ПРАКТИЧЕСКАЯ РАБОТА 3

Тема: Файлы и классы

студента 3ИСП-2

Эбулисова Эмиля

Ход работы:

Вариант 25

Задание 14-5: Следующие задания требуется решить с использованием классов. При этом обязательно оформить методы для выполнения каждого из действий: по вводу данных, выводу их в файл, чтению данных из файла и выводу их на экран, сортировке данных.

Дана структура с именем AEROFLOT, состоящая из полей:

 название пункта назначения рейса;

 номер рейса;

 тип самолета.

Написать программу, которая выполняет следующие действия:

 ввод с клавиатуры данных в массив, состоящий из 7 элементов типа

AEROFLOT, и занесение их в файл данных;

 чтение данных из файла и вывод их на экран;

 вывод на экран пунктов назначения и номеров рейсов, обслужи-

ваемых самолетом, тип которого введен с клавиатуры (если таких нет –

вывести об этом сообщение);

 список должен быть упорядочен по алфавиту названий пунктов

назначения.

#include <iostream>

#include <fstream>

#include <string>

#include <algorithm>

using namespace std;

struct AEROFLOT {

    string destination;

    int flightNumber;

    string aircraftType;

};

bool compareByDestination(const AEROFLOT& a, const AEROFLOT& b) {

    return a.destination < b.destination;

}

int main() {

    const int numFlights = 7;

    AEROFLOT flights[numFlights];

    // Ввод данных с клавиатуры и запись в файл

    ofstream outFile("flights.txt");

    for (int i = 0; i < numFlights; ++i) {

        cout << "Введите название пункта назначения, номер рейса и тип самолета (через пробел): ";

        cin >> flights[i].destination >> flights[i].flightNumber >> flights[i].aircraftType;

        outFile << flights[i].destination << ' ' << flights[i].flightNumber << ' ' << flights[i].aircraftType << endl;

    }

    outFile.close();

    // Чтение данных из файла и вывод на экран

    ifstream inFile("flights.txt");

    AEROFLOT readFlights[numFlights];

    for (int i = 0; i < numFlights; ++i) {

        inFile >> readFlights[i].destination >> readFlights[i].flightNumber >> readFlights[i].aircraftType;

        cout << "Рейс " << readFlights[i].flightNumber << " в " << readFlights[i].destination << " (тип самолета: " << readFlights[i].aircraftType << ")" << endl;

    }

    inFile.close();

    // Вывод пунктов назначения и номеров рейсов по типу самолета

    string searchType;

    cout << "Введите тип самолета для поиска: ";

    cin >> searchType;

    bool found = false;

    sort(readFlights, readFlights + numFlights, compareByDestination);

    for (int i = 0; i < numFlights; ++i) {

        if (readFlights[i].aircraftType == searchType) {

            cout << "Рейс " << readFlights[i].flightNumber << " в " << readFlights[i].destination << endl;

            found = true;

