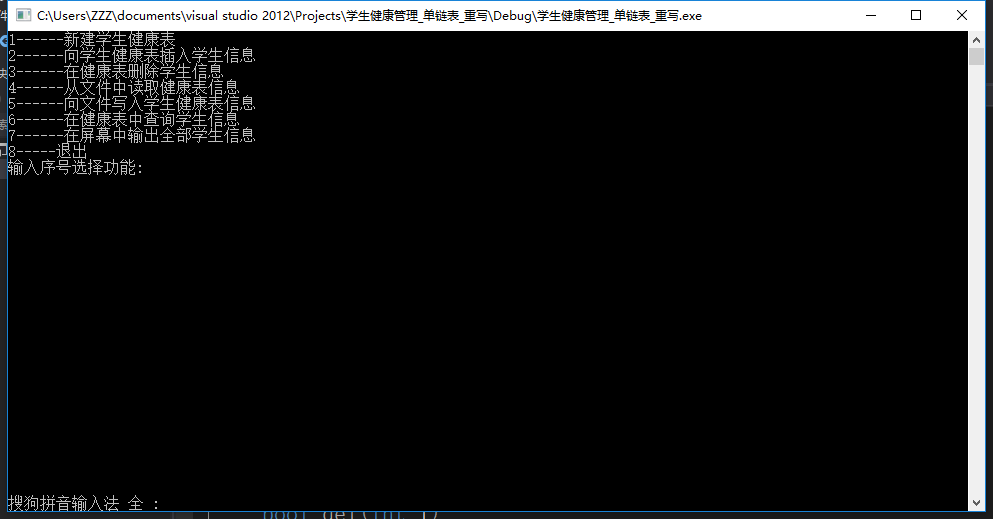
建立链表类

操作使用

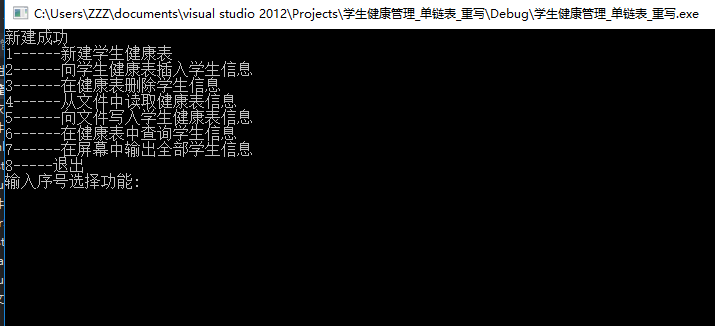
实验实现

**主要功能模块分析:**

主要功能界面及模块：

****

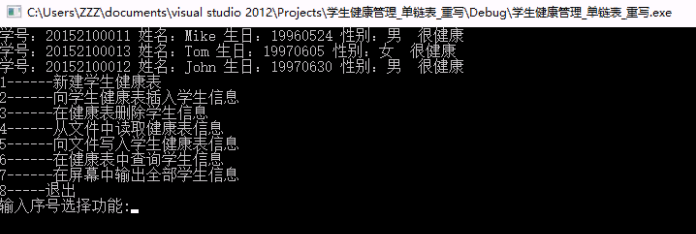
这是用户操作的主要界面，用户的所有操作都围绕这个界面进行。并且里面设置了刷新，使得程序不会受到之前输出的数据干扰，也使得界面更加简洁明了，易于使用。



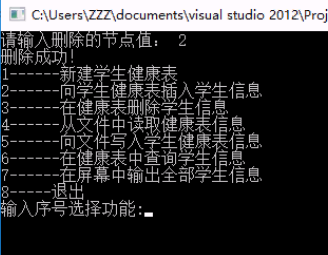
这是程序的新建界面，新建界面并不输入数据，但是在此处会对链表进行初始化，防止后面操作时出现了空指针，防止直接操作空指针导致程序奔溃，可以更好的保证程序的安全性。



