

Лекція 21. Структури

Структура - це тип даних, який створюється програмістом і призначений для об'єднання даних різних типів в єдине ціле.

Оголошення структури здійснюється за допомогою ключового слова *struct*.

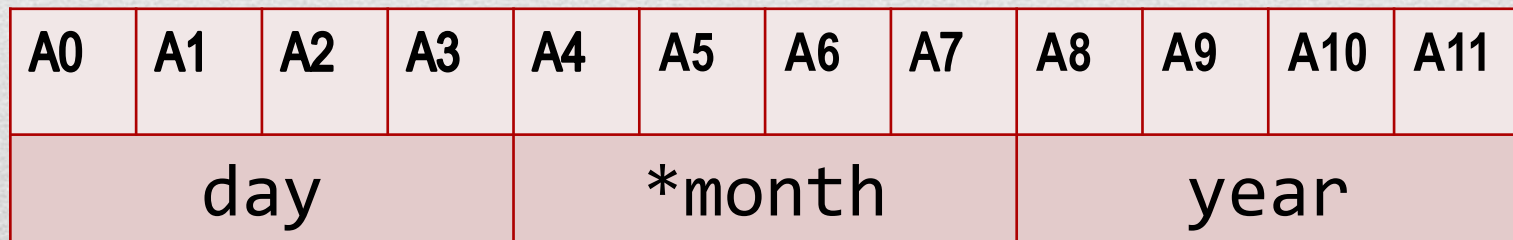
Синтаксис опису структури виглядає наступним чином :

```
struct [ім'я_структури]
{
    тип1 елемент1;
    тип2 елемент2;
    . . . . .
    типN елементN;
} [список описів];
```

Елементи структури називаються **полями**, можуть мати будь-який тип, в тому числі бути покажчиками на тип самої структури.



```
struct date
{
    int day;           // 4 байта
    char *month;      // 4 байта
    int year;         // 4 байта
};
```



Ініціалізація полів структури може здійснюватися двома способами:

- присвоювання значень елементам структури в процесі оголошення змінної, що відноситься до типу структури;
 - присвоювання початкових значень елементів структури з використанням функцій вводу-виводу (наприклад, `printf ()` і `scanf ()`).
-

ПРИКЛАД №1

```
struct book {  
  
int date=1998; // дата видання  
int pages=250; // кількість сторінок  
char author[50]="D. Ritchi"; // автор  
char title[100]= "The C programming language I"; //назва книги  
float price=740.5; // вартість  
  
};  
  
int main(){  
struct book kniga;  
printf("%s-%s %d page\n", kniga.author, kniga.title, kniga.pages);  
return 0;  
}
```

```
D. Ritchi-The C programming language I 250 page  
Для продовження натисніть будь-яку клавішу . . .
```

Звернення до певного члена структури
проводиться за допомогою операції точка:

<ім'я об'єкту що представляє структуру>.<ім'я елемента>

`kniga.author`

ПРИКЛАД №2

```
struct book {
int date;           // дата видання
int pages;          // кількість сторінок
char author[50];    // автор
char title[100];    // назва книги
float price;        // вартість
};

int main(){

struct book kniga1 = { 1998,230,"D. Ritchi","The C programming
language.",140.2 };

                                або
struct book kniga1 = { 1998, 230, "D. Ritchi" };

kniga1.pages = 250;

printf("%s-%s %d page",kniga1.author, kniga1.title, kniga1.pages);
return 0;
}
```

```
#define n 2
```

ПРИКЛАД №3

```
struct book {  
int date;           // дата видання  
int pages;         // кількість сторінок  
char author[50];   // автор  
char title[100];  // назва книги  
float price;       // вартість  
};  
  
int main(){  
    struct book kniga[] = {1998, 250, "D. Ritchi", "The C  
programming language I", 540.2, 1999, 350, "D. Ritchi", "The C  
programming language II", 740.5};  
  
    for (int i = 0; i < n; i++)  
        printf("%s-%s %d page\n", kniga[i].author, kniga[i].title,  
kniga[i].pages);  
return 0;  
}
```

Поля структур можуть мати довільний тип, як простий, так і складений. Зокрема поле може бути масивом, символьним рядком, або вкладеною структурою.

