

ЛАБОРАТОРНАЯ РАБОТА № 32

ДОБАВЛЕНИЕ, УДАЛЕНИЕ ДАННЫХ ИЗ ФАЙЛА

Цель работы: получение практических навыков по добавлению и удалению данных из файла.

Краткие теоретические сведения

Создать программу, в которой реализованы создание, добавление и просмотр файла, содержащего информацию о фамилии и среднем балле студентов. Процесс добавления информации заканчивается при нажатии точки.

```
#include <stdio.h>
#include <stdlib.h>
struct Sved {
    char Fam[30];
    double S_Bal;
} zap,zapt;
char Spis[]="c:\\work\\Sp.dat";
FILE *F_zap;
FILE* Open_file(char*, char*);

void main (void)
{
    int i, j, kodR, size = sizeof(Sved), kod_read;
    while(1) {
        puts("Создать – 1\n Добавить– 3\nПросмотреть– 2\nВыход – 0");
        scanf("%d",&kodR);
        switch(kodR) {
            case 1:
            case 3:
                if(kodR==1) F_zap = Open_file (Spis,"w+");
                else F_zap = Open_file (Spis,"a+");
                while(2) {
                    puts("\n Fam (. – end) ");
```

```

        scanf("%s",zap.Fam);
        if((zap.Fam[0])=='.') break;
        puts("\n Ball: ");
        scanf("%lf",&zap.S_Bal);
        fwrite(&zap,size,1,F_zap);
    }
    fclose(F_zap);
    break;
case 2: F_zap = Open_file (Spis,"r+"); int nom=1;
while(2) {
    if(!fread(&zap,size, 1, F_zap)) break;
    printf(" %2d: %20s %5.2lf\n",
        nom++, zap.Fam, zap.S_Bal);
}
fclose(F_zap);
break;
case 0: return;           // exit(0);
}                          // Закрывает switch()
}                          // Закрывает while()
}
// Функция обработки ошибочной ситуации при открытии файла
FILE* Open_file(char *file, char *kod)
{
FILE *f;
    if(!(f = fopen(file, kod))) {
        puts("Open File Error!");
        exit(1);
    }
    return f;
}

```

Советы по программированию

При выполнении вариантов заданий придерживайтесь следующих ключевых моментов.

1. Объекты типов структуры и объединения применяются для логически связанных между собой данных различных типов.

2. После описания шаблона структурного типа данных ставится точка с запятой.

3. Элементы данных, входящие в структуры и объединения, называются полями. Поля могут быть любого базового (стандартного) типа данных, массивом, указателем, объединением или структурой.

4. Для обращения к полю используется операция принадлежности (привязки, выбора) «.» (точка) при обращении через *ID* структуры, или «->» (стрелка) при обращении через указатель.

5. Структуры одного типа можно присваивать друг другу с использованием стандартной функции *memcpy*.

6. Ввод-вывод структур выполняется поэлементно.

7. Структуры, память под которые выделяет компилятор, можно инициализировать значениями их полей.

8. Файл – это именованный объект, хранящий данные на каком-либо носителе, хотя может располагаться и на электронном диске в ОП.

9. Файл не имеет фиксированной длины, т.е. может увеличиваться или уменьшаться в процессе обработки.

10. Перед работой файл необходимо открыть (функция *fopen*), а после работы закрыть (функция *fclose*).

Порядок выполнения работы

1. Изучить теоретические сведения.

2. Выполнить задание.

Задания для выполнения

1. Набрать и выполнить пример.

2. Переделать программу, используя функции для создания, добавления и просмотра данных.