МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №8**

**з дисципліни «Основи програмування»**

**на тему: *«*Бінарні файли. Робота с файлами засобами мови С++.»**

Виконав: студент гр. ПЗ2011

Кулик Сергій Вадимович

Прийняла: ас. Нежуміра О.І.

Дніпро, 2020

Тема. Бінарні файли. Робота с файлами засобами мови С++  
Мета:   
1) сформувати навики роботи з бінарними файлами;   
2) засвоїти реалізацію файлового введення/виведення у стилі мови С++.

**Завдання**

Розробити програму роботи з бінарними файлами:

1) ініціалізувати в програмі масив записів для предметної області з лабораторної роботи №5 «Структурний тип даних в мові С++»;   
2) записати дані з масиву в бінарний файл;   
3) обробити отриманий бінарній файл за індивідуальним завданням;   
4) вивести оброблений файл на консоль.

Вимоги до програми:   
− імена файлів задавати в командному рядку;  
− дані для запису у файл описати в масиві структур з 8 – 10 елементів;  
− зчитані з файлу дані забороняється зберігати в масиві;  
− забороняється зберігати в оперативної пам’яті одночасно більше двох зчитаних з файлу блоків; − перевіряти всі операції роботи з файлами (відкриття, читання, запису).

Варіант 5

У вхідному файлі циклічно зсунути в бік кінця файлу перші три блоки.

**1. Зовнішні специфікації:**  
Вхід: масив структур.  
Вихід: бінарний файл з записаним всередину масивом, оробленим за вимогою завдання.

**2.** **Алгоритм**Поміняти місцями у файлі перший та другий блок, а потім перший та третій.

**3. Текст програми**

#include <fstream>

#include <iostream>

#define lenght 128

using namespace std;

struct Progress

{

int raiting1;

int raiting2;

int raiting3;

};

struct Students

{

char surname[lenght];

int group;

Progress raiting;

};

//У вхідному файлі циклічно зсунути в бік кінця файлу перші три блоки.

int main()

{

setlocale(LC\_ALL, "ru");

Students array[] = { {"Kulik", 911, 80, 75, 80}, {"Bublik", 911, 60, 75, 80},{"Antonenko", 912, 60, 90, 82},

{"Kostiuk", 911, 80, 75, 99}, {"Belous", 911, 80, 75, 80}, {"Timchuk", 912, 58, 74, 80}, {"Pazika", 911, 80, 66, 80},

{"Maksimenko", 911, 80, 75, 80}, {"Gavrish", 911, 78, 95, 70}, {"Griva", 911, 60, 50, 50} };

char f\_name[lenght];

printf("Введите имя файла для открытия(создания):");

cin.getline(f\_name, lenght);

fstream file;

//Запись масива в файл

file.open(f\_name, ios::binary | ios::out | ios::trunc);

//Проверка на открытие

if (!file.is\_open())

cout << "Файл не может быть открыт.\n";

else

cout << "Файл успешно открыт\n";

file.write((char\*)array, sizeof array);

file.close();

int size\_of\_record = sizeof Students;

Students students\_first, students\_second;

fstream inout;

inout.open(f\_name, ios::binary | ios::in | ios::out);

if (!inout.is\_open())

cout << "Файл не может быть открыт.\n";

else

cout << "Файл успешно открыт\n";

//прочитать первый блок

inout.seekg(ios::beg);

inout.read((char\*)(&students\_first), size\_of\_record);

//прочитать второй блок

inout.seekg(size\_of\_record, ios::beg);

inout.read((char\*)(&students\_second), size\_of\_record);

//записать первый блок на позицию второго

inout.seekp(size\_of\_record, ios::beg);

inout.write((char\*)(&students\_first), size\_of\_record);

//записать второй блок на позицию первого

inout.seekp(ios::beg);

inout.write((char\*)(&students\_second), size\_of\_record);

//прочитать третий блок

inout.seekg(2\*size\_of\_record, ios::beg);

inout.read((char\*)(&students\_first), size\_of\_record);

//записать третий блок на позицию первого

inout.seekp(ios::beg);

inout.write((char\*)(&students\_first), size\_of\_record);

//записать первый блок на позицию третьего

inout.seekp(2 \* size\_of\_record, ios::beg);

inout.write((char\*)(&students\_second), size\_of\_record);

inout.close();

//печать информации про студентов

file.open(f\_name, ios::binary | ios::in);

if (!file.is\_open())

cout << "Файл не может быть открыт.\n";

else

cout << "Файл успешно открыт\n";

cout << "\nInformation about the students\n===================================================\n";

