

Министерство цифрового развития, связи и массовых  
коммуникаций  
Российской Федерации Сибирский государственный  
университет телекоммуникаций и информатики  
кафедра ТС и ВС

***Лабораторная работа 3***  
ПО ДИСЦИПЛИНЕ  
«Объектно-ориентированное  
программирование»

Выполнил: Демин С. А.  
Группа: ИКС-433  
Вариант: 5

Проверил: Нейдорф П. Я.

Новосибирск, 2025

## Текстовые и двоичные (бинарные) файлы

**Цель работы:** Работа с текстовыми и двоичными (бинарными) файлами в языке C++)

### Задание:

Согласно своему варианту выполнить следующие действия:

1. С помощью текстового редактора создать текстовый файл, в который занести исходную информацию. Рекомендуется вначале занести количество записей, а затем последовательно все записи, причем тестовые поля вводить в отдельной строке, а числовые – можно в отдельной строке или в одной через пробел или разделитель табуляцию.

Например, содержимое файла, который будет включать 7 записей вида – фамилия, группа, 5 оценок :

7

Иванов

Ф01-а

5 4 5 4 3

Петров

А01-б

5 4 5 3 5

...

2. Разработать КЛАСС для создания объектов с полями из текстового файла согласно вашему варианту.

3. Написать первую программу, которая считывает информацию из созданного текстового файла в поля объекта и записывает ее в двоичный (бинарный ) файл

4. Написать вторую программу, которая считывает информацию из двоичного (бинарного) файла и выполняет над данными поставленную в варианте задания задачу. После повторного чтения двоичного файла результат работы выводится на экран.

Все текстовые и двоичный файлы должны располагаться или в личной папке или в папке проекта.

### Вариант:

Создать двоичный файл с информацией об успеваемости студентов некоторого факультета за все время обучения.

ФИО студента	Успеваемость (средний балл)					
	1 семестр	2 семестр	3 семестр	4 семестр	...	10 семестр
Иванов	5	4,8	3,7	4,5	...	4
...	...	...	...	...	...	...

Добавить в файл поле «Средняя успеваемость». Удалить из файла информацию о студентах со средним баллом менее 3.5.

### **Текст программы создания бинарного файла**

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

using namespace std;

class Student {
public:
    char name[50];
    double grades[10];};

int main() {
    ifstream in("students.txt",ios::in);
    ofstream out("students.bin",ios::binary);
    if (!in) {
        cout << "Ошибка открытия файла students.txt!" << endl;}

    int n;
    in>>n;
    in.ignore();

    cout<<"Вывод информации о студентах:\n\n"<<endl;
    for (int i=0;i<n;i++){
        Student s;
        in.getline(s.name,50);
        for (int j=0;j<10;j++) {
            in>>s.grades[j];}

        in.ignore();
        double sum=0;
        for (int j=0;j<10;j++) {
            sum+=s.grades[j];}

        double average=sum/10;

        cout<<"Студент: "<<s.name<<"\n";
        cout<<"Оценки: ";
        for (int j=0;j<10;j++){
            cout<<s.grades[j]<<" ";}

        cout<<"\nСредний балл: "<<average<<"\n\n";

        out.write((char*)&s,sizeof(Student));}

    in.close();
    out.close();
    cout<<"Бинарный файл создан..."<<endl;}
```

## Текст программы обработки бинарного файла

```
#include <iostream>
#include <fstream>

using namespace std;

class StudentOld {
public:
    char name[50];
    double grades[10];};

class StudentNew {
public:
    char name[50];
    double grades[10];
    double average;

    void calcaverage(){
        double sum=0;
        for(int i=0;i<10;i++) {
            sum+=grades[i];}
        average=sum/10;}

    void writefile(ofstream& out){
        out.write((char*)this, sizeof(StudentNew));}

    void vivod(){
        cout<<name<<" - "<<average<<endl;}
};

int main() {
    ifstream in("students.bin", ios::binary);
    if (!in){
        cout << "Файл не найден!" << endl;
        return 1;}

