

Минобрнауки РФ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тульский государственный университет»

Институт прикладной математики и компьютерных наук  
Кафедра информационной безопасности

Языки программирования  
Отчет по выполнению лабораторной работы № \_

---

Выполнил

---

Проверил

---

Тула 202\_

## Цель лабораторной работы:

---

### Ход выполнения работы

#### Задание на работу:

Объявить структуру для хранения информации о результатах сдачи экзамена: номер группы, фамилия студента, название экзамена, оценка (неуд., уд., хор., отл.). Объявить массив «Табель успеваемости» и поместить в него сведения о студентах из разных групп. Разработать программу, позволяющую выполнить создание новой записи, запись всех данных в файл, чтение табеля из файла, выдачу информации о студентах по фамилии, номеру группы, подсчет среднего балла по выбранному студенту и по выбранной группе.

#### Текст программы.

```
#include <iostream>
#include <stdio.h>
#include <vector>
#include <fstream>
#include <string>
void new_id();
int command_palette();
void print_out();
void RFF();
void WTF();
void print_avr();
void pause();
struct info
{
    int group;
    std::string family;
    std::string examname;
    int res;
};
std::vector<info> studtable;
std::fstream working_file;
void printf_d(info a);
```

```

int main(){
    int y = 1;
    while(y){
        y = command_palette();
        switch (y) {
            case 1:          new_id();          break;
            case 2:          print_out();        break;
            case 3:          print_avr();        break;
            case 4:          WTF();              break;
            case 5:          RFF();              break;
            default:         break;
        }
    }
    return 0;
}

int command_palette(){
    int var;
    printf("\nВариант взаимодействия:\n"
        "1\tСоздание новой записи\n"
        "2\tВыдача информации о студентах по фамилии или номеру
    группы\n"
        "3\tПодсчёт среднего балла по выбранному студенту или
    выбранной группе\n"
        "4\tЗапись информации в файл\n"
        "5\tЧтение информации из файла\n"
        "0\tВыход из программы\n");
    scanf("%d", &var);    getchar();
    return var;
}

void new_id(){
    printf("\nСколько новых записей вы хотите создать?\n");
    int y;
    scanf("%i", &y);
    int a, b;
    char * c = new char[200], * d = new char[200];
    info B;
    printf("\nДля каждого студента введите: номер группы,
    фамилию, название экзамена, результат (2/3/4/5)\n");
    for(int i = 0; i < y; i++){
        scanf("%d%s%s%d", &a, c, d, &b);
        B.family = c;
        B.group = a;
        B.exaname = d;
        B.res = b;
        studtable.push_back(B);
    }
}

```

```

bool exact(int a, int b) {
    bool res = 0;
    while(b) {
        if(a == b) { res = 1; break; }
        b/=10;
    }
    return res;
}

bool exact(char* a, int size_a, std::string b) {
    bool res = 1;    int i;
    for(i=0;i<size_a && i<b.size(); i++) {
        if(a[i]!=b[i]) res = 0; }
    if(i==b.size() && a[i]!=b[i]) res = 0;
    return res;
}

int get_num(char * q, int size) {
    int x=0;
    for(int i=0;i<size;i++) { x = x* 10 + q[i] - 48; }
    return x;
}

void print_out() {
    if(!studtable.size()) {
        printf("Сейчас в памяти нет данных, выводить
нечего.\n");    return;    }
    printf("Введите номер группы или фамилию студента, или
введите ");
    printf("Точку для вывода всех значений:\n");
    char *inp, ch = ' ';    int ssize = 0;
    inp = new char[1000];
    DO_IT_AGAIN;;
    ch = getchar();
    if(ch == '.')    goto PRINT_ALL;
    while(ch != '\n') { inp[ssize++] = ch; ch = getchar(); }
    if(ssize == 0) goto DO_IT_AGAIN;
    if(inp[0]>='0'&&inp[0]<='9') {
        int group_number = get_num(inp, ssize);
        for(auto i: studtable) {
            if(exact(group_number, i.group)){printf_d(i);}
        }
    }
    else {
        for(auto i: studtable) {
            if(exact(inp, ssize, i.family)){printf_d(i);}
        }
    }
    return;
}