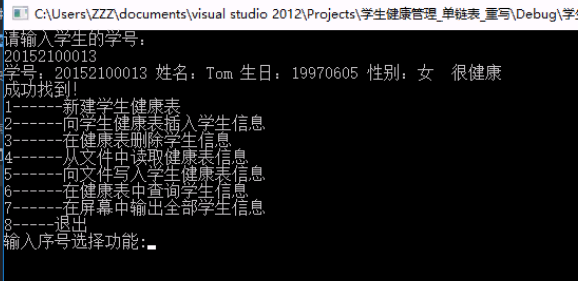
这是程序的插入界面，当输入数据不符合要求时，会有相应的输出提示，并且用户需重新输入直到数据符合要求为止，且其中使用了cin.sync()函数，防止前面输入的数据对后面有影响，并且会提示用户插入的位置，当输入值过大时，会自动插在单链表的末端。如果有很多数据，用户需不断进入此界面进行学生数据的插入。



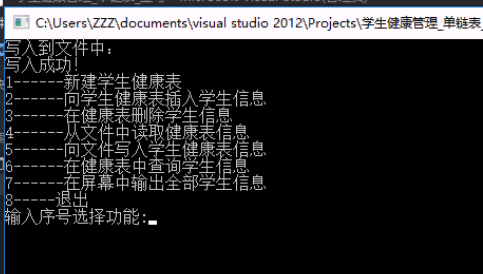
这是单链表的输出界面，用户可以使用这个输出信息的功能，方便查看链表中已有的数据，方便用户整理。

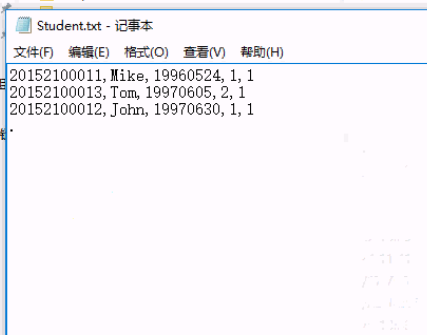


这是程序的删除界面，删除界面是通过用户输入所要删除的结点来删除对应的节点。

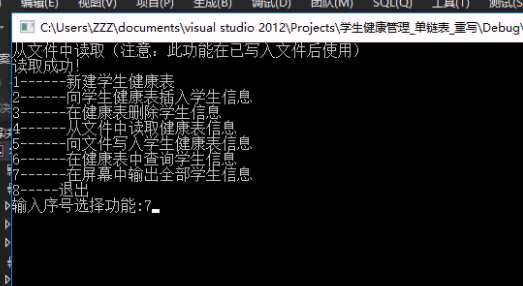


这是程序的查询界面，输入学生的学号，即可输出所查找的学生的所有信息。





这是程序的写入界面，写入后即可在计算机本地查看文件，更加方便查看。可以打开文件查看，但是不推荐修改文件，所有修改应当在这个程序的基础下进行修改，手动修改会造成在运行程序时出现错误。



这是程序的读取界面，读取之前要新建链表，否则出错。读取只能在两种情况下使用，一是确保本地有文件时可以进行读取、二是在程序运行过程中，有过写文件的操作，在此时需要返回上次的数据时使用。

**采用的软硬件平台介绍:**

软件平台：

Windows10：

美国[微软](http://baike.baidu.com/view/2353.htm" \t "http://baike.baidu.com/_blank)公司所研发的新一代跨平台及设备应用的[操作系统](http://baike.baidu.com/subview/880/4940471.htm" \t "http://baike.baidu.com/_blank)。

Visual studio 2012：

[Visual Studio](http://baike.baidu.com/view/28727.htm" \t "http://baike.baidu.com/item/_blank)是微软公司推出的开发环境。是最流行的[Windows](http://baike.baidu.com/view/4821.htm" \t "http://baike.baidu.com/item/_blank)平台应用程序开发环境。

硬件平台：

处理器：Intel Core i7-4710MQ

内存：8GB

显卡：Intel HD Graphics 4600

结果分析和讨论

本实验主要完成以下几个方面的内容：1.建立可以直接操作的学生类、2.建立学生链表、3.结合1、2的内容建立一个学生健康管理系统。

在实验的过程中，出现许多问题。对文件的操作部分，一开始想使用模板类内定义函数直接对文件进行读写操作，但这种情况不能控制，因为模板中存在多种情况。最后专门写了针对学生链表的操作函数进行操作。

程序的插入删除操作也存在多种情况，对于链表，插入和删除有如下情况：1.对头结点进行操作、2.在中间结点操作、3.在尾结点操作。而中间结点和尾结点可以看做是同一种情况。

参考文献

《C++程序设计（第2版）》谭浩强

《数据结构（用面向对象方法和C++语言描述）》

《算法导论》

《数据结构（C语言版）》