



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение высшего  
образования  
**"МИРЭА - Российский технологический университет"**

**РТУ МИРЭА**

---

Институт информационных технологий (ИТ)  
Кафедра математического обеспечения и стандартизации информационных  
технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1-2**  
**по дисциплине**  
**«Структуры и алгоритмы обработки данных»**

Тема. Разработка и программная реализация  
задач, использующих внешние структуры данных - файлы  
(текстовые и двоичные)

Выполнил студент группы ИКБО-25-22

Ракитин В.А.

Принял доцент

Бузыкова Ю.С.

Москва 2023

**Цель:** получить навыки применения файловых потоков языка C++ (или файлов языка Си) по управлению текстовым и двоичным файлами

## **Ход работы**

### **Задание 1**

#### **1. Условие задания**

##### **1.1 Формулировка задачи**

Разработать программу, управления текстовым файлом.

##### **1.2 Требования**

Разработать функции для выполнения операций над текстовым файлом.

- создание текстового файла средствами текстового редактора кодировки ASCII, содержащего десятичные числа по несколько чисел на строке;
- вывод содержимого текстового файла;
- добавление новой записи в конец файла;
- прочитав значение числа, указав его порядковый номер в файле, и вернуть его значение;
- определить количество чисел в файле.

Контроль открытия и существования файла выполнить в основной программе перед вызовом функции. Перед закрытием файла, проверить отсутствие ошибок ввода и вывода (метод good).

##### **1.3 Персональный вариант**

Создать новый файл из значений исходного. Значениями строк нового файла будут суммы каждого десятка чисел исходного, с указанием перед суммой ее порядкового номера.

#### **2. Тестовый пример**

Копия содержания текстового файла на примере 49 записей представлена на рисунке 1.1.

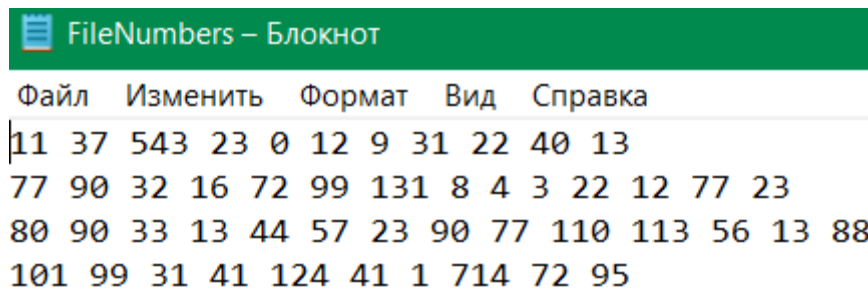


Рисунок 1.1 – Содержание текстового файла

### 3. Реализация программы

#### 3.1 Функционал приложения

Код функции, реализующей создание текстового файла, представлен в листинге 1.1.

Листинг 1.1 – Код реализации функции создания файла

```
void createFile() {  
    ofstream file("FileNumbers.txt");  
    file << "11 37 543 23 0 12 9 31 22 40 13" << endl;  
    file << "77 90 32 16 72 99 131 8 4 3 22 12 77 23" << endl;  
    file << "80 90 33 13 44 57 23 90 77 110 113 56 13 88" << endl;  
    file << "101 99 31 41 124 41 1 714 72 95" << endl;  
    if (file.good()) {  
        file.close();  
        cout << "Файл успешно создан" << endl;  
    }  
    else exit(1);  
}
```

Код функции, реализующей вывод содержимого текстового файла, представлен в листинге 1.2.

Листинг 1.2 – Код реализации функции вывода файла

```
void printFile(const string filename) {  
    ifstream file(filename);  
    string line;  
    while (getline(file, line)) {  
        cout << line << endl;  
    }  
    file.clear();  
    if (file.good()) file.close();  
    else exit(2);  
}
```

Код функции, реализующей добавление новой записи в конец файла, представлен в листинге 1.3.

Листинг 1.3 – Код реализации функции добавления новой записи в конец файла

```
void addRecord() {  
    ofstream file("FileNumbers.txt", ios::app);  
    string record;
```

```

cout << "Введите новую запись (десятичные числа через пробел): ";
cin.ignore();
getline(cin, record);
file << record << endl;
if (file.good()) {
    cout << "Запись успешно добавлена" << endl;
    file.close();
}
else exit(3);
}

```

Код функции, реализующей вывод значения числа по указанному порядковому номеру, представлен в листинге 1.4.

Листинг 1.4 – Код реализации функции вывода значения числа по номеру

```

void readNumber(int index) {
    ifstream file("FileNumbers.txt");
    int number;
    string line;
    int count = 0;
    bool flag = false;
    while (getline(file, line)) {
        stringstream ss(line);
        while (ss >> number) {
            count++;
            if (count == index) {
                cout << "Значение числа с порядковым номером " << index <<
": " << number << endl;
                return;
            }
        }
    }
    cout << "Число с порядковым номером " << index << " не найдено" << endl;
    file.clear();
    if (file.good()) file.close();
    else exit(4);
}

```

Код функции, реализующей определение количества чисел в файле, представлен в листинге 1.5.

