

# **Пример выполнения задания в консольном режиме**

Написать программу по обработке файла, содержащего информацию о рей-тинге студентов. Каждая запись должна содержать ФИО студента и полученный балл рейтинга.

Вывести информацию, отсортирован-ную в порядке увеличения рейтинга.

Результаты выполнения программы сохранить в текстовом файле.

При работе с файлом должны быть выполнены следующие действия: создание нового файла, просмотр всех записей, добавление новой записи, сортировка.

Для создания текстового файла в консольном приложении используем функцию *fprintf()*. Текст программы:

. . .

```
#include <stdio.h>
```

```
#include <io.h>
```

```
struct TZap {
```

```
    char FIO[30];
```

```
    double s_b;
```

```
} Zap;
```

- Шаблон

структуры

- Структурная переменная

`int size = sizeof(TZap);`      - Размер записи

`FILE *Fz, *Ft;`                      - Указатели файлов

`char File_Zap[] = "zapisi.dat";`

`// char File_Zap[20];`

`char File_Rez[] = "result.txt";`

`// char File_Rez[20];`

`void Out(TZap);`                      - Прототип функции

```
void main()
```

```
{
```

```
    int kod,      - Для выбора действия
```

```
    i, j,        - Переменные циклов
```

```
    kol;         - Количество записей
```

```
    long len;    - Объем файла
```

```
    TZap st,     - Элемент структуры
```

```
    *mas_Z;      - Указатель для массива
```

```
        Ft = fopen(File_Rez, "w"); - Открытие  
текстового файла для сохранения результатов
```

```
while (true) {    - Бесконечный цикл
    puts("\n Create – 1\n Add – 2
        \n View – 3\n Sort – 4
        \n EXIT – 0");
    scanf("%d", &kod);
```

Если добавить пункт: «Введите имя файла», то можно в процессе выполнения программы выбирать нужный файл.

```
switch(kod) {  
    case 1:                - Создание нового файла  
        Fz = fopen(File_Zap, "wb");    - Открытие  
        if (Fz == NULL) {            - Проверка  
            puts("\n Create ERROR!");  
            return;  
        }  
        fclose(Fz);                - Заккрытие файла  
        printf("\n Create New File %s !\n",  
            File_Zap);  
    break;                - Конец case 1
```

case 2:                - Добавление новой записи

Fz = fopen(File\_Zap,"ab");    - Открытие

printf("\n FIO – ");

fflush(stdin); - Очистка потока ввода

gets(Zap.FIO);

printf(" Ball – ");

scanf("%lf", &Zap.s\_b);

fwrite(&Zap, size, 1, Fz);    - Запись в файл

fclose(Fz);                    - Заккрытие файла

break;                        - Конец case 2



case 3:

```
if ( (Fz = fopen(File_Zap,"rb") ) == NULL) {
```

- Открытие файла для чтения с проверкой

```
puts("\n Open ERROR!");
```

```
return;
```

```
}
```

```
printf("\n\t----- Informations -----");
```

- Вывод текста с новой строки с табуляцией на экран

```
fprintf(Ft, "\n\t----- Informations -----");
```

- Запись текста с новой строки с табуляцией в файл Ft

`while(1) {`      - Бесконечный цикл

`// if ( ! fread(&Zap,size,1,Fz) ) break;`

`if ( fread(&Zap,size,1,Fz) == 0) break;`

- Считываем из файла Fz по одной записи Zap до тех пор пока не дойдем до конца файла, т.е. пока результат функции fread не равен 0

`Out(Zap);`      - Вызов функции для вывода информации на экран и записи в текстовый файл

`}`      - Конец цикла `while(1)`

`fclose(Fz);`      - Заккрытие файла

`break;`      - Конец `case 3`

case 4:            - Сортировка по рейтингу

    Fz = fopen(File\_Zap,"rb");    - Открытие

len = filelength(fileno(Fz));    - Опреде-ление  
объема файла в байтах

    kol = len / size;    -    Расчет        количества  
записей в файле

    mas\_Z = new TZap[kol]; -            Создание  
динамического массива

    fread(mas\_Z, size, kol, Fz); - Чтение kol  
записей каждая размером size байт из  
файла Fz и запись по адресу mas\_Z

    fclose(Fz);        - Закрытие файла

`printf("\n\t----- S O R T ----- \n");`      - Вывод  
текста на экран

`fprintf(Ft, "\n\t----- S O R T ----- \n");`    - Запись  
такого же текста в файл результатов Ft

`for ( i = 0; i < kol - 1; i++)`      - Сортировка

`for ( j = i+1; j < kol; j++)`

`if (mas_Z[i].s_b > mas_Z[j].s_b) {`

`st = mas_Z[i];`      - Перестановка

`mas_Z[i] = mas_Z[j];`

`mas_Z[j] = st;`

`}`

for (i = 0; i < kol; i++) - Вывод всех

Out(mas\_Z[i]); записей

delete []mas\_Z;

break; - Конец case 4

case 0: - Конец работы

fclose(Ft); - Заккрытие текстового файла

return; - Выход из функции main

} - Конец оператора switch

} - Конец оператора while(true)

} - Конец функции main

## Функция вывода одной записи на экран и записи в файл

```
void Out ( TZap z )
```

```
{
```

```
    printf ( "\n %20s , %6.3lf ", z.FIO, z.s_b);
```

- Вывод на экран полей FIO и s\_b записи z

```
    fprintf ( Ft, "\n %20s , %6.3lf ", z.FIO, z.s_b);
```

- Запись в текстовый файл Ft аналогичной информации

```
}
```

Выбрав пункт «1», открываем файл

```
Fz = fopen(File_Zap, "wb");
```

с логическим именем Fz (указатель) с физическим именем File\_Zap равным в нашем примере *zapisi.dat* (массив char), который будет располагаться в текущей рабочей папке.

Режим открытия wb – для создания не существующего бинарного файла, или замены существующего файла на новый со стиранием предыдущей информации.

Выбирая пункт «2», открываем файл

```
Fz = fopen(File_Zap,"ab");
```

- аналогично файл с указателем Fz и именем File\_Zap.

Но режим открытия ab – для создания первой записи у не существующего бинарного файла, или добавления новой записи в конец существующего файла.

Выбирая пункт «2» добавляем столько записей, сколько надо. В этом пункте можно поставить цикл.



Выбрав пункт «3», просматриваем содержимое файла и записываем в результирующий текстовый файл Ft.

Сортированные записи выведем на экран и запишем в файл, выбрав пункт «4».

При работе пунктов «3» и «4» файл открываем для чтения

```
Fz = fopen(File_Zap,"rb");
```

Режим открытия rb – для чтения информации из существующего бинарного файла, при его отсутствии – ошибка.