Для **вкладених структур** діють такі самі правила оголошення, як і для звичайних структур.

```
struct point {  
  
    double x;  
    double y;  
};  
  
struct RGB {  
  
    int red, green, blue;  
};  
  
struct circle {  
  
    struct point center;  
    struct RGB color;  
    double radius;  
  
};
```

Ініціалізацію структурної змінної типу `struct circle` можна виконати наступним чином:

```
struct point {
    double x;
    double y;
};

struct RGB {
    int red, green, blue;
};

struct circle {
    struct point center;
    struct RGB color;
    double radius;
};
```

```
struct circle c = {
    { 2.1, 3.2 },
    { 255, 255, 255 },
    1.2
};
```

ПРИКЛАД №4

```
#include <stdio.h>
#include <stdlib.h>
struct date {
int day;
char month[20];
int year;
};
struct persone {
char firstname[20];
char lastname[20];
struct date bd;
};
int main() {system("chcp 1251"); system("cls");
struct persone p;
printf("Введіть і'мя : "); scanf("%s", p.firstname);
printf("Введіть прїзвище : "); scanf("%s", p.lastname);
printf("Введіть дату народження \nЧисло: ");
scanf("%d", &p.bd.day);
printf("Місяць: "); scanf("%s", p.bd.month);
printf("Рік: "); scanf("%d", &p.bd.year);
printf("\n : %s %s, дата народження %d %s %d p.",
p.firstname, p.lastname, p.bd.day, p.bd.month, p.bd.year);
return 0;
}
```

```
#define l 20
```

```
struct names { /*перший шаблон*/
```

```
char im[l];
```

```
char prizv[l];
```

```
};
```

```
struct harakter { /*другий шаблон*/
```

```
struct names druzi; /*вкладена структура*/
```

```
char bludo[l];
```

```
char robota[l];
```

```
float zarob;
```

```
};
```

```
int main(){
```

```
system("chcp 1251");system("cls");
```

```
struct harakter x1={ {"Іван", "Петренко"}, "вареники", "інженер",  
30250.00};
```

```
printf("%s ", x1.druzi.im);
```

```
printf("%s ", x1.druzi.prizv);
```

```
printf("%s ", x1.robota);
```

```
printf("%s ", x1.bludo);
```

```
printf("%f ", x1.zarob);
```

```
return 0;
```

```
}
```

ПРИКЛАД №5

Створити та вивести на екран базу даних з 5 студентів. Полями є: прізвище; ім'я; день, місяць та рік народження, адреса (дані вводяться з клавіатури)

```

#define N 5
struct person {
    char fam[15], name[15]; int day, month, year; char address[50];
};
void input(person *stud, int n);          void output(person *stud, int n);
int main() { system("chcp 1251"); system("cls");
struct person *student;
student = (struct person*)malloc(N*sizeof(struct person));
input(student, 2); output(student, 2);
free(student); system("pause"); return 0;
}
void input(person *stud, int n) {
    for (int i = 0; i < n; i++) {
        printf("\tЗапис%d :\n", i + 1);
        printf("Прізвище - ");      scanf("%s", stud[i].fam);
        printf("Ім'я - ");          scanf("%s", stud[i].name);
        printf("День народж. - ");  scanf("%d", &stud[i].day);
        printf("Місяць - ");        scanf("%d", &stud[i].month);
        printf("Рік - ");           scanf("%d", &stud[i].year);
        printf("Адреса - ");        scanf("%s", stud[i].address);
    }
}
void output(person *stud, int n) {
    for (int i = 0; i < n; i++)
        printf("\tЗапис %d %s %s %d %d %d %s\n", i + 1, stud[i].fam, stud[i].name,
        stud[i].day, stud[i].month, stud[i].year, stud[i].address);
}

```

```
#define n 3
struct book{
    char title[15]; char author[15]; int value;
};
int main(){
    system("chcp 1251"); system("cls");
    struct book *lib;
    lib = (struct book*)malloc(n * sizeof(struct book));
    for (int i = 0; i < n; i++){
        printf(«Назва %d книги : ", i + 1);
        gets_s((lib + i)->title);
        printf(«Автор %d книги : ", i + 1);
        gets_s((lib + i)->author);
        printf(«Ціна %d книги : ", i + 1);
        scanf_s("%d", &(lib + i)->value);
        getchar();
    }
    for (int i = 0; i < n; i++){
        printf("\n %d. %s ", i + 1, (lib + i)->author);
        printf("%s %d", (lib + i)->title, (lib + i)->value);
    }
    system("pause"); return 0;
}
```

Структуру, оголошену в глобальному контексті, видно всім.
Структура також може бути оголошена всередині функції:

```
int main() {
    struct point_t {
        int x;
        int y;
    };
    struct point_t A;
    float dist;

    A.x = 10;
    A.y = 20;

    dist = sqrt((float)(A.x*A.x + A.y*A.y));

    printf("x = %.3f", dist);
    return 0;
}
```

Синтаксис мови дозволяє створювати екземпляри структури відразу ж після визначення:

```
struct point_t {  
    int x;  
    int y;  
};  
struct point_t A;
```

```
struct point_t {  
    int x;  
    int y;  
} A;
```