Листинг 1.5 – Код реализации функции определения количества чисел в файле

```

void countNumbers() {
    ifstream file("FileNumbers.txt");
    int number;
    int count = 0;
    while (file >> number) {
        count++;
    }
    cout << "Количество чисел в файле: " << count << endl;
    file.clear();
    if (file.good()) file.close();
    else exit(5);
}

```

Код функции, реализующей создание нового файла из значений исходного, в котором значениями являются суммы каждого десятка чисел

исходного, с указанием перед суммой ее порядкового номера (персональный вариант), представлен в листинге 1.6.

#### Листинг 1.6 – Код реализации функции нового файла

```
void createNewFile() {
    ifstream inputFile("FileNumbers.txt"); // Открытие исходного файла для
    чтения
    ofstream outputFile("NewFileNumbers.txt"); // Открытие нового файла для
    записи

    vector<int> numbers;
    int number;
    int lineNumber = 1;
    int sum = 0;

    while (inputFile >> number) {
        numbers.push_back(number);

        if (numbers.size() == 10) {
            for (int n : numbers) {
                sum += n;
            }

            outputFile << "Сумма " << lineNumber << "-го десятка чисел: " <<
sum << std::endl;

            numbers.clear();
            sum = 0;
            lineNumber++;
        }
    }

    if (!numbers.empty()) {
        for (int n : numbers) {
            sum += n;
        }

        outputFile << "Сумма " << lineNumber << "-го десятка чисел: " << sum
<< std::endl;
    }
    inputFile.clear();
    outputFile.clear();
    if (inputFile.good()) inputFile.close();
    else exit(6);
    if (outputFile.good()) {
        outputFile.close();
        cout << "Новый файл создан успешно." << endl;
    }
    else exit(7);
}
```

### 3.2 Код основной программы

Код основной программы представлен в листинге 1.7.

#### Листинг 1.7 – Код основной программы

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

```

#include <sstream>

using namespace std;

// Создание текстового файла с десятичными числами
void createFile() {
    ofstream file("FileNumbers.txt");
    file << "11 37 543 23 0 12 9 31 22 40 13" << endl;
    file << "77 90 32 16 72 99 131 8 4 3 22 12 77 23" << endl;
    file << "80 90 33 13 44 57 23 90 77 110 113 56 13 88" << endl;
    file << "101 99 31 41 124 41 1 714 72 95" << endl;
    if (file.good()) {
        file.close();
        cout << "Файл успешно создан" << endl;
    }
    else exit(1);
}

// Вывод содержимого текстового файла
void printFile(const string filename) {
    ifstream file(filename);
    string line;
    while (getline(file, line)) {
        cout << line << endl;
    }
    file.clear();
    if (file.good()) file.close();
    else exit(2);
}

// Добавление новой записи в конец файла
void addRecord() {
    ofstream file("FileNumbers.txt", ios::app);
    string record;
    cout << "Введите новую запись (десятичные числа через пробел): ";
    cin.ignore();
    getline(cin, record);
    file << record << endl;
    if (file.good()) {
        cout << "Запись успешно добавлена" << endl;
        file.close();
    }
    else exit(3);
}

// Чтение значения числа по его порядковому номеру в файле
void readNumber(int index) {
    ifstream file("FileNumbers.txt");
    int number;
    string line;
    int count = 0;
    bool flag = false;
    while (getline(file, line)) {
        stringstream ss(line);
        while (ss >> number) {
            count++;
            if (count == index) {
                cout << "Значение числа с порядковым номером " << index <<
": " << number << endl;
                return;
            }
        }
    }
}

```

```

        cout << "Число с порядковым номером " << index << " не найдено" << endl;
        file.clear();
        if (file.good()) file.close();
        else exit(4);
    }

    //Определение количества чисел в файле
    void countNumbers() {
        ifstream file("FileNumbers.txt");
        int number;
        int count = 0;
        while (file >> number) {
            count++;
        }
        cout << "Количество чисел в файле: " << count << endl;
        file.clear();
        if (file.good()) file.close();
        else exit(5);
    }

    void createNewFile() {
        ifstream inputFile("FileNumbers.txt"); // Открытие исходного файла для
        чтения
        ofstream outputFile("NewFileNumbers.txt"); // Открытие нового файла для
        записи

        vector<int> numbers;
        int number;
        int lineNumber = 1;
        int sum = 0;

        while (inputFile >> number) {
            numbers.push_back(number);

            if (numbers.size() == 10) {
                for (int n : numbers) {
                    sum += n;
                }

                outputFile << "Сумма " << lineNumber << "-го десятка чисел: " <<
sum << std::endl;

                numbers.clear();
                sum = 0;
                lineNumber++;
            }
        }

        if (!numbers.empty()) {
            for (int n : numbers) {
                sum += n;
            }

            outputFile << "Сумма " << lineNumber << "-го десятка чисел: " << sum
<< std::endl;
        }
        inputFile.clear();
        outputFile.clear();
        if (inputFile.good()) inputFile.close();
        else exit(6);
        if (outputFile.good()) {