        }

    }

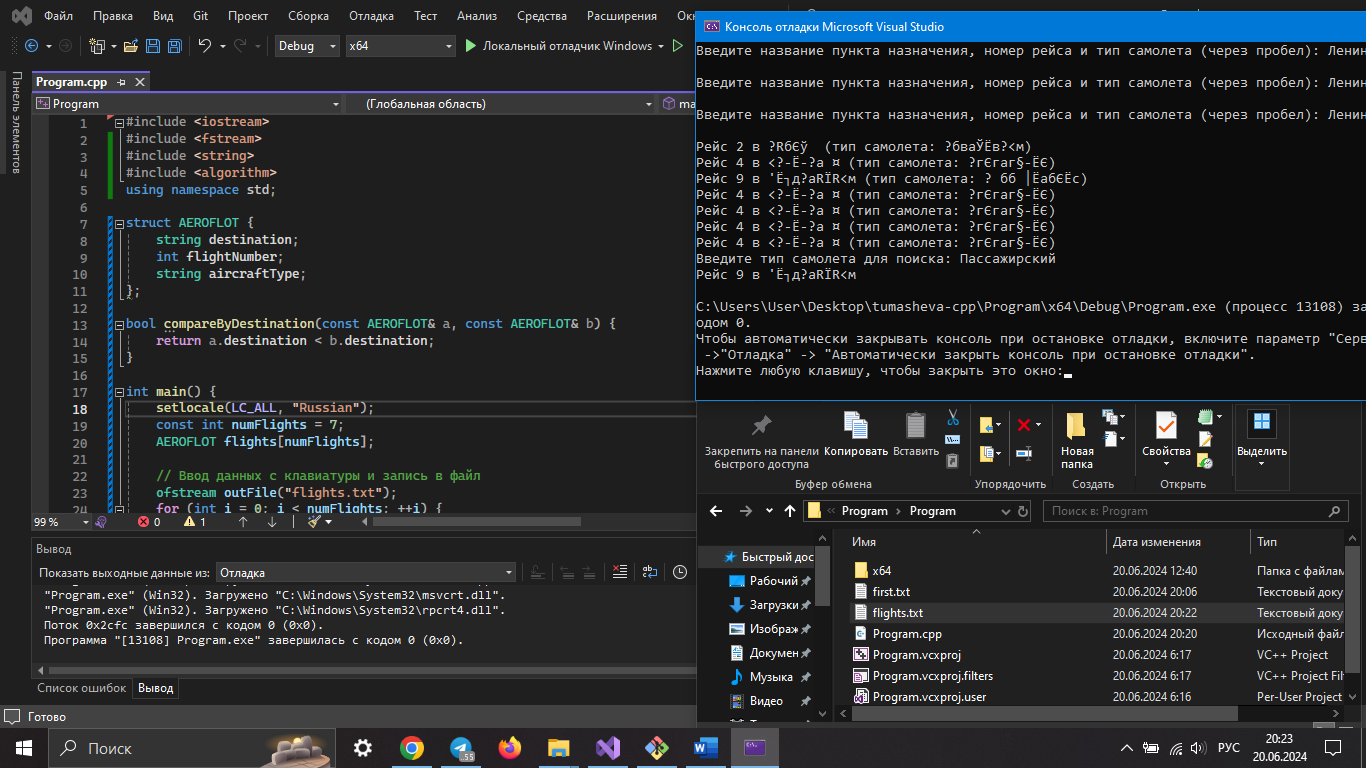
    if (!found) {

        cout << "Самолетов с таким типом не найдено." << endl;

    }

    return 0;

}



Задание 14-11. Дана структура с именем MARSH, состоящая из полей:

179

 название начального пункта назначения;

 название конечного пункта назначения;

 номер маршрута.

Написать программу, которая выполняет следующие действия:

 ввод с клавиатуры данных в массив, состоящий из 8 элементов типа

MARSH, и занесение их в файл данных;

 чтение данных из файла и вывод их на экран;

 вывод на экран информации о маршрутах, которые начинаются или

заканчиваются в пункте, название которого введено с клавиатуры (если та-

ких нет – вывести об этом сообщение);

 список должен быть упорядочен по номерам маршрутов.

#include <iostream>

#include <fstream>

#include <string>

#include <algorithm>

using namespace std;

struct MARSH {

    string startDestination;

    string endDestination;

    int routeNumber;

};

bool compareByRouteNumber(const MARSH& a, const MARSH& b) {

    return a.routeNumber < b.routeNumber;

}

int main() {

    const int numRoutes = 8;

    MARSH routes[numRoutes];

    // Ввод данных с клавиатуры и запись в файл

    ofstream outFile("routes.txt");

    for (int i = 0; i < numRoutes; ++i) {

        cout << "Введите начальный пункт, конечный пункт и номер маршрута (через пробел): ";

        cin >> routes[i].startDestination >> routes[i].endDestination >> routes[i].routeNumber;

        outFile << routes[i].startDestination << ' ' << routes[i].endDestination << ' ' << routes[i].routeNumber << endl;

