ЛАБОРАТОРНАЯ РАБОТА №9

“Программирование с использованием перегрузки функций и операторов”

**Цель работы:** сформировать знания и умения по работе с файлами в ЯП Си++, приобрести практические навыки создания программ на языке Си++ с использованием файлового ввода вывода и классов.

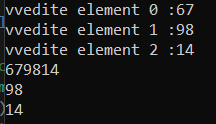


Рисунок 1 – Результат выполнения программы

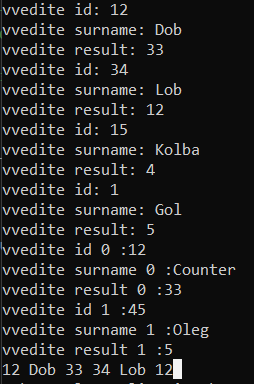
****

Рисунок 2 – Результат выполнения примера 2

**Задание 3.** Создать метод, находящий победителя соревнований.

**Решение:**

#include<iostream>

#include<fstream>

#include<vector>

#include<string>

using namespace std;

class sportsmen {

public:

char id[256];

char surname[256];

char result[256];

sportsmen()

{

cout << "Введите ID: ";

cin >> id;

cout << "Введите фамилию: ";

cin >> surname;

cout << "Введите результат: ";

cin >> result;

}

~sportsmen() {};

void store(fstream& stream);

void retrive(fstream& stream);

};

template <class typeArray>

class tArray {

public:

vector <typeArray> B;

size\_t n;

tArray(size\_t s)

{

n = s;

B.resize(s);

}

void outputTFile(const char\* filename);

void inputTFile(const char\* filename);

void input();

void output();

};

template <class typeArray>

void tArray <typeArray>::outputTFile(const char\* filename)

{

//функция вывода

//в текстовый файл

ofstream out(filename); //инициализируем поток ввода

/\*если не удалось открыть файл, выводим сообщение и завершаем работу\*/

if (!out)

{

cout << "Невозможно открыть файл " << filename << " !";

exit;

}

for (int i = 0; i < n; i++)

{

out << B[i] << " ";

}

//вывод элементов в файл

}

template <class typeArray>

void tArray <typeArray>::inputTFile(const char\* filename)

{

//функция ввода из

//текстового файла

ifstream in(filename);

if (!in)

{

cout << "Невозможно открыть файл " << filename << " !";

exit;

}

for (int i = 0; i < n; i++)

{

in >> B[i];

}

}

template <class typeArray>

void tArray <typeArray>::input()

{

const int n = 3;

sportsmen B[n];

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

cout << "Введите ID " << i << ": ";

cin >> B[i].id;

cout << "Введите фамилию " << i << ": ";

cin >> B[i].surname;

cout << "Введите результат " << i << ": ";

cin >> B[i].result;

}

}

template <class typeArray>

void tArray <typeArray>::output()

{

//функция ввода из

//текстового файла

for (int i = 0; i < n; i++)

{

cout << B[i].id << " ";

cout << B[i].surname << " ";

cout << B[i].result << " ";

cout << endl;

}

}

class sportsmenArray : public tArray<sportsmen> {

public:

sportsmenArray(int l) : tArray<sportsmen>(l) { };//конструктор

~sportsmenArray() { };

void inputBFile(const char\* filename); //ввод массива

void outputBFile(const char\* filename); //вывод массива

void winner(const char\* filename);

};

void sportsmenArray::outputBFile(const char\* filename)

{

fstream out(filename, ios::out | ios::binary);

if (!out)

{

cout << "Невозможно открыть файл " << filename << " !";

exit;

}

for (int i = 0; i < n; i++)

{

B[i].store(out);

}

out.close();

}

void sportsmenArray::inputBFile(const char\* filename)

{

fstream in(filename, ios::in | ios::binary); //инициализируем поток ввода

if (!in)

{

cout << "Невозможно открыть файл " << filename << " !";

exit;

}

for (int i = 0; i < n; i++)

{

B[i].retrive(in);

}

in.close();

}

void sportsmen::store(fstream& stream)