```

```

        outputFile.close();
        cout << "Новый файл создан успешно." << endl;
    }
    else exit(7);
}

int main() {
    setlocale(LC_ALL, "Rus");
    int choise;
    cout << "1 - Создание файла" << endl
        << "2 - Вывод файла" << endl
        << "3 - Добавить запись в конец файла" << endl
        << "4 - Вывести число по индексу" << endl
        << "5 - Вывести количество чисел в файле" << endl
        << "6 - Создать новый файл, где значениями строк будут суммы каждого
десятька чисел исходного, с указанием перед суммой ее порядкового номера" <<
endl
        << "7 - Вывод нового файла" << endl
        << "0 - Завершить работу программы" << endl << endl;
    while (true) {
        cout << "Выберите действие: 1, 2, 3, 4, 5, 6, 7, 0: ";
        cin >> choise;
        switch (choise)
        {
            case 1:
                createFile();
                break;
            case 2:
                printFile("FileNumbers.txt");
                break;
            case 3:
                addRecord();
                break;
            case 4:
                int index;
                cout << "Введите индекс числа: ";
                cin >> index;
                readNumber(index);
                break;
            case 5:
                countNumbers();
                break;
            case 6:
                createNewFile();
                break;
            case 7:
                printFile("NewFileNumbers.txt");
                break;
            case 0:
                return 0;
            default:
                cout << "Неверный ввод" << endl;
                break;
        }
    }
}

```

#### 4. Результаты тестирования

Результат тестирования программы представлен на рисунке 1.2.



```

1 - Создание файла
2 - Вывод файла
3 - Добавить запись в конец файла
4 - Вывести число по индексу
5 - Вывести количество чисел в файле
6 - Создать новый файл, где значениями строк будут суммы каждого десятка чисел исходного, с указанием перед суммой ее порядкового номера
7 - Вывод нового файла
0 - Завершить работу программы

Выберите действие: 1, 2, 3, 4, 5, 6, 7, 0: 1
Файл успешно создан
Выберите действие: 1, 2, 3, 4, 5, 6, 7, 0: 2
11 37 543 23 0 12 9 31 22 40 13
77 90 32 16 72 99 131 8 4 3 22 12 77 23
80 90 33 13 44 57 23 90 77 110 113 56 13 88
101 99 31 41 124 41 1 714 72 95
Выберите действие: 1, 2, 3, 4, 5, 6, 7, 0: 3
Введите новую запись (десятичные числа через пробел): 12 23 34 45 56 67 78 89
Запись успешно добавлена
Выберите действие: 1, 2, 3, 4, 5, 6, 7, 0: 2
11 37 543 23 0 12 9 31 22 40 13
77 90 32 16 72 99 131 8 4 3 22 12 77 23
80 90 33 13 44 57 23 90 77 110 113 56 13 88
101 99 31 41 124 41 1 714 72 95
12 23 34 45 56 67 78 89
Выберите действие: 1, 2, 3, 4, 5, 6, 7, 0: 4
Введите индекс числа: 13
Значение числа с порядковым номером 13: 90
Выберите действие: 1, 2, 3, 4, 5, 6, 7, 0: 5
Количество чисел в файле: 57
Выберите действие: 1, 2, 3, 4, 5, 6, 7, 0: 6
Новый файл создан успешно.
Выберите действие: 1, 2, 3, 4, 5, 6, 7, 0: 7
Сумма 1-го десятка чисел: 728
Сумма 2-го десятка чисел: 542
Сумма 3-го десятка чисел: 397
Сумма 4-го десятка чисел: 728
Сумма 5-го десятка чисел: 1230
Сумма 6-го десятка чисел: 392
Выберите действие: 1, 2, 3, 4, 5, 6, 7, 0: 0

```

Рисунок 1.2 – Результат тестирования программы

## Задание 2

### 1. Условие задания

#### 1.1 Формулировка задачи

Разработать программу управление двоичными файлами с записями фиксированной длины

#### 1.2 Требования

Файл состоит из записей определенной структуры, согласно варианту. Записи имеют ключ, уникальный в пределах файла.

Разработать функции для выполнения операций:

- преобразование тестовых данных из текстового файла в двоичный файл;
- сохранение данных двоичного файла в текстовом, так, чтобы используя их можно было восстановить двоичный файл;
- вывод всех записей двоичного файла;

- доступ к записи по ее порядковому номеру в файле, используя механизм прямого доступа к записи в двоичном файле;
- удаление записи с заданным значением ключа, выполнить путем замены на последнюю запись.
- манипулирование записями в двоичном файле согласно дополнительным операциям, определенным в варианте.

### 1.3 Персональный вариант

Аэропорт (табло прибытия пассажирских авиарейсов сохраняет все данные в файле): пункт вылета, номер рейса, дата прилета, время прилета, информация о задержке прилета в часах.

- Сформировать в новом двоичном файле информацию по рейсам, задержанным в указанную дату.
- Изменить время прилета для заданного рейса на новое значение.

## 2. Тестовый пример

Копия содержания текстового файла на примере 7 записей для преобразования в двоичный файл представлена на рисунке 2.1.