    StudentOld sOld;
    int count = 0;

    while (in.read((char*)&sOld,sizeof(StudentOld))){
        count++;}
    in.close();

    in.open("students.bin", ios::binary);
    StudentNew* students=new StudentNew[count];

    for(int i = 0; i < count; i++){
        in.read((char*)&sOld, sizeof(StudentOld));
        for(int j=0;j<50;j++){
            students[i].name[j]=sOld.name[j];}
        for(int j=0;j<10;j++){
            students[i].grades[j]=sOld.grades[j];}
        students[i].calcaverage();}
```

```

}
in.close();

ofstream out("students.bin", ios::binary);

cout << "Остались студенты со средним баллом >= 3.5:" << endl;
for (int i = 0; i < count; i++) {
    if (students[i].average >= 3.5) {
        students[i].writefile(out);
        students[i].vivod();}}
out.close();}

```

## Текстовый файл students.txt

3

Демин Сергей

5 4.3 5 3.2 4.1 5 4.8 3.1 5 4.8

Иванов Иван

3 3 3 3 3 3 3 3 5 5

Синица Михаил 5 5 5 5 5 5 5 5 5 5

## Результаты:

### 1. Вывод информации о студентах:

Студент: Демин Сергей

Оценки: 5 4.3 5 3.2 4.1 5 4.8 3.1 5 4.8

Средний балл: 4.43

Студент: Иванов Иван

Оценки: 3 3 3 3 3 3 3 3 5 5

Средний балл: 3.4

Студент: Синица Михаил

Оценки: 5 5 5 5 5 5 5 5 5 5

Средний балл: 5

Бинарный файл создан...

### 2. Остались студенты со средним баллом >= 3.5:

Демин Сергей - 4.43

Синица Михаил - 5

## Описание результатов:

**Программа создания бинарного файла** успешно прочитала текстовый файл students.txt, содержащий информацию о 3 студентах с их оценками за 10 семестров. Для каждого студента программа вычислила средний балл и отобразила всю информацию на экране. Затем данные были записаны в бинарный файл students.bin в формате структуры Student (без поля среднего балла).

**Программа обработки бинарного файла** корректно выполнила поставленную задачу:

- ⊗ Прочитала все записи из бинарного файла
- ⊗ Вычислила средний балл для каждого студента
- ⊗ Удалила из файла студента Иванова Ивана (средний балл 3.4 < 3.5)
- ⊗ Сохранила в файле только студентов Демина Сергея и Синицу Михаила
- ⊗ Вывела на экран список оставшихся студентов с их средними баллами

### **Выводы:**

В ходе выполнения лабораторной работы успешно освоены принципы работы с текстовыми и бинарными файлами в C++, реализовано преобразование текстовых данных в бинарный формат с последующей фильтрацией студентов по среднему баллу, что подтвердило эффективность бинарных файлов для хранения структурированных данных и практическую ценность объектно ориентированного подхода при обработке информации.

### **Контрольные вопросы:**

#### **1. Понятие файла**

Файл — это именованная область данных на носителе информации, доступная для чтения и записи. В программировании файл рассматривается как последовательность байтов, которая может быть обработана программой.

#### **2. Способ описания файла**

В языке C++ файл описывается с помощью объектов классов `ifstream` (для чтения), `ofstream` (для записи) и `fstream` (для чтения и записи). Эти классы определены в заголовочном файле `<fstream>`.

#### **3. Описание структурного типа**

Структурный тип (структура) в C++ определяется с помощью ключевого слова `struct`. Пример:

#### **4. Стандартные функции для работы с файлами**

Основные методы: `open()`, `close()`, `read()`, `write()`, `getline()`, `<<`, `>>`. Для бинарных файлов используются методы `read()` и `write()`.

#### **5. С помощью какой функции осуществляется подготовка файла для чтения?**

Для подготовки файла к чтению используется метод `open()` с указанием имени файла и режима (например, `ios::in` для текстового режима, `ios::binary` для бинарного).

**6. С помощью каких функций выполняется чтение данных из файла?**

Для текстовых файлов: `getline()`, `>>`. Для бинарных файлов: `read()`.

**7. Как распознать конец файла данных?**

С помощью метода `eof()`, который возвращает `true`, если достигнут конец файла. Также можно проверять успешность операции чтения.