实验报告（一）

姓　　名：孟岩

学　　号：111700426

专　　业：微电子科学与工程

年　　级：2017

班　　级：微电二班

指导老师：潘林

实验时间：2020.11.11

数据结构与算法（C 语言版）实验报告

实验一　线性表

一、 实验内容：

利用单链表实现下述应用

目的：线性表的建立、新增、插入、删除和查询

具体内容：学生成绩统计。问题描述如下：

（1）以“学生”为数据元素的线性表

（2）用 typedef、struct 写出其链式存储结构的类型定义，其中“学生”为复合型数据类型，包括：学号、姓名、性别、年龄、考试科目、考试成绩。

（3）统计考试科目的平均成绩，考试科目 3 门，分别是“高数”、“外语”和“物理”

（4）功能 1：根据学号进行成绩录入（新增），删除

（5）功能 2：根据学号进行成绩查询，成绩修改

（6）功能 3：根据考试科目分别求总平均分、男女生平均分

（7）功能 4：显示出各门成绩的最高分同学信息

二、实验目的：

1. 线性表的应用，C 语言和环境的熟悉

2. 理解掌握线性表（顺序表、链表）的建立、新增、插入和删除的程序设计

3. 线性表（顺序表、链表）的应用

三、程序流程图

1. 定义链表，必要元素：学号，姓名，性别，年龄，考试科目，考试成绩

2. 初始化链表，使用 txt 文件进行初始化

3. 添加数据

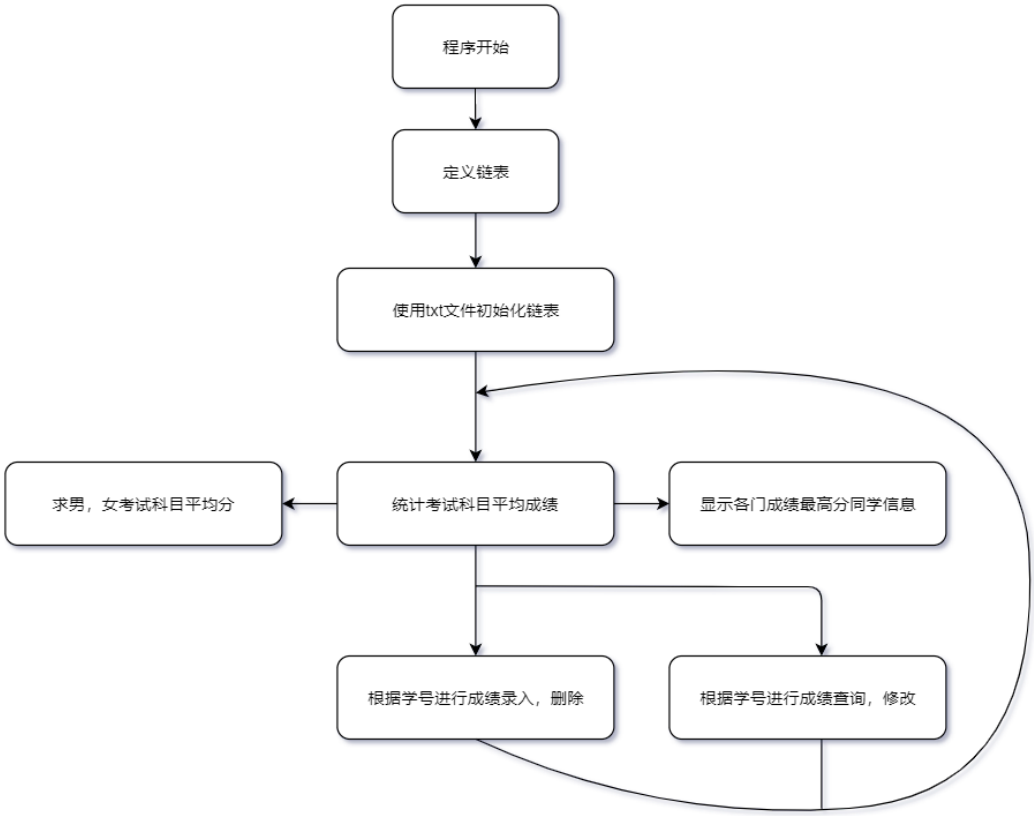
4. 求平均值：高数，外语，物理

5. 添加功能：根据学号，实现插入，删除

6. 添加功能：根据学号，实现查询，成绩修改

7. 添加功能：求各科总平均分，男女分别平均分

8. 添加功能：求各科成绩最高分同学的信息并显示



四、程序设计

1、数据类型定义：

使用 class 进行完全封装，将结构体与链表封装在 class 中

```
class Student
{
  public:
  //定义链表的必要元素
    typedef struct LinkList_Student
    {
      unsigned long student_number;   //学号
      string name;                    //姓名
      string gender;                  //性别
      int age;                       //年龄
      int math_grade;                 //高数分数
      int english_grade;              //英语成绩
      int physics_grade;              //物理成绩

      struct LinkList_Student *next;
    }LinkList_Student,*List;

  private:
```

```
    int size = 100;
    LinkList_Student test_Score = {0};
    string file_addreess = "D:/code/data_structure/LinkList/data.txt";//文件地址

public:
    void setSize(int setSize);
    int getSize(void);
    void split(const char *s,vector<string>& strs, char delim);
    void Init_List();       //初始化链表,封闭操作
    void Input_List(void);  //将文件内容写入链表
    void average(double arg[]);   //求平均成绩,输入数组存储
    void highest_grade(void);     //查找各科分数最高的同学
