

实 验 报 告

(与程序设计有关)

课程名称： 数据结构

实验题目： 查找

班级学号： 2203050320

姓 名： 闻家尉

成 绩：

沈 阳 理 工 大 学

2023 年 11 月 20 日

实验目的及要求：

1. 掌握线性表查找的基本思想和实现方法；
2. 掌握线性表分块查找的算法。

软硬件环境： window 11, sublime text 4

算法或原理分析（实验内容）：

1. 问题描述

利用分块查找算法在线性表（学生情况表）list 中查找给定值 key（学号）的结点，并将该结点的部分数据进行修改。

2. 基本要求

输入学号、选择课程名、输入修改后的成绩，在数据文件 score.txt 中查找相应的记录进行修改，将修改后的线性表（学生情况表）数据输出到文件 score.txt 中。

3. 算法提示

①分块查找的查找过程就分事下两步进行：

先在线性表中确定待查找的结点属于哪一块，由于块与块之间按关键字大小有序，因此，块间查找可采用二分查找算法；

②在所确定的块内查找待查找的结点即可无序也可有序，因此，块内查找一般可采用顺序查找算法。找到指定结点后，按要求修改结点中的有关数据。

程序代码或实现过程:

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>

using namespace std;

struct Student {
    int id;
    string course;
    double grade;
};

int sign;
vector<Student> readStudentsFromFile(const string& filename) {

    vector<Student> students;
    ifstream file(filename);

    if (file.is_open()) {
        string line;
        while (getline(file, line)) {
            istringstream iss(line);
            Student student;
            iss >> student.id >> student.course >> student.grade;
            students.push_back(student);
        }
        file.close();
    }

    return students;
}

void writeStudentsToFile(const string& filename, const vector<Student>& students) {
    ofstream file(filename);

    if (file.is_open()) {
        for (const auto& student : students) {
            file << student.id << " " << student.course << " " << student.grade << endl;
        }
        file.close();
    }
}
```

```

}

void updateGrade(vector<Student>& students, int id, const string& course, double
newGrade) {

    for (auto& student : students) {
        if (student.id == id && student.course == course) {
            student.grade = newGrade;
            sign = 1;
            break; // Assuming each student has a unique (id, course) pair
        }
    }

    if(!sign){
        cout<<"文件不存在."<<endl;
    }else cout << "找到目标文件" << endl;
}

int main() {

    string filename = "score.txt";
    vector<Student> students = readStudentsFromFile(filename);
    cout << "请输出查询信息" << endl;

    int searchId ;
    cin>>searchId;
    string searchCourse;
    cin>>searchCourse;
    double newGrade = 0;
    cin>>newGrade;

    updateGrade(students, searchId, searchCourse, newGrade);

    writeStudentsToFile(filename, students);

    return 0;
}

```



结果分析：



教师签字	杨书彤	日期	
------	-----	----	--