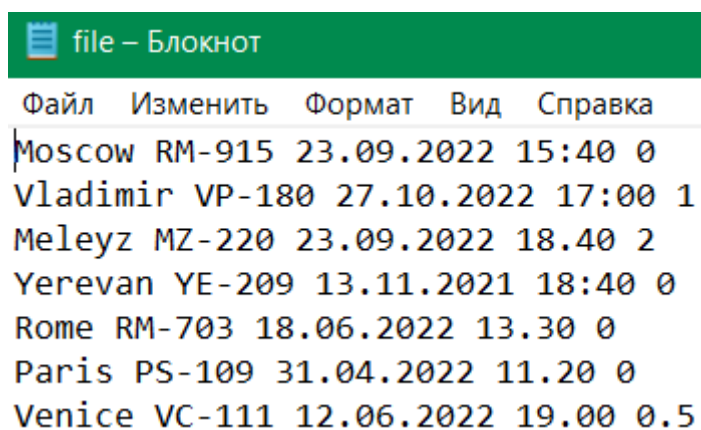


Рисунок 2.1 – Содержимое текстового файла

Содержание двоичного файла (результат операции вывода двоичного файла на экран), полученного из данных текстового файла представлено на рисунке 2.2.

№	Пункт прилета	Номер рейса	Дата прилета	Время прилета	Задержка в часах
1	Moscow	RM-915	23.09.2022	15:40	0
2	Vladimir	VP-180	27.10.2022	17:00	1
3	Meleyz	MZ-220	23.09.2022	18.40	2
4	Yerevan	YE-209	13.11.2021	18:40	0
5	Rome	RM-703	18.06.2022	13.30	0
6	Paris	PS-109	31.04.2022	11.20	0
7	Venice	VC-111	12.06.2022	19.00	0.5

Рисунок 2.2 – Содержимое бинарного файла

### 3. Реализация программы

#### 3.1 Функционал приложения

Код функции, реализующей преобразование тестовых данных из текстового файла в двоичный файл, представлен в листинге 2.1.

Листинг 2.1 – Код реализации функции преобразования тестовых данных из текстового файла в двоичный файл

```
void createBinFile() { //Создание бинарного файла
    ifstream fileTXT("file.txt");
    ofstream fileBIN("file.bin");
    if (!fileTXT.is_open()) {
        cout << "Ошибка при открытии текстового файла" << endl;
        return;
    }
    if (!fileBIN.is_open()) {
        cout << "Ошибка при открытии бинарного файла" << endl;
        return;
    }
    Airport airport;
    while (!fileTXT.eof()) {
        fileTXT >> airport.arrivalPoint;
        fileTXT >> airport.flightNumber;
        fileTXT >> airport.arrivalDate;
        fileTXT >> airport.arrivalTime;
        fileTXT >> airport.flightDelay;
        fileBIN.write((char*)& airport, sizeof(Airport));
    }
    fileBIN.close();
    fileTXT.close();
    cout << "Данные из файла txt успешно добавлена в файл bin" << endl;
}
```

Код функции, реализующей сохранение данных двоичного файла в текстовом, представлен в листинге 2.2.

Листинг 2.2 – Код реализации функции сохранения данных двоичного файла в текстовом

```
void saveBinInNewTxt() { //Извлечь данные из бинарного файла в новый
    текстовый файл
    ifstream fileBIN("file.bin");
    ofstream newFileTXT("newFile.txt");
    if (!fileBIN.is_open()) {
        cout << "Ошибка при открытии бинарного файла" << endl;
        return;
    }
    if (!newFileTXT.is_open()) {
        cout << "Ошибка при открытии текстового файла" << endl;
        return;
    }
}
```

```

    }
    Airport airport;
    fileBIN.read((char*)&airport, sizeof(Airport));
    while (!fileBIN.eof()) {
        newFileTXT << airport.arrivalPoint << " ";
        newFileTXT << airport.flightNumber << " ";
        newFileTXT << airport.arrivalDate << " ";
        newFileTXT << airport.arrivalTime << " ";
        newFileTXT << airport.flightDelay << "\n";
        fileBIN.read((char*)&airport, sizeof(Airport));
    }
    cout << "Данные из бинарного файла успешно добавлены в новый текстовый
файл" << endl;
    fileBIN.close();
    newFileTXT.close();
}

```

Код функции, реализующей вывод всех записей двоичного файла, представлен в листинге 2.3.

Листинг 2.3 – Код реализации функции вывода всех записей двоичного файла

```

void readBinFile() { //Вывод бинарного файла
    Airport airport;
    ifstream fileBIN("file.bin");
    if (!fileBIN.is_open()) {
        cout << "Ошибка при открытии бинарного файла" << endl;
        return;
    }
    int i = 1;
    fileBIN.read((char*)&airport, sizeof(Airport));
    cout << "Вывод бинарного файла" << endl;
    cout << "№" << "\t";
    cout << left << setw(20) << "Пункт прилета" << "\t";
    cout << left << setw(20) << "Номер рейса" << "\t";
    cout << left << setw(20) << "Дата прилета" << "\t";
    cout << left << setw(20) << "Время прилета" << "\t";
    cout << left << setw(20) << "Задержка в часах" << endl;
    while (!fileBIN.eof()) {
        cout << i << "\t";
        cout << left << setw(20) << airport.arrivalPoint << "\t";
        cout << left << setw(20) << airport.flightNumber << "\t";
        cout << left << setw(20) << airport.arrivalDate << "\t";
        cout << left << setw(20) << airport.arrivalTime << "\t";
        cout << left << setw(20) << airport.flightDelay << endl;
        fileBIN.read((char*)&airport, sizeof(Airport));
        i++;
    }
    fileBIN.close();
}

```

Код функции, реализующей доступ к записи по ее порядковому номеру в файле, используя механизм прямого доступа к записи в двоичном файле, представлен в листинге 2.4.