    }

    outFile.close();

    // Чтение данных из файла и вывод на экран

    ifstream inFile("routes.txt");

    MARSH readRoutes[numRoutes];

    for (int i = 0; i < numRoutes; ++i) {

        inFile >> readRoutes[i].startDestination >> readRoutes[i].endDestination >> readRoutes[i].routeNumber;

        cout << "Маршрут " << readRoutes[i].routeNumber << ": " << readRoutes[i].startDestination << " - " << readRoutes[i].endDestination << endl;

    }

    inFile.close();

    // Вывод информации о маршрутах по введенному пункту

    string searchDestination;

    cout << "Введите название пункта для поиска: ";

    cin >> searchDestination;

    bool found = false;

    sort(readRoutes, readRoutes + numRoutes, compareByRouteNumber);

    for (int i = 0; i < numRoutes; ++i) {

        if (readRoutes[i].startDestination == searchDestination || readRoutes[i].endDestination == searchDestination) {

            cout << "Маршрут " << readRoutes[i].routeNumber << ": " << readRoutes[i].startDestination << " - " << readRoutes[i].endDestination << endl;

            found = true;

        }

    }

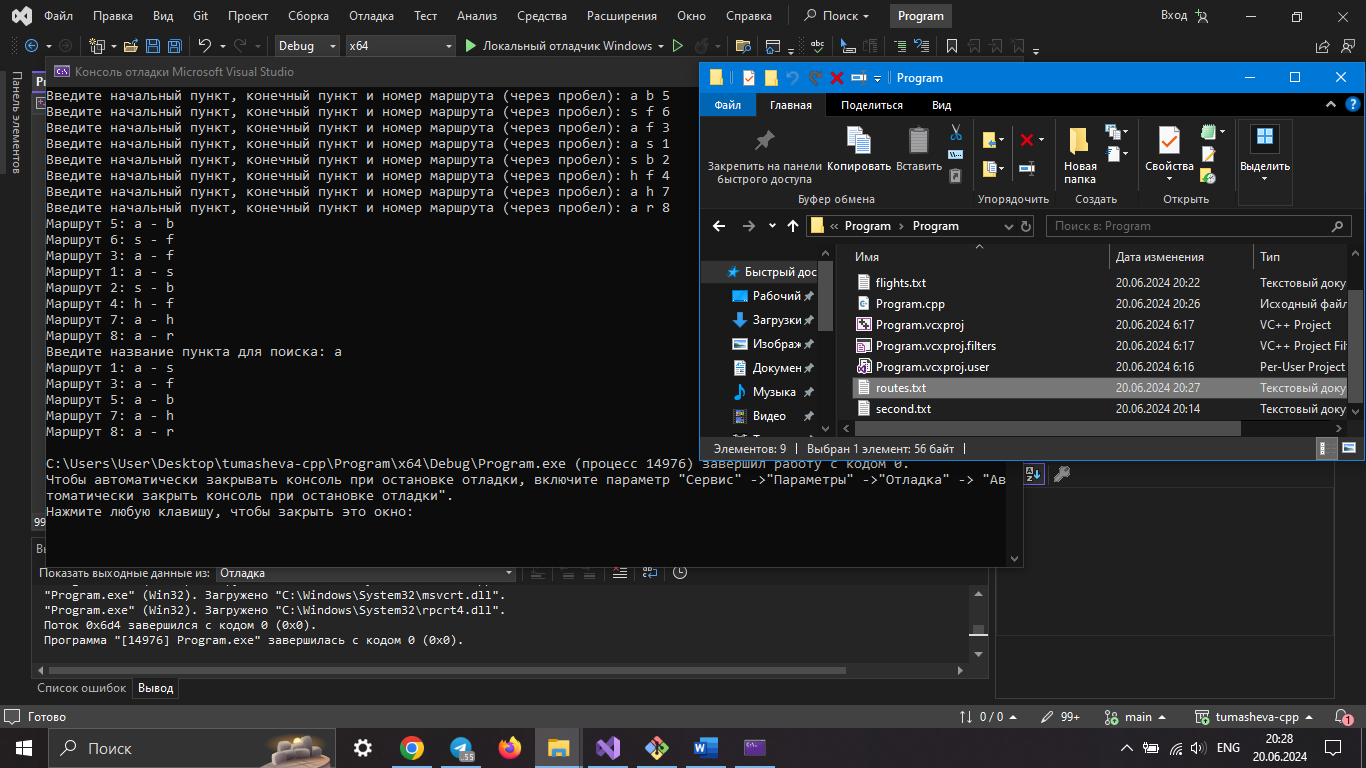
    if (!found) {

        cout << "Маршрутов с таким пунктом не найдено." << endl;

    }

    return 0;

}



Задание 14-25. Дана структура с именем SOTRUD, состоящая из полей:

 фамилия, имя сотрудника;

 должность;

 год рождения;

 год поступления на работу.

Написать программу, которая выполняет следующие действия:

 ввод с клавиатуры данных в массив, состоящий из 8 элементов типа

SOTRUD, и занесение их в файл данных;

 чтение данных из файла и вывод их на экран;

 вывод на экран сведений сотрудников, которые имеют стаж рабо-

ты, больший введенного с клавиатуры (если таких нет – вывести об этом

сообщение);

 список должен быть упорядочен по алфавиту фамилий.

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <algorithm>

struct SOTRUD {

std::string surname;

std::string name;

std::string position;

int birth\_year;

int hire\_year;

};