    List seek_student(unsigned long student_number);   //查看输入学号的同学是否在链表中并返回地址
    void query_grade(void);       //根据学号查成绩
    void amend_grade(void);       //根据学号修改成绩
    void delete_student(void);    //根据学号删除
    void add_student(void);       //添加一条
    Student(string s);
    ~Student();

};
```

2、主要代码

```
//使用文件初始化链表
void Student::Input_List(void)
{
   List p, L;
   L = &(this->test_Score);
   fstream file;
   string s_data;
   stringstream s_change;
   vector<string> vs_data;     //初始化容器

   file.open(this->file_addreess);
   if (!file) {
      cout << "未找到相关文件，无法打开！检查文件路径！" << endl;
      exit(ERROR);
   }

   getline(file,s_data);  //跳过第一行
   while (file.good()) {
      getline(file,s_data);
```

```
        this->split(s_data.data(),vs_data); //字符串

        p = new LinkList_Student;
        s_change<<vs_data[0];  //将 string 转换成其它类型
        s_change>>p->student_number; //学号
        s_change.clear();

        p->name = vs_data[1];      //名字
        p->gender = vs_data[2];    //性别

        s_change<<vs_data[3];      //年龄
        s_change>>p->age;
        s_change.clear();

        s_change<<vs_data[4];      //高数分数
        s_change>>p->math_grade;
        s_change.clear();

        s_change<<vs_data[5];      //英语分数
        s_change>>p->english_grade;
         s_change.clear();

        s_change<<vs_data[6];      //物理分数
        s_change>>p->physics_grade;
         s_change.clear();

        p->next = NULL;
        vs_data.clear();          //清空容器

    }

//将字符串进行分割
void Student::split(const char *s, vector<string>& strs, char delim = ' ')
{
    if(s == nullptr) {
        return;
    }

    const char *head, *tail;
    head = tail = s;

    while(*head != '\0') {
        while(*head != '\0' && *head == delim) {
            head++;
```

