

Навчальна практика

```
struct [<ім'я структури>]  
{  
    <тип 1> ім'я поля 1;  
    <тип 2> ім'я поля 2 . . . ;  
};
```

де *struct* - службове слово;
<ім'я структури> - ім'я типу структура;
<тип 1>, *<тип 2>* - тип даних;
ім'я поля 1, ім'я поля 2,... - імена полів структури;

Структура

Приклад 1:

```
#include <stdio.h>
#include <stdlib.h>
struct ZrPlat{
    char family[20];
    int mis;
    int summa;
};
int main()
{ //Масив структур
  struct ZrPlat z[6] =
  { {"Ivanov", 1, 20000},
    {"Petrov", 1, 15000},
    {"Fedorov", 1, 4500},
    {"Bubnov", 1, 15500},
    {"Sidorov", 2, 12350},
    {"Vasichkin", 2, 8000}
  };
  //Виводимо масив структур на екран
  printf("\tFamily:\t Mis:  Summa: \n");
  for (int i=0; i<6; i++)
    printf("%15s %3d %8d\n",z[i].family, z[i].mis, z[i].summa);
  return 0;
}
```

Family:	Mis:	Summa:
Ivanov	1	20000
Petrov	1	15000
Fedorov	1	4500
Bubnov	1	15500
Sidorov	2	12350
Vasichkin	2	8000

Приклад 2:

```
#define _CRT_SECURE_NO_WARNINGS
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#define n 4
#define k 3
int main(){
    struct ZACHETKA{
        char last_name[30];
        int no;
        int oценка[k];
    };
    ZACHETKA* box = (ZACHETKA*)calloc(n, sizeof(ZACHETKA));
    ZACHETKA tmp;
    for (int i=0; i<n; i++) {
        printf("Vvedite Familiy\n");
        scanf("%s", &box[i].last_name);
        printf("Vvedite nomer\n");
        scanf("%d",&box[i].no);
        printf("Vvedite oцenki (%d)\n", k);
        for(int j=0; j<k; j++)
            scanf("%d",&box[i].ocenka[j]);
    }
}
```

```
for(int i=n-1; i>0 ; i--)// Сортування за прізвищем
    for(int j=0; j<i ; j++)
        if(strcmp( box[ j ].last_name, box[ j+1 ].last_name ) > 0)
            {
                tmp = box[ j ];
                box[ j ]= box[ j+1 ];
                box[ j+1 ]= tmp;
            }
for(int i=0; i<n; i++) //Виводимо масив структур на екран
{
    printf("\n%s ",box[i].last_name);
    printf("%d ",box[i].no);
    for(int j=0; j<k; j++)
        printf("%d ",box[i].ocenska[j]);
}
return 0;
}
```

Базові функції для роботи з файлами описані в бібліотеці **stdio.h**. Вся робота з файлом виконується через файлову змінну - покажчик на структуру типу FILE:

FILE * fp;

Відкрити файл можна функцією `fopen`, що має 2 параметри:

`FILE * fopen (char * ім'я файлу, char * режим доступу);`

Параметр *режим_доступу* визначає, які дії будуть дозволені з файлом:

“r” - відкриваємо для читання текстовий файл;

“r + b” - відкриваємо для довільного доступу (читання і запис) бінарний файл;

“a” - відкриваємо текстовий файл для додавання даних в кінець файлу;

“w” - відкриваємо файл для запису.

Робота з файлом

Назва	Що робить
fopen()	Відкриває файл
fclose()	Закриває файл
putc()	Записує символ у файл
fputc()	Записує символ у файл
getc()	Читає символ з файлу
fgetc()	Читає символ з файлу
fgets()	Читає рядок з файлу
fputs()	Записує рядок у файл
fseek()	Встановлює покажчик поточної позиції на певний байт файлу
ftell()	Повертає поточне значення покажчика у файлі
fprintf()	Для файлу те саме, що printf() для консолі
fscanf()	Для файлу те саме, що scanf() для консолі
feof()	Повертає значення true (істина), якщо досягнуто кінець файлу
error()	Повертає значення true, якщо виникла помилка
rewind()	Встановлює покажчик поточної позиції на початок файлу
remove()	Знищує файл
fflush()	Дозапис потоку у файл

Робота з файлом

Приклад. Перевірка, чи вдалося відкрити файл з поточної папки

```
#define _CRT_SECURE_NO_WARNINGS
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
int main() {
    SetConsoleCP(1251);SetConsoleOutputCP(1251);
    FILE* fp; char buf[]="myFile.txt";
    fp = fopen(buf, "a");
    if (fp == NULL) {
        printf("\n не вдалося відкрити файл\n");system("pause");
        exit(1);// Вийти з кодом завершення 1
    }
    else printf("\nфайл відкрито\n");
    fputs("Zhytomyr is a city in the north of the western half of
    Ukraine\n", fp);
    fclose(fp);
    fp = fopen(buf, "r");
    char str[100];
    while (!feof(fp)) { fgets(str, 100, fp);
        printf("\n%s", str);
    }
system("pause");return 0;
}
```

Приклад виконання завдання:

```
#include <stdio.h>
#include <stdlib.h>
int N;

struct NNN {
    int n;
    char name[10];
    char sity[10];
    float den;
}*buf = NULL;

void Buff(FILE *f) {
    fseek(f, 0, SEEK_END);
    N = ftell(f) / sizeof(NNN);
    rewind(f);
    buf = (NNN*)malloc(N * sizeof(NNN));
    fread(buf, sizeof(NNN), N, f);
}
```

```
void InPut() {
    NNN *buf1 = (NNN*)malloc((N + 1) * sizeof(NNN));
    for (int i = 0; i < N; i++)
        buf1[i] = buf[i];
    free(buf);
    buf = buf1;
    printf("\n%d\n", N);
    printf("\n n:");
    scanf("%d", &buf[N].n);
    printf("\nname: ");
    scanf("%s", &buf[N].name);
    printf("\nsity: ");
    scanf("%s", &buf[N].sity);
    printf("\nden:  ");
    scanf("%f", &buf[N].den);
    N++;
}
void OutPut() {
    for (int i = 0; i < N; i++)
        printf("\n%d\t%s\t%s\t%4.2f", buf[i].n, buf[i].name,
            buf[i].sity, buf[i].den);
}
```

```
void Menu() {
    int menuM = 1;
    do
    {
        printf("\n0-exit\n1-input\n2-output\n");
        scanf("%d", &menuM);
        switch (menuM)
        {
            case 1: InPut();break;
            case 2: OutPut();break;
            default:break;
        }
    } while (menuM != 0);
}
```

```
int main() {
    FILE *f;
    if ((f = fopen("text", "ab")) == NULL)
    {
        printf("ERROR\n");
        return 1;
    }

    Buff(f);
    fclose(f);

    Menu();

    f = fopen("text", "wb");
    for (int i = 0; i < N; i++)
        fwrite(&buf[i], sizeof(NNN), 1, f);
    fclose(f);
    return 0;
}
```