Листинг 2.4 – Код реализации функции доступа к записи по ее порядковому номеру в файле

```

void getByNumber() { //Получение данных о рейсе по её номеру
    ifstream fileBIN("file.bin");

```

```

        if (!fileBIN.is_open()) {
            cout << "Ошибка при открытии бинарного файла" << endl;
            return;
        }
        int num;
        cout << "Введите номер: "; cin >> num;
        Airport airport;
        fileBIN.seekg(sizeof(airport) * (num - 1), ios::cur);
        fileBIN.read((char*)&airport, sizeof(airport));
        cout << "Запись с номером " << num << ": " << endl
            << airport.arrivalPoint << " "
            << airport.flightNumber << " "
            << airport.arrivalDate << " "
            << airport.arrivalTime << " "
            << airport.flightDelay;
        fileBIN.close();
        cout << endl;
    }
}

```

Код функции, реализующей удаление записи с заданным значением ключа путем замены на последнюю запись, представлен в листинге 2.5.

**Листинг 2.5 – Код реализации функции удаления записи с заданным значением ключа путем замены на последнюю запись**

```

void deleteByKey() { //Удаление рейса по ключу, его замена на Tula TC-111
11.11.2011 11:11 1
    fstream fileBIN("file.bin");
    if (!fileBIN.is_open()) {
        cout << "Ошибка при открытии бинарного файла" << endl;
        return;
    }
    Airport airport, airportNew;
    string race;
    char arrivalPointNew[30], flightNumberNew[30], arrivalDateNew[30],
arrivalTimeNew[30], flightDelayNew[30];
    int i = 0;
    bool flag = false;
    cout << "Введите номер рейса, который необходимо заменить: "; cin >>
race;
    while (!fileBIN.eof()) {
        fileBIN.read((char*)&airport, sizeof(Airport));
        if (airport.flightNumber == race) {
            flag = true;
            cout << "Введите новый пункт прилета: "; cin >>
arrivalPointNew;
            cout << "Введите новый номер рейса:"; cin >>
flightNumberNew;
            cout << "Введите новую дату прилета:"; cin >>
arrivalDateNew;
            cout << "Введите новое время прилета:"; cin >>
arrivalTimeNew;
            cout << "Введите новую задержку прилета в часах:"; cin >>
flightDelayNew;

            fileBIN.seekg(sizeof(Airport) * i);
            fileBIN.read((char*)&airportNew, sizeof(Airport));
            for (int j = 0; j < 30; j++) {
                airportNew.arrivalPoint[j] = arrivalPointNew[j];
                airportNew.flightNumber[j] = flightNumberNew[j];
                airportNew.arrivalDate[j] = arrivalDateNew[j];
                airportNew.arrivalTime[j] = arrivalTimeNew[j];
            }
        }
        i++;
    }
}

```

```

        airportNew.flightDelay[j] = flightDelayNew[j];
    }

    fileBIN.seekp(sizeof(Airport) * i, ios::beg);
    fileBIN.write((char*)&airportNew, sizeof(Airport));
    cout << "Информация о рейсе изменена" << endl;
    break;
}
i++;
}
if (!flag) {
    cout << "Такой рейс не найден" << endl;
}
}

```

Код функции, реализующей создание нового бинарного файла с задержками по дате, представлен в листинге 2.6

Листинг 2.6 – Код реализации функции создания нового бинарного файла с задержками по дате

```

void createNewBinWitchDateDelays() { // Создание нового бинарного файла с
задержками по дате
    ifstream fileBIN("file.bin");
    ofstream newFileBIN("newFile.bin");
    if (!fileBIN.is_open()) {
        cout << "Ошибка при открытии бинарного файла" << endl;
        return;
    }
    if (!newFileBIN.is_open()) {
        cout << "Ошибка при открытии нового бинарного файла" << endl;
        return;
    }

    Airport airport;
    string date;
    cout << "Введите дату, по которой надо создать новый бинарный файл:
"; cin >> date;
    while (!fileBIN.eof()) {
        fileBIN.read((char*)&airport, sizeof(Airport));
        if (airport.arrivalDate == date) {
            newFileBIN << airport.arrivalPoint << " ";
            newFileBIN << airport.flightNumber << " ";
            newFileBIN << airport.arrivalDate << " ";
            newFileBIN << airport.arrivalTime << " ";
            newFileBIN << airport.flightDelay << endl;
            cout << airport.arrivalPoint << " "
                << airport.flightNumber << " "
                << airport.arrivalDate << " "
                << airport.arrivalTime << " "
                << airport.flightDelay << endl;
        }
    }
    cout << "Новый бинарный файл с задержками по дате создан" << endl;

    fileBIN.close();
    newFileBIN.close();
}

```

Код функции, реализующей изменение времени прилета для

определенного рейса в бинарном файле, представлен в листинге 2.7.