bool compareBySurname(const SOTRUD& a, const SOTRUD& b) {

return a.surname < b.surname;

}

int main() {

setlocale(LC\_ALL, "Russian");

const int numEmployees = 3;

std::vector<SOTRUD> employees(numEmployees);

// Ввод данных с клавиатуры

for (int i = 0; i < numEmployees; ++i) {

std::cout << "Введите данные для сотрудника " << i + 1 << ":\n";

std::cout << "Фамилия: ";

std::cin >> employees[i].surname;

std::cout << "Имя: ";

std::cin >> employees[i].name;

std::cout << "Должность: ";

std::cin >> employees[i].position;

std::cout << "Год рождения: ";

std::cin >> employees[i].birth\_year;

std::cout << "Год поступления на работу: ";

std::cin >> employees[i].hire\_year;

}

// Запись данных в файл

std::ofstream outFile("employees.txt");

for (const auto& emp : employees) {

outFile << emp.surname << " " << emp.name << " " << emp.position << " "

<< emp.birth\_year << " " << emp.hire\_year << "\n";

}

outFile.close();

// Чтение данных из файла и вывод на экран

std::ifstream inFile("employees.txt");

std::cout << "\nДанные сотрудников из файла:\n";

while (inFile >> employees[numEmployees - 1].surname >> employees[numEmployees - 1].name

>> employees[numEmployees - 1].position >> employees[numEmployees - 1].birth\_year

>> employees[numEmployees - 1].hire\_year) {

std::cout << employees[numEmployees - 1].surname << " " << employees[numEmployees - 1].name << " "

<< employees[numEmployees - 1].position << " " << employees[numEmployees - 1].birth\_year << " "

<< employees[numEmployees - 1].hire\_year << "\n";

}

inFile.close();

// Вывод сведений о сотрудниках с большим стажем

int inputYear;

std::cout << "\nВведите год для фильтрации сотрудников: ";

std::cin >> inputYear;

std::cout << "\nСотрудники с большим стажем (поступили после " << inputYear << " года):\n";

std::sort(employees.begin(), employees.end(), compareBySurname);

for (const auto& emp : employees) {

if (emp.hire\_year > inputYear) {

std::cout << emp.surname << " " << emp.name << " " << emp.position << "\n";

}

}

return 0;

}

