



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н. Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)  
КАФЕДРА «Информационная безопасность» (ИУ8)

Лабораторная работа № 6  
ПО КУРСУ  
«Алгоритмические языки»  
на тему «Работа с потоками ввода и вывода в языке Си++»

Студент

ИУ8-13  
(Группа)

В. С. Ажгирей  
(И. О. Фамилия)

Преподаватель:

М. В. Малахов  
(И.О. Фамилия)

## Введение

### Цели и задачи работы

Цель работы состоит в овладении навыками разработки программ на языке Си++, использующих ввод/вывод в файлы. Для достижения цели необходимо выполнить следующие задачи:

- изучить необходимые учебные материалы, посвященные потокам ввода и вывода языка Си++;
- разработать программы на языке Си++ для решения заданных вариантов заданий;
- отладить программы;
- выполнить решение контрольного примера небольшой размерности с помощью программы и ручной расчет контрольного примера;
- подготовить отчет по лабораторной работе.

### Условия для 1 варианта

В лабораторной работе необходимо разработать приложение, работающее в различных режимах: чтения, записи, очистки файла. Также задается текстовый или двоичный режим открытия файла. В режиме записи исходные данные вводятся с клавиатуры (количество объектов, заданных структурным типом, данные этих объектов), введенные данные *дозаписываются* в файл, не затирая предыдущую информацию. В режиме чтения данные, сохраненные в режиме записи в файле, читаются из файла и выводятся на консоль. В режиме очистки файла производится удаление всей информации из файла (по окончании работы данного режима файл на ПК должен существовать и не иметь содержимого).

Режимы приложения задаются как параметры командной строки и сохраняются в глобальные переменные: вначале чтение или запись, затем текстовый или двоичный режим, далее имя файла, например:

write binary MyFile.dat режим записи, файл в двоичном режиме, имя файла MyFile.dat;

read text MyFile.txt режим чтения, файл в текстовом режиме, имя файла MyFile.txt.

Также в ПО необходимо обрабатывать ситуацию, при которой читаемый файл оказался битым, т.е. в нем хранятся некорректные данные, например, поле типа `int` хранит строковое значение.

ПО должно содержать несколько осмысленных функций, разработанных студентом.

***Необходимо использовать библиотеку `fstream`.***

- 1) Структура «студент» (поля: ФИО, группа, номер зачетной книжки, массив 4-х оценок за сессию)

## Основная часть

Исходный текст файла main.cpp:

```
#include "sources.h"

std::string mode, representation, fileName;

int main(int argc, char *argv[])
{
    if (argc < 4)
    {
        std::cout << "Error: Insufficient command line arguments" << std::endl;
        return 1;
    }

    mode = argv[1];
    representation = argv[2];
    fileName = argv[3];

    fileManager(mode, representation, fileName);

    return 0;
}
```

Исходный текст файла sources.h:

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

struct Student
{
    std::string full_name;
    unsigned short int study_group;
    unsigned int credit_card_number;
    unsigned short int grades[4];
};

void fileManager(const std::string mode, const std::string representation, const
std::string fileName);
```

## Исходный текст файла sources.cpp:

```
#include "sources.h"

void printStudent(const Student &student)
{
    std::cout << "Full name: " << student.full_name << std::endl;
    std::cout << "Study group: " << student.study_group << std::endl;
    std::cout << "Credit card number: " << student.credit_card_number <<
std::endl;
    std::cout << "Assessment for exams: ";
    for (size_t i = 0; i < 4; ++i)
    {
        std::cout << student.grades[i] << ' ';
    }
    std::cout << '\n'
        << "#####" << std::endl;
}

void fileManager(const std::string mode, const std::string representation, const
std::string fileName)
{
    if (mode == "-r")
    {
        std::ifstream readFile;
        if (representation == "bin")
        {
            readFile.open(fileName, std::ios::binary);
        }
        else
        {
            readFile.open(fileName);
        }

        if (!readFile)
        {
            std::cerr << "Failed to open file" << std::endl;
            return;
        }

        Student student;
        if (representation == "bin")
        {
            char currentByte;
            while (readFile.get(currentByte))
            {
                student.full_name = "";
                char endString = '\0';
                while (currentByte != endString)
                {

```