### Листинг 2.7 – Код реализации функции изменения времени прилета для определенного рейса в бинарном файле

```
void changeArrivalTimeInBin() { //Изменение времени прилета для
определенного рейса в бинарном файле
    fstream fileBIN("file.bin");
    if (!fileBIN.is_open()) {
        cout << "Ошибка при открытии бинарного файла" << endl;
        return;
    }
    Airport airport, airportNew;
    string race;
    char time[30];
    int i = 0;
    bool flag = false;
    cout << "Введите номер рейса, у которого нужно заменить время: "; cin
>> race;
    while (!fileBIN.eof()) {
        fileBIN.read((char*)&airport, sizeof(Airport));
        if (airport.flightNumber == race) {
            flag = true;
            cout << "Введите другое время прилета в формате ЧЧ:ММ - ";
cin >> time;

            fileBIN.seekg(sizeof(Airport) * i);
            fileBIN.read((char*)&airportNew, sizeof(Airport));
            for (int j = 0; j < 30; j++) {
                airportNew.arrivalTime[j] = time[j];
            }

            fileBIN.seekp(sizeof(Airport) * i, ios::beg);
            fileBIN.write((char*)&airportNew, sizeof(Airport));
            cout << "Информация о рейсе изменена" << endl;
            break;
        }
        i++;
    }
    if (!flag) {
        cout << "Такой рейс не найден" << endl;
    }
}
```

## 3.2 Код основной программы

Код основной программы представлен в листинге 2.8.

### Листинг 2.8 – Код основной программы

```
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
#include <sstream>
using namespace std;

struct Airport{ //Структура аэропорта
    char arrivalPoint[30]; //Пункт прилета
    char flightNumber[30]; //Номер рейса, формат RM-915
    char arrivalDate[30]; //Дата прилета 23.09.2003
    char arrivalTime[30]; //Время прилета 15.39
    char flightDelay[30]; //Задержка в часах 1.40
```

```

};

void createBinFile() { //Создание бинарного файла
    ifstream fileTXT("file.txt");
    ofstream fileBIN("file.bin");
    if (!fileTXT.is_open()) {
        cout << "Ошибка при открытии текстового файла" << endl;
        return;
    }
    if (!fileBIN.is_open()) {
        cout << "Ошибка при открытии бинарного файла" << endl;
        return;
    }
    Airport airport;
    while (!fileTXT.eof()) {
        fileTXT >> airport.arrivalPoint;
        fileTXT >> airport.flightNumber;
        fileTXT >> airport.arrivalDate;
        fileTXT >> airport.arrivalTime;
        fileTXT >> airport.flightDelay;
        fileBIN.write((char*)& airport, sizeof(Airport));
    }
    fileBIN.close();
    fileTXT.close();
    cout << "Данные из файла txt успешно добавлена в файл bin" << endl;
}

void readBinFile() { //Вывод бинарного файла
    Airport airport;
    ifstream fileBIN("file.bin");
    if (!fileBIN.is_open()) {
        cout << "Ошибка при открытии бинарного файла" << endl;
        return;
    }
    int i = 1;
    fileBIN.read((char*)&airport, sizeof(Airport));
    cout << "Вывод бинарного файла" << endl;
    cout << "№" << "\t";
    cout << left << setw(20) << "Пункт прилета" << "\t";
    cout << left << setw(20) << "Номер рейса" << "\t";
    cout << left << setw(20) << "Дата прилета" << "\t";
    cout << left << setw(20) << "Время прилета" << "\t";
    cout << left << setw(20) << "Задержка в часах" << endl;
    while (!fileBIN.eof()) {
        cout << i << "\t";
        cout << left << setw(20) << airport.arrivalPoint << "\t";
        cout << left << setw(20) << airport.flightNumber << "\t";
        cout << left << setw(20) << airport.arrivalDate << "\t";
        cout << left << setw(20) << airport.arrivalTime << "\t";
        cout << left << setw(20) << airport.flightDelay << endl;
        fileBIN.read((char*)&airport, sizeof(Airport));
        i++;
    }
    fileBIN.close();
}

void saveBinInNewTxt() { //Извлечь данные из бинарного файла в новый
текстовый файл
    ifstream fileBIN("file.bin");
    ofstream newFileTXT("newFile.txt");
    if (!fileBIN.is_open()) {
        cout << "Ошибка при открытии бинарного файла" << endl;
        return;
    }
}

