

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ЭЛЕКТРОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА «ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
АВТОМАТИЗИРОВАННЫХ СИСТЕМ»

ОТЧЁТ

«ЛАБОРАТОРНАЯ №8: ОБРАБОТКА ФАЙЛА»

Дисциплина: «Программирование»

Выполнил:

Студент группы ИВТ-21-26

Безух Владимир Сергеевич

Проверил:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь, 2022

Содержание

1.	Постановка задачи	3
2.	Анализ задачи.....	4
3.	Описание переменных.....	5
4.	Исходный код.....	6
5.	Анализ результатов.....	9

1. Постановка задачи

Дана структура «Студент»:

- фамилия, имя, отчество;
- дата рождения;
- домашний адрес;
- рейтинг.

Нужно удалить элементы, у которых даты рождения совпадают. Затем добавить элемент перед элементом с заданной фамилией.

2. Анализ задачи

Для удаления одинаковых элементов по дате рождения нужно:

- отсортировать элементы по полю даты рождения;
- в отсортированном виде найти элементы с датой рождения в единственном экземпляре.

Добавить новый элемент по ключу — тривиальная задача.

3. Описание переменных

std::vector<Student> students; — хранение записей о студентах.

std::ifstream f_input("FileInput.txt"); — для использования файла с
ВХОДНЫМИ ДАННЫМИ.

std::ofstream f_output("FileOutput.txt"); — для использования файла с
ВЫХОДНЫМИ ДАННЫМИ.

bool flag_new_group = true; — проверка начала новой группы элементов
по дате рождения.

std::vector<Student> result_students; — выходные данные.

4. Исходный код

```
#include <vector>
#include <string>
#include <fstream>
#include <iostream>
#include <iterator>
#include <algorithm>

struct FullName {
    std::string forename;
    std::string surname;
    std::string patronym;

    bool operator==(const FullName& right) const
    {
        return (forename == right.forename) &&
               (surname == right.surname) &&
               (patronym == right.patronym);
    }
};

struct Date {
    unsigned int day;
    unsigned int month;
    unsigned int year;

    bool operator==(const Date& right) const
    {
        return (day == right.day) &&
               (month == right.month) &&
               (year == right.year);
    }

    bool operator!=(const Date& right) const {
        return !(*this == right);
    }

    bool operator<(const Date& right) const {
        if (year < right.year) return true;
        else if (year == right.year)
        {
            if (month < right.month) return true;
            else if (month == right.month) return day < right.day;
            else return false;
        }
        else return false;
    }
};

struct Address {
    std::string city;
    std::string street_name;
    std::string house_number;

    bool operator==(const Address& right) const
    {
        return (city == right.city) &&
               (street_name == right.street_name) &&
               (house_number == right.house_number);
    }
};
```

```

struct Student {
    FullName personal_name;
    Date date_of_birth;
    Address personal_address;
    int personal_rating;

    bool operator==(const Student& right) const
    {
        return (personal_name == right.personal_name) &&
            (date_of_birth == right.date_of_birth) &&
            (personal_address == right.personal_address) &&
            (personal_rating == right.personal_rating);
    }

    friend std::istream& operator>>(std::istream& input, Student& student)
    {
        input >> student.personal_name.forename;
        input >> student.personal_name.surname;
        input >> student.personal_name.patronym;
        input >> student.date_of_birth.day;
        input >> student.date_of_birth.month;
        input >> student.date_of_birth.year;
        input >> student.personal_address.city;
        input >> student.personal_address.street_name;
        input >> student.personal_address.house_number;
        input >> student.personal_rating;

        return input;
    }

    friend std::ostream& operator<<(std::ostream& output, const Student& student)
    {
        output << student.personal_name.forename << ' ';
        output << student.personal_name.surname << ' ';
        output << student.personal_name.patronym << ' ';
        output << student.date_of_birth.day << ' ';
        output << student.date_of_birth.month << ' ';
        output << student.date_of_birth.year << ' ';
        output << student.personal_address.city << ' ';
        output << student.personal_address.street_name << ' ';
        output << student.personal_address.house_number << ' ';
        output << student.personal_rating << '\n';

        return output;
    }
};