```

        student.full_name += currentByte;
        readFile.get(currentByte);
    }

    readFile.read((char *)&student.study_group, sizeof(unsigned short
int));
    readFile.read((char *)&student.credit_card_number,
sizeof(unsigned int));
    for (size_t i = 0; i < 4; ++i)
    {
        readFile.read((char *)&student.grades[i], sizeof(unsigned
short int));
    }

    printStudent(student);
}
else
{
    std::string study_group, credit_card_number, mark1, mark2, mark3,
mark4;
    while (readFile >> student.full_name >> study_group >>
credit_card_number >> mark1 >> mark2 >> mark3 >> mark4)
    {
        student.study_group = (unsigned short int)std::stoi(study_group);
        student.credit_card_number = (unsigned
int)std::stoi(credit_card_number);
        student.grades[0] = (unsigned short int)std::stoi(mark1);
        student.grades[1] = (unsigned short int)std::stoi(mark2);
        student.grades[2] = (unsigned short int)std::stoi(mark3);
        student.grades[3] = (unsigned short int)std::stoi(mark4);

        printStudent(student);
    }
}

readFile.close();
}
else if (mode == "-w")
{
    std::ofstream writeFile;
    if (representation == "bin")
    {
        writeFile.open(fileName, std::ios::binary | std::ios::app);
    }
    else
    {
        writeFile.open(fileName, std::ios::app);
    }
}

```

```

    if (!writeFile)
    {
        std::cerr << "Failed to open file" << std::endl;
        return;
    }

    unsigned int n;
    std::cout << "Enter count of students: ";
    std::cin >> n;
    for (size_t i = 0; i < n; ++i)
    {
        std::string full_name;
        std::cout << "Enter the student's name: ";
        std::cin >> full_name;

        unsigned short study_group;
        std::cout << "Enter the study group: ";
        std::cin >> study_group;

        unsigned int credit_card_number;
        std::cout << "Enter credit card number: ";
        std::cin >> credit_card_number;

        unsigned short int mark1, mark2, mark3, mark4;
        std::cout << "Enter 4 assessments for exams: ";
        std::cin >> mark1 >> mark2 >> mark3 >> mark4;
        unsigned short int grades[] = {mark1, mark2, mark3, mark4};

        if (representation == "bin")
        {
            writeFile.write(full_name.c_str(), full_name.size() + 1);
            writeFile.write((char *)&study_group, sizeof(unsigned short
int));
            writeFile.write((char *)&credit_card_number, sizeof(unsigned
int));
            for (size_t i = 0; i < 4; ++i)
            {
                writeFile.write((char *)&grades[i], sizeof(unsigned short
int));
            }
        }
        else
        {
            writeFile << full_name << " " << study_group << " " <<
credit_card_number << " " << mark1 << " " << mark2 << " " << mark3 << " " <<
mark4 << std::endl;
        }

        std::cout << "#####" <<
std::endl;