{ //сохраняем в поток

stream.write((char\*)&id, sizeof(int));//записываем идентификатор

stream.write(surname, 10);//записываем фамилию

stream.write((char\*)&result, sizeof(int));//записываем результат

}

void sportsmen::retrive(fstream& stream)

{ //извлекаем из потока

stream.read((char\*)&id, sizeof(int));//считываем идентификатор

stream.read(surname, 10);//считываем фамилию

stream.read((char\*)&result, sizeof(int));//считываем результат

}

void sportsmenArray::winner(const char\* filename)

{

int n = 3;

char \*min = B[0].result;

for (int i = 0; i < n; i++)

{

if (min > B[i].result)

{

min = B[i].result;

}

}

cout << "Лучший результат: " << min << endl;

}

ofstream& operator<<(ofstream& stream, sportsmen ob)

{

stream << ob.id << " " << ob.surname << " " << ob.result << endl;

return stream;

};

ifstream& operator>>(ifstream& stream, sportsmen& ob)

{

stream >> ob.id; stream >> ob.surname; stream >> ob.result;

return stream;

}

int main()

{

setlocale(LC\_ALL, "Rus");

sportsmenArray C(3);

C.input();//ввод с клавиатуры

C.outputBFile("file.txt");//вывод в текстовый файл

C.inputBFile("file.txt");//ввод из текстового файла

C.output();//вывод на экран

C.winner("file.txt");

return 0;

}

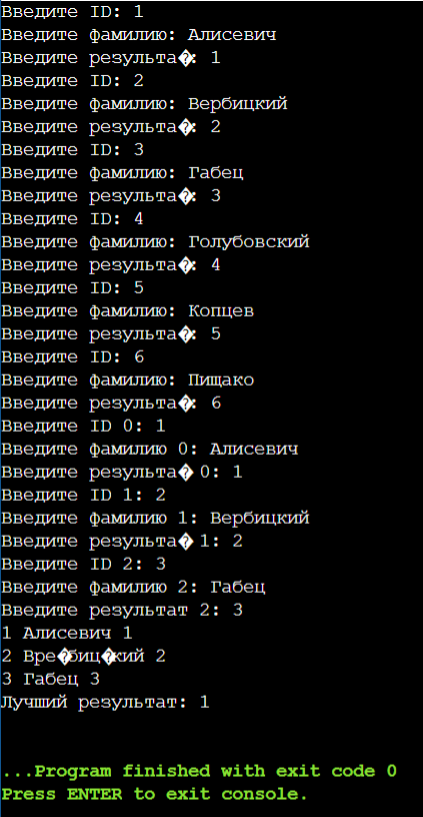


Рисунок 3 – Результат выполнения программы

**Контрольные вопросы:**

1. **В чем различие между текстовыми и бинарными файлами в Си++ с точки зрения структуры и с точки зрения применения?**

Текстовый файл — это файл, в котором данные хранятся в виде символов ASCII, и обычно используется для хранения потока символов. Текстовые файлы организованы вокруг строк, каждая из которых заканчивается символом новой строки (‘\n’). Файлы исходного кода сами по себе являются текстовыми файлами.

Двоичный файл — это файл, в котором данные хранятся в файле так же, как они хранятся в основной памяти для обработки. Он хранится в двоичном формате вместо символов ASCII. Обычно он используется для хранения числовой информации (int, float, double). Обычно двоичный файл может быть создан только внутри программы, и его содержимое может быть прочитано только программой.

1. **Какие функции можно использовать для записи и чтения из бинарных файлов?**

Открытие файлов осуществляется с помощью конструкторов классов ifstream, ofstream и fstream, которые в качестве параметров получают имя файла и режим доступа (может отсутствовать). Для работы с файлами необходимо подключить библиотеку <fstream.h>.

Если мы хотим, использовать бинарные файлы, то для работы с ними воспользуемся функциями:

istream write(char \*адрес переменной, число байт) – запись

ostream read(const char \*адрес переменной, число байт)- чтение.

1. **Каким образом организовать вывод в файл переменной определенного класса?**

Через перегрузку потоков ввода и вывода.

**Вывод:** в ходе выполнения лабораторной работы были сформированы знания и умения по работе с файлами в ЯП Си++, приобретены практические навыки создания программ на языке Си++ с использованием файлового ввода вывода и классов.