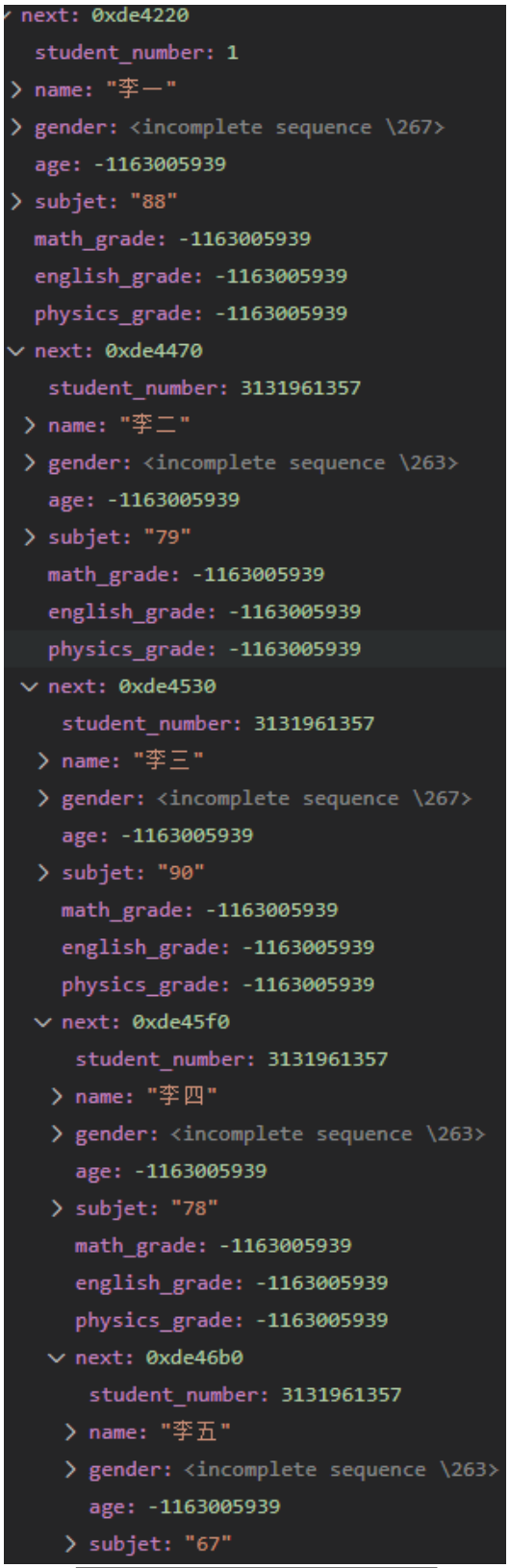


图 1 链表初始化部分内容

```
        }

        tail = head;
        while(*tail != '\0' && *tail != delim) {
            tail++;
        }

        if(head != tail) {
            strs.push_back(string(head,tail));
            head = tail;
        } else {
            break;
        }
    }
}


//查询要查找的学号是否存在于链表，返回指针
Student::List Student::seek_student(unsigned long student_number)
{
    List p;
    p = this->test_Score.next;

    while(p!=NULL && (p->student_number != student_number)){
        p = p->next;
    }

    if(p!=NULL)
        return p;
    else{
        cout<<"该学号不存在，请检查输入"<<endl;
        return NULL;
    }
}

//删除某个节点
void Student::delete_student(void)
{
    List p,d;
    p = &(this->test_Score);
    unsigned int student_number = 0;
    cout<<"请输入要删除的学号：";
    cin>>student_number;
```

```
    while(p->next!=NULL && (p->next->student_number != student_number)){
        p = p->next;
    }

    if(p->next!=NULL){
        d = p->next;
        p->next = p->next->next;
        free(d);
    }
    else{
        cout<<"该学号不存在，请检查输入"<<endl;
    }
}

int main()
{
    Student s("D:/code/data_structure/LinkList/data.txt");
    double average[9] = {0};//mm,me,mp,wm,we,wp,am,ae,ap
    s.Init_List();
    s.Input_List();
    s.average(average);
    s.highest_grade();

    s.query_grade();
    s.amend_grade();
    s.delet_student();
    s.add_student();
    getchar();
}
```

```
男生高数平均：79.8
男生英语平均：81.4
男生物理平均：77.2
女生高数平均：77.1667
女生英语平均：86
女生物理平均：82.3333
高数平均：78.4833
英语平均：83.7
物理平均：79.7667
高数成绩最好：3 李三 成绩：90
英语成绩最好：8 李八 成绩：90
物理成绩最好：5 李五 成绩：95
请输入要查询的学号：3
学号：3
姓名：李三
性别：男
年龄：18
数学成绩：90
英语成绩：87
物理成绩：69
请输入要修改成绩的学号：3
您将修改李三的成绩
请输入要修改成绩的科目：1.高数,2.英语,3.物理
2
请输入要修改的成绩：67
修改后的成绩:数学:90 英语：67 物理：69
请输入要删除的学号：4
请输入要添加的信息：学号 姓名 性别 年龄 高数
12
```

图 2 运行程序后的执行结果

五、一些心得

1. 使用 C++的过程中熟悉了类的封装与使用

2. 在编码过程中使用到了容器，并使用了数据流处理，使用了 `stringstream` 进行 `string` 与其它数据类型之间进行转换

3. 学会了对单链表进行熟练度增删改查操作

4. 使用 `VSCode+mingw-w64` 进行编码配置，了解了较多 `C++`编码的流程，掌握了 `shell` 的使用

5. 使用 `Git+GitHub` 进行版本控制，掌握了 `Git` 的简单使用

6. 由于 `shell` 输入无法支持中文，暂不知如何修改，添加信息时无法写入中文

GitHub 代码链接：`https://github.com/MY-stone/data_structure`

笔记链接：

`http://note.youdao.com/noteshare?id=cb8aabd4352b5f422bc7ea1e3cf41a61`