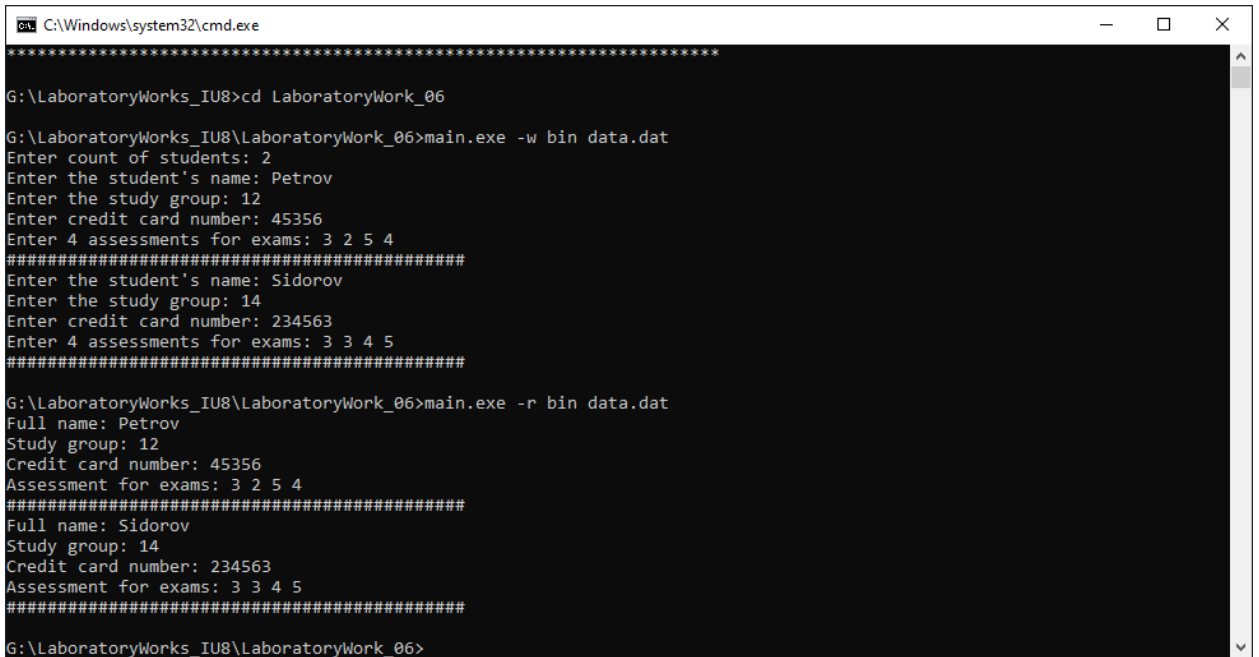
```

```
    }

    writeFile.close();
}
else
{
    std::ofstream clearFile;
    if (representation == "bin")
    {
        clearFile.open(fileName, std::ios::binary | std::ios::trunc);
    }
    else
    {
        clearFile.open(fileName, std::ios::trunc);
    }
    clearFile.close();
}
}
```



## Снимки выполнения работы программы



```
C:\Windows\system32\cmd.exe

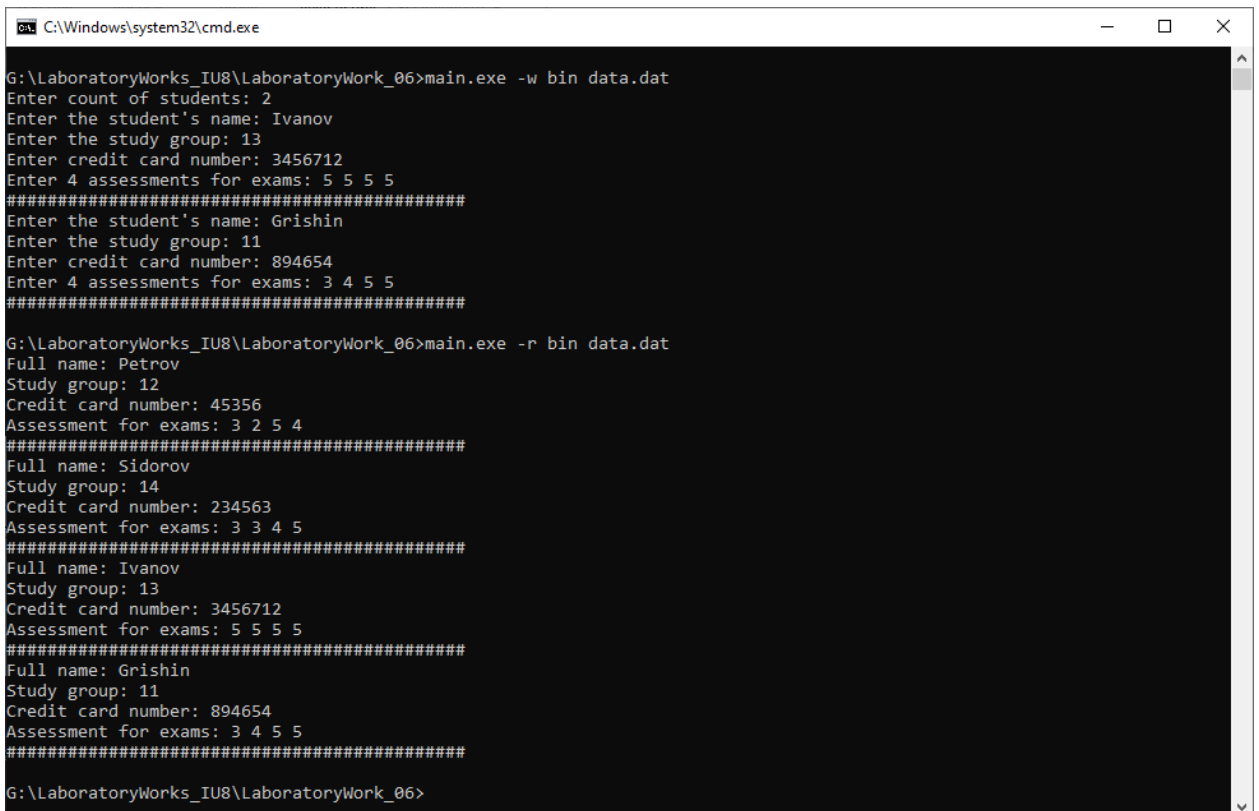
G:\LaboratoryWorks_IU8>cd LaboratoryWork_06