Students students;

file.read((char\*)(&students), size\_of\_record);

while (!file.eof()) {

cout << "Фамилия: " << students.surname << " \tГрупа: " << students.group << " Оценки за модуль: "

<< students.raiting.raiting1 << ' ' << students.raiting.raiting2 << ' ' << students.raiting.raiting3 << endl;

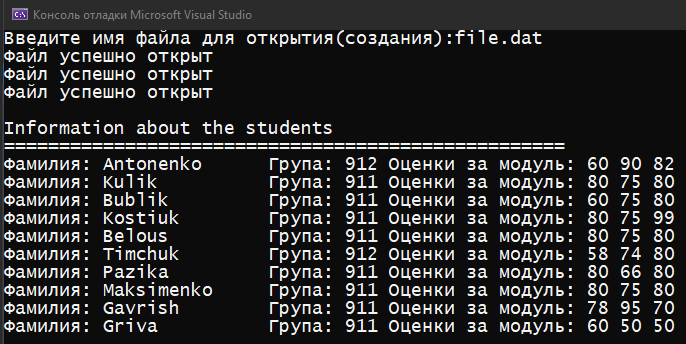
file.read((char\*)(&students), size\_of\_record);

}

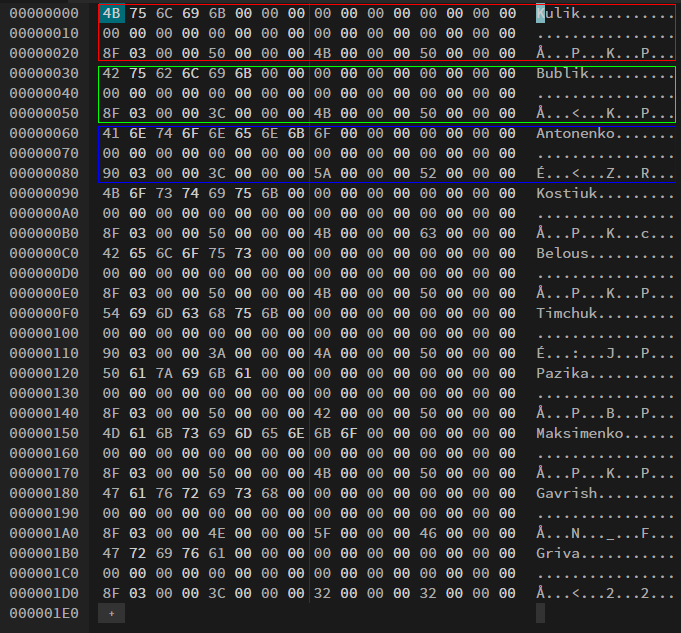
file.close();

}

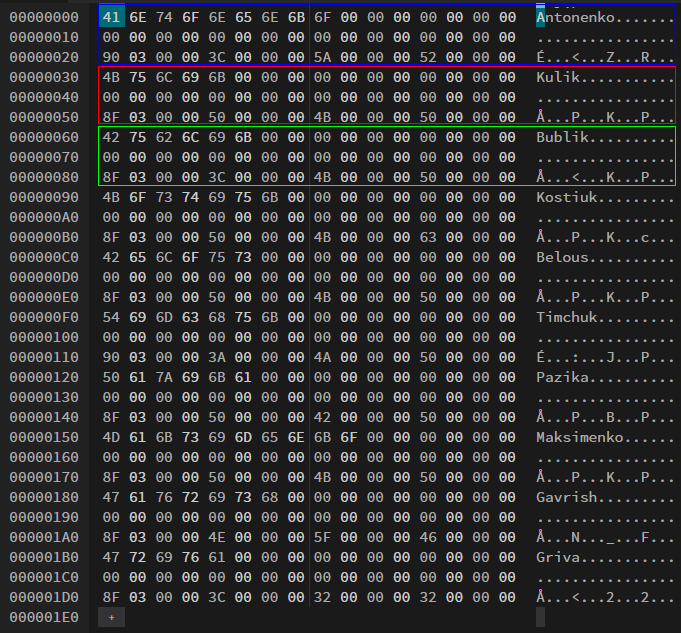
**4. Результати виконання прграми**

****

Скріншот файлу до обробки



Скріншот файлу після обробки



Зеленим, синім та червоним кольром позначені блоки над якими виконувались дії. Як можна бачити в середині файлу інформація зберігається так само як і в оперативній пам'яті. Можна бачити не використані елементи символьних рядків, які використовуються для зберігання імені студента. В консолі вони не друкуються.

**5. Висновок**

Бінарні файли зберігають інформацію у внутрішньому форматі. Вага бінарного файлу менша, ніж аналогічного йому текстового файлу. Для зчитування інформації з бінарного файлу необхідні спеціальні програми.