```



```

    }
    if (!newFileTXT.is_open()) {
        cout << "Ошибка при открытии текстового файла" << endl;
        return;
    }
    Airport airport;
    fileBIN.read((char*)&airport, sizeof(Airport));
    while (!fileBIN.eof()) {
        newFileTXT << airport.arrivalPoint << " ";
        newFileTXT << airport.flightNumber << " ";
        newFileTXT << airport.arrivalDate << " ";
        newFileTXT << airport.arrivalTime << " ";
        newFileTXT << airport.flightDelay << "\n";
        fileBIN.read((char*)&airport, sizeof(Airport));
    }
    cout << "Данные из бинарного файла успешно добавлены в новый текстовый
файл" << endl;
    fileBIN.close();
    newFileTXT.close();
}

void getByNumber() { //Получение данных о рейсе по её номеру
    ifstream fileBIN("file.bin");
    if (!fileBIN.is_open()) {
        cout << "Ошибка при открытии бинарного файла" << endl;
        return;
    }
    int num;
    cout << "Введите номер: "; cin >> num;
    Airport airport;
    fileBIN.seekg(sizeof(airport) * (num - 1), ios::cur);
    fileBIN.read((char*)&airport, sizeof(airport));
    cout << "Запись с номером " << num << ": " << endl
        << airport.arrivalPoint << " "
        << airport.flightNumber << " "
        << airport.arrivalDate << " "
        << airport.arrivalTime << " "
        << airport.flightDelay;
    fileBIN.close();
    cout << endl;
}

void deleteByKey() { //Удаление рейса по ключу, его замена на Tula TC-111
11.11.2011 11:11 1
    fstream fileBIN("file.bin");
    if (!fileBIN.is_open()) {
        cout << "Ошибка при открытии бинарного файла" << endl;
        return;
    }
    Airport airport, airportNew;
    string race;
    char arrivalPointNew[30], flightNumberNew[30], arrivalDateNew[30],
arrivalTimeNew[30], flightDelayNew[30];
    int i = 0;
    bool flag = false;
    cout << "Введите номер рейса, который необходимо заменить: "; cin >>
race;
    while (!fileBIN.eof()) {
        fileBIN.read((char*)&airport, sizeof(Airport));
        if (airport.flightNumber == race) {
            flag = true;
            cout << "Введите новый пункт прилета: "; cin >>
arrivalPointNew;

```

```

    flightNumberNew;    cout << "Введите новый номер рейса:"; cin >>
    arrivalDateNew;     cout << "Введите новую дату прилета:"; cin >>
    arrivalTimeNew;     cout << "Введите новое время прилета:"; cin >>
    flightDelayNew;     cout << "Введите новую задержку прилета в часах:"; cin >>

    fileBIN.seekg(sizeof(Airport) * i);
    fileBIN.read((char*)&airportNew, sizeof(Airport));
    for (int j = 0; j < 30; j++) {
        airportNew.arrivalPoint[j] = arrivalPointNew[j];
        airportNew.flightNumber[j] = flightNumberNew[j];
        airportNew.arrivalDate[j] = arrivalDateNew[j];
        airportNew.arrivalTime[j] = arrivalTimeNew[j];
        airportNew.flightDelay[j] = flightDelayNew[j];
    }

    fileBIN.seekp(sizeof(Airport) * i, ios::beg);
    fileBIN.write((char*)&airportNew, sizeof(Airport));
    cout << "Информация о рейсе изменена" << endl;
    break;
}
i++;
}
if (!flag) {
    cout << "Такой рейс не найден" << endl;
}
}

void createNewBinWitchDateDelays() { // Создание нового бинарного файла с
задержками по дате
    ifstream fileBIN("file.bin");
    ofstream newFileBIN("newFile.bin");
    if (!fileBIN.is_open()) {
        cout << "Ошибка при открытии бинарного файла" << endl;
        return;
    }
    if (!newFileBIN.is_open()) {
        cout << "Ошибка при открытии нового бинарного файла" << endl;
        return;
    }
    Airport airport;
    string date;
    cout << "Введите дату, по которой надо создать новый бинарный файл:"
";cin >> date;
    while (!fileBIN.eof()) {
        fileBIN.read((char*)&airport, sizeof(Airport));
        if (airport.arrivalDate == date) {
            newFileBIN << airport.arrivalPoint << " ";
            newFileBIN << airport.flightNumber << " ";
            newFileBIN << airport.arrivalDate << " ";
            newFileBIN << airport.arrivalTime << " ";
            newFileBIN << airport.flightDelay << endl;
            cout << airport.arrivalPoint << " "
                << airport.flightNumber << " "
                << airport.arrivalDate << " "
                << airport.arrivalTime << " "
                << airport.flightDelay << endl;
        }
    }
    cout << "Новый бинарный файл с задержками по дате создан" << endl;
}