G:\LaboratoryWorks_IU8\LaboratoryWork_06>main.exe -w bin data.dat
Enter count of students: 2
Enter the student's name: Petrov
Enter the study group: 12
Enter credit card number: 45356
Enter 4 assessments for exams: 3 2 5 4
#####
Enter the student's name: Sidorov
Enter the study group: 14
Enter credit card number: 234563
Enter 4 assessments for exams: 3 3 4 5
#####

G:\LaboratoryWorks_IU8\LaboratoryWork_06>main.exe -r bin data.dat
Full name: Petrov
Study group: 12
Credit card number: 45356
Assessment for exams: 3 2 5 4
#####
Full name: Sidorov
Study group: 14
Credit card number: 234563
Assessment for exams: 3 3 4 5
#####

G:\LaboratoryWorks_IU8\LaboratoryWork_06>
```

Рисунок 1 – Запуск программы в режиме бинарной записи в файл data.dat .  
Запуск программы в режиме чтения бинарных данных из файла data.dat .



```
C:\Windows\system32\cmd.exe

G:\LaboratoryWorks_IU8\LaboratoryWork_06>main.exe -w bin data.dat
Enter count of students: 2
Enter the student's name: Ivanov
Enter the study group: 13
Enter credit card number: 3456712
Enter 4 assessments for exams: 5 5 5 5
#####
Enter the student's name: Grishin
Enter the study group: 11
Enter credit card number: 894654
Enter 4 assessments for exams: 3 4 5 5
#####

G:\LaboratoryWorks_IU8\LaboratoryWork_06>main.exe -r bin data.dat
Full name: Petrov
Study group: 12
Credit card number: 45356
Assessment for exams: 3 2 5 4
#####
Full name: Sidorov
Study group: 14
Credit card number: 234563
Assessment for exams: 3 3 4 5
#####
Full name: Ivanov
Study group: 13
Credit card number: 3456712
Assessment for exams: 5 5 5 5
#####
Full name: Grishin
Study group: 11
Credit card number: 894654
Assessment for exams: 3 4 5 5
#####

G:\LaboratoryWorks_IU8\LaboratoryWork_06>
```

Рисунок 2 – Запуск программы в режиме бинарной дозаписи в файл data.dat .  
Запуск программы в режиме чтения бинарных данных из файла data.dat .

```
Выбрать C:\Windows\system32\cmd.exe
G:\LaboratoryWorks_IU8\LaboratoryWork_06>main.exe -w text data.txt
Enter count of students: 2
Enter the student's name: Ivanov
Enter the study group: 12
Enter credit card number: 89351
Enter 4 assessments for exams: 2 4 4 3
#####
Enter the student's name: Sidorov
Enter the study group: 33
Enter credit card number: 98723
Enter 4 assessments for exams: 4 5 5 5
#####

G:\LaboratoryWorks_IU8\LaboratoryWork_06>main.exe -r text data.txt
Full name: Ivanov
Study group: 12
Credit card number: 89351
Assessment for exams: 2 4 4 3
#####
Full name: Sidorov
Study group: 33
Credit card number: 98723
Assessment for exams: 4 5 5 5
#####

G:\LaboratoryWorks_IU8\LaboratoryWork_06>_
```

Рисунок 3 – Запуск программы в режиме текстовой записи в файл data.txt .  
Запуск программы в режиме чтения текстовых данных из файла data.txt .

```
C:\Windows\system32\cmd.exe
G:\LaboratoryWorks_IU8\LaboratoryWork_06>main.exe -w text data.txt
Enter count of students: 1
Enter the student's name: Petrov
Enter the study group: 13
Enter credit card number: 84534
Enter 4 assessments for exams: 5 5 5 5
#####

G:\LaboratoryWorks_IU8\LaboratoryWork_06>main.exe -r text data.txt
Full name: Ivanov
Study group: 12
Credit card number: 89351
Assessment for exams: 2 4 4 3
#####
Full name: Sidorov
Study group: 33
Credit card number: 98723
Assessment for exams: 4 5 5 5
#####
Full name: Petrov
Study group: 13
Credit card number: 84534
Assessment for exams: 5 5 5 5
#####

G:\LaboratoryWorks_IU8\LaboratoryWork_06>
```

Рисунок 4 – Запуск программы в режиме текстовой дозаписи в файл data.txt .  
Запуск программы в режиме чтения текстовых данных из файла data.txt .

## Заключение

Задачи лабораторной работы были решены, результаты проверены. Изучена на практике работа с файлами и библиотека fstream в языке C/C++, а также глобальные переменные.