```

```

// продолжение функции print_out
PRINT_ALL;;
getchar();
if(studtable.size()>100) {
    printf("Значений больше 100, будет выводиться по
25.\n");
    long long I = 0, R = studtable.size();
    while(I++<R) {
        if(!(I%25)&&I>0) pause();
        printf_d(studtable[I]);
    }
    for(auto i: studtable) printf_d(i);
}
void print_avr(){
    if(!studtable.size()) {
        printf("Сейчас в памяти нет данных, вывести
нечего.\n"); return;
    }
    bool PRT = 0;
    printf("Введите номер группы или фамилию студента, или
введите ");
    printf("точку для вывода среднего значения по всему объёму
данных:\n");
    std::pair<int, int> avr = {0, 0};
    char *inp, ch = ' '; int ssize = 0;
    inp = new char[1000];
    DO_IT_AGAIN;;
    ch = getchar();
    if(ch == '.'){ PRT = 1; goto PRINT_ALL; }
    while(ch != '\n') { inp[ssize++] = ch; ch = getchar(); }
    if(ssize == 0) goto DO_IT_AGAIN;
    if(inp[0]>='0'&&inp[0]<='9') {
        int group_number = get_num(inp, ssize);
        for(auto i: studtable) {
            if(exact(group_number, i.group)) {
                avr.first++;
                avr.second+=i.res;
            }
        }
    }
    else {
        for(auto i: studtable) {
            if(exact(inp, ssize, i.family)) {
                avr.first++;
                avr.second+=i.res;
            }
        }
    }
    PRINT_ALL;;
    if(PRT) { getchar(); avr.first = studtable.size();
        for(auto i: studtable) avr.second+=i.res; }
    printf("Среднее значение: %.6f\n", (float)avr.second /
avr.first);
}

```

```

void RFF(){
    std::fstream FL;    char* FS = new char[200];
    REPEAT_AGAIN;;
    printf("Введите имя файла для чтения (с расширением):\n");
    scanf("%s", FS);    getchar();    FL.open(FS);
    if(!FL.is_open())    {
        printf("Невозможно открыть файл!\nПопытаться открыть
другой? y/n\n");
        char a;
        scanf("%c", &a); getchar();
        if(a=='y' || a=='Y') goto REPEAT_AGAIN;
        return;    }
    if(studtable.size()>0)    {
        printf("Перезаписать существующие данные, ");
        printf("сохранить их в отдельный файл, не открывать
файл, ");
        printf("или добавить к существующим? y/s/n/a\n");
        char a;        scanf("%c", &a);        getchar();
        if(a>100)    a-=115;
        if(a<0)    {
            printf("Операция отменена.\n");    return;    }
        if(!a) {    WTF();    goto REPEAT_AGAIN;    }
        if(a!=97)        studtable.clear();
    }
    long long str, STR;    FL>>str;    STR=str;
    while(str-->0)    {
        info i;
        FL>>i.group>>i.family>>i.exaname>>i.res;
        studtable.push_back(i);
    }
    FL.close();
    printf("Импортировано %lld записей\n", STR);
}

```

```

void WTF(){
    if(!studtable.size())    {
        printf("Сейчас в памяти нет данных, записывать
ничего.\n");    return;    }
    std::fstream FL;    char* FS = new char[200];
    REPEAT_AGAIN;;
    printf("Введите имя сохраняемого файла (с
расширением):\n");
    scanf("%s", FS);    getchar();
    FL.open(FS, std::ios::in);
    if(FL.is_open())    {
        if(!FL.eof())    {
            printf("Похоже, файл не пустой. Перезаписать
данные? y/n\n");
            char a;    scanf("%c", &a); getchar();
            if(a=='n' || a=='N')    goto FILE_IS_NOT_FREE;
        }
    }
    FL.close();
    FL.open(FS, std::ios::out | std::ios::trunc);
    if(!FL.is_open()) printf("Невозможно открыть файл!\n");
    if(!FL.is_open()) {
        FILE_IS_NOT_FREE;;
        printf("Попытаться открыть другой? y/n\n");
        char a;    scanf("%c", &a); getchar();
        if(a=='y' || a=='Y') goto REPEAT_AGAIN;
        return;    }
    FL<<studtable.size();    FL<<std::endl;
    for(auto i:studtable)    {
FL<<i.group<<' '<<i.family<<' '<<i.exaname<<' ' <<i.res
<<std::endl;    }
    FL.close();    printf("Экспортировано %ld записей\n",
studtable.size());}

void pause(){
    char a = 'a';
    printf("Нажмите Enter для продолжения...");
    while(a != '\n')    a = getchar();
}

void printf_d(info a){
    printf("%d %s %s %d\n", a.group, a.family.c_str(),
a.exaname.c_str(), a.res);
}