```

```

        fileBIN.close();
        newFileBIN.close();
    }

void changeArrivalTimeInBin() { //Изменение времени прилета для
определенного рейса в бинарном файле
    fstream fileBIN("file.bin");
    if (!fileBIN.is_open()) {
        cout << "Ошибка при открытии бинарного файла" << endl;
        return;
    }
    Airport airport, airportNew;
    string race;
    char time[30];
    int i = 0;
    bool flag = false;
    cout << "Введите номер рейса, у которого нужно заменить время: "; cin
>> race;
    while (!fileBIN.eof()) {
        fileBIN.read((char*)&airport, sizeof(Airport));
        if (airport.flightNumber == race) {
            flag = true;
            cout << "Введите другое время прилета в формате ЧЧ:ММ - ";
cin >> time;

            fileBIN.seekg(sizeof(Airport) * i);
            fileBIN.read((char*)&airportNew, sizeof(Airport));
            for (int j = 0; j < 30; j++) {
                airportNew.arrivalTime[j] = time[j];
            }

            fileBIN.seekp(sizeof(Airport) * i, ios::beg);
            fileBIN.write((char*)&airportNew, sizeof(Airport));
            cout << "Информация о рейсе изменена" << endl;
            break;
        }
        i++;
    }
    if (!flag) {
        cout << "Такой рейс не найден" << endl;
    }
}

int main() {
    setlocale(LC_ALL, "Rus");
    cout << "1 - Создание бинарного файла с данными их файла txt" << endl
        << "2 - Вывод бинарного файла" << endl
        << "3 - Сохранение данных из бинарного файла в текстовый" <<
endl
        << "4 - Вывод данных о рейсе по её номеру в файле" << endl
        << "5 - Замена рейса по номеру рейса" << endl
        << "6 - Создать новый бинарный файл с задержками по дате" <<
endl
        << "7 - Изменить время прилета для определенного рейса в
бинарном файле" << endl
        << "0 - Завершить работу программы" << endl << endl;
    int choose;
    while (true) {
        string race, time;
        cout << "Выберите 1, 2, 3, 4, 5, 6, 7, 0: ";
        cin >> choose;
        switch (choose)
        {

```

```

        case 1:
            createBinFile();
            break;
        case 2:
            readBinFile();
            break;
        case 3:
            saveBinInNewTxt();
            break;
        case 4:
            getByNumber();
            break;
        case 5:
            deleteByKey();
            break;
        case 6:
            createNewBinWitchDateDelays();
            break;
        case 7:
            changeArrivalTimeInBin();
            break;
        case 0:
            return 0;
        default:
            cout << "Неверный ввод" << endl;
            break;
    }
}

```

#### 4. Результаты тестирования

Результат тестирования программы представлен на рисунке 2.3.

```

1 - Создание бинарного файла с данными их файла txt
2 - Вывод бинарного файла
3 - Сохранение данных из бинарного файла в текстовый
4 - Вывод данных о рейсе по её номеру в файле
5 - Замена рейса по номеру рейса
6 - Создать новый бинарный файл с задержками по дате
7 - Изменить время прилета для определенного рейса в бинарном файле
0 - Завершить работу программы

Выберите 1, 2, 3, 4, 5, 6, 7, 0: 1
Данные из файла txt успешно добавлена в файл bin
Выберите 1, 2, 3, 4, 5, 6, 7, 0: 2
Вывод бинарного файла
№      Пункт прилета      Номер рейса      Дата прилета      Время прилета      Задержка в часах
1      Moscow      RM-915      23.09.2022      15:40      0
2      Vladimir      VP-180      27.10.2022      17:00      1
3      Meleyz      MZ-220      23.09.2022      18.40      2
4      Yerevan      YE-209      13.11.2021      18:40      0
5      Rome      RM-703      18.06.2022      13.30      0
6      Paris      PS-109      31.04.2022      11.20      0
7      Venice      VC-111      12.06.2022      19.00      0.5
Выберите 1, 2, 3, 4, 5, 6, 7, 0: 3
Данные из бинарного файла успешно добавлены в новый текстовый файл
Выберите 1, 2, 3, 4, 5, 6, 7, 0: 4
Введите номер: 4
Запись с номером 4:
Yerevan YE-209 13.11.2021 18:40 0
Выберите 1, 2, 3, 4, 5, 6, 7, 0: 5
Введите номер рейса, который необходимо заменить: PS-109
Введите новый пункт прилета: Qqqqqqq
Введите новый номер рейса: QQ-000
Введите новую дату прилета: 11.11.2011
Введите новое время прилета: 11.11
Введите новую задержку прилета в часах: 11
Информация о рейсе изменена
Выберите 1, 2, 3, 4, 5, 6, 7, 0: 6
Введите дату, по которой надо создать новый бинарный файл: 23.09.2022
Moscow RM-915 23.09.2022 15:40 0
Meleyz MZ-220 23.09.2022 18.40 2
Новый бинарный файл с задержками по дате создан
Выберите 1, 2, 3, 4, 5, 6, 7, 0: 7
Введите номер рейса, у которого нужно заменить время: MZ-220
Введите другое время прилета в формате ЧЧ:ММ - 13.33
Информация о рейсе изменена
Выберите 1, 2, 3, 4, 5, 6, 7, 0: 2
Вывод бинарного файла
№      Пункт прилета      Номер рейса      Дата прилета      Время прилета      Задержка в часах
1      Moscow      RM-915      23.09.2022      15:40      0
2      Vladimir      VP-180      27.10.2022      17:00      1
3      Meleyz      MZ-220      23.09.2022      13.33      2
4      Yerevan      YE-209      13.11.2021      18:40      0
5      Rome      RM-703      18.06.2022      13.30      0
6      Qqqqqqq      QQ-000      11.11.2011      11.11      11
7      Venice      VC-111      12.06.2022      19.00      0.5
Выберите 1, 2, 3, 4, 5, 6, 7, 0: 0

```

Рисунок 2.3 – Результат тестирования программы

**Вывод:** в ходе выполнения практической работы были получены навыки применения файловых потоков языка C++ (или файлов языка Си) по управлению текстовым и двоичным файлами.