```

```

int main()
{
    std::vector<Student> students;
    std::ifstream f_input("FileInput.txt");
    std::ofstream f_output("FileOutput.txt");

    while (!f_input.eof()) {
        Student student;
        f_input >> student;
        students.emplace_back(student);
    }

    sort(students.begin(), students.end(),
        [](const Student& left, const Student& right)
        { return left.date_of_birth < right.date_of_birth; });

    bool flag_new_group = true;
    std::vector<Student> result_students;
    for (auto prev = students.begin(); prev != students.end(); ++prev) {
        auto curr = prev; ++curr;
        if (curr == students.end()) {
            if (flag_new_group)
                result_students.emplace_back(*prev);

            break;
        }

        if (prev->date_of_birth != curr->date_of_birth) {
            if (flag_new_group)
                result_students.emplace_back(*prev);

            flag_new_group = true;
        } else
            flag_new_group = false;
    }

    std::string surname; std::cin >> surname;
    Student student; std::cin >> student;
    auto iterator = result_students.begin();
    for (; iterator->personal_name.surname != surname; ++iterator);
    result_students.emplace(iterator, student);

    for (auto& result_student : result_students)
        f_output << result_student;

    f_input.close();
    f_output.close();
}

```

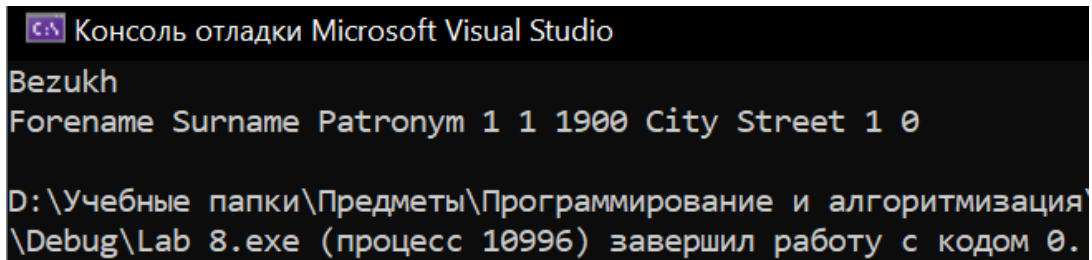

5. Анализ результатов

Исходный файл (рис. 1).

```
1 Vladimir Bezukh Sergeevich 22 2 2000 Perm Dedyukina 20 75
2 Ivan Ivanov Ivanovich 1 1 1990 Moscow Tverskaya 1 25
3 Vasiliy Vasiliev Vasilievich 1 1 1990 Petersburg Komarovo 1 50
4 Dmitriy Shostakovich Dmitriyevich 25 9 1906 Moscow Lubyanka 1 100
5 Vasia Pupkin Ivanovich 10 10 2010 Saratov Sadovaya 42 -50
6 Sergei Sergeev Sergeevich 1 1 1990 Vladivostok Lugovaya 20 25
```

Рисунок 1 — Студенты

Вставка элемента по ключу (рис. 2).



```
Консоль отладки Microsoft Visual Studio
Bezukh
Forename Surname Patronym 1 1 1900 City Street 1 0
D:\Учебные папки\Предметы\Программирование и алгоритмизация\
Debug\Lab 8.exe (процесс 10996) завершил работу с кодом 0.
```

Рисунок 2 — Вставка нового студента

Результирующий файл (рис. 3).

```
1 Dmitriy Shostakovich Dmitriyevich 25 9 1906 Moscow Lubyanka 1 100
2 Forename Surname Patronym 1 1 1900 City Street 1 0
3 Vladimir Bezukh Sergeevich 22 2 2000 Perm Dedyukina 20 75
4 Vasia Pupkin Ivanovich 10 10 2010 Saratov Sadovaya 42 -50
```

Рисунок 3 — Результат после удаления трёх студентов и вставки нового