```

### Описание функций.

| Название функции     | За что отвечает   |
|----------------------|---|
| main                 | Выбор варианта действия с программой  |
| command_palette      | Вывод вариантов действий и ввод ответа пользователя   |
| new_id               | Создание новой записи в "Табеле успеваемости"   |
| print_out            | Вывод записей из "Табеля" по номеру группы или фамилии студента   |
| print_avr            | Вывод средних значений из "Табеля" по номеру группы или фамилии студента  |
| pause                | "Пауза" во время вывода, ждёт, пока пользователь не введёт символ перехода на новую строку  |
| get_num              | Преобразует массив элементов типа char в число  |
| printf_d             | Выводит запись из табеля  |
| RFF (Read From File) | Читает записи из файла  |
| WTF (Write To File)  | Записывает записи в файл  |
| exact                | Сравнивает записи при поиске необходимых для вывода; перегружена – для строк ищется, есть ли в строке искомая подстрока, для чисел – есть ли в начале цифр искомые. |

### Описание структуры данных "info".

| Элемент структуры | Роль   |
|-------------------|--|
| group             | Элемент типа int, содержит число – группу, в которую входит студент. |
| family            | Элемент типа string (строка), содержит фамилию студента.             |
| examname          | Элемент типа string (строка), содержит название экзамена.            |
| res               | Элемент типа int, содержит число – результат студента на экзамене.   |



## **Результаты работы программы.**

### **Содержание файла "test.txt":**

8  
11 Zverev Physics 2  
23 Berdyayev Physics 2  
44 Hlebov Math 2  
44 Nebulov Math 3  
44 Fedurchuk Math 4  
44 Neveselov Physics 4  
23 Karandash Chemistry 3  
44 Neveselov Math 3  
Содержание файла "t2.txt":

3  
12 Gnydil Physics 2  
19 Taganrog Physics 4  
19 Taganrog Math 4

### **Варианты взаимодействия с программой:**

1 Создание новой записи  
2 Выдача информации о студентах по фамилии или номеру группы  
3 Подсчёт среднего балла по выбранному студенту или выбранной группе  
4 Запись информации в файл  
5 Чтение информации из файла  
0 Выход из программы

### **Содержание файла "t3.txt" после работы с программой:**

13  
11 Faust Physics 4  
23 Heist Physics 5  
11 Zverev Physics 2  
23 Berdyayev Physics 2  
44 Hlebov Math 2  
44 Nebulov Math 3  
44 Fedurchuk Math 4  
44 Neveselov Physics 4  
23 Karandash Chemistry 3  
44 Neveselov Math 3  
12 Gnydil Physics 2  
19 Taganrog Physics 4  
19 Taganrog Math 4

### **Вывод программы (удалён вывод вариантов взаимодействия):**

1  
Сколько новых записей вы хотите создать?  
2  
Для каждого студента введите: номер группы, фамилию, название экзамена, результат (2/3/4/5)  
11 Faust Physics 4  
23 Heist Physics 5

2

Введите номер группы или фамилию студента, или введите точку для вывода всех значений:

11

11 Faust Physics 4

2

Введите номер группы или фамилию студента, или введите точку для вывода всех значений:

Heist

23 Heist Physics 5

3

Введите номер группы или фамилию студента, или введите точку для вывода среднего значения по всему объёму данных:

.

Среднее значение: 4.500000

5

Введите имя файла для чтения (с расширением):

test.txt

Перезаписать существующие данные, сохранить их в отдельный файл, не открывать файл, или добавить к существующим? y/s/n/a

a

Импортировано 8 записей

2

Введите номер группы или фамилию студента, или введите точку для вывода всех значений:

23

23 Heist Physics 5

23 Berdyayev Physics 2

23 Karandash Chemistry 3

3

Введите номер группы или фамилию студента, или введите точку для вывода среднего значения по всему объёму данных:

11

Среднее значение: 3.000000

5

Введите имя файла для чтения (с расширением):

t2.txt

Перезаписать существующие данные, сохранить их в отдельный файл, не открывать файл, или добавить к существующим? y/s/n/a

a

Импортировано 3 записей

4

Введите имя сохраняемого файла (с расширением):

t3.txt

Похоже, файл не пустой. Перезаписать данные? y/n

y

Экспортировано 13 записей

0