

Дисциплина «Основы алгоритмизации и
программирование»

Специальность 2-40 01 01

ПОИТ

Структуры



Белокопыцкая Юлия
Александровна,
преподаватель

Определение структуры

Структура –

Элементы структуры называются полями, могут иметь любой тип, в том числе быть указателями на тип самой структуры.

Общий вид структуры

Объявление типа ***struct*** имеет вид:

struct имя_типа {описание элементов};

Пример:

```
struct book  
{  
    char title[81];  
    char author[13];  
    int year;  
    int page;  
    float price;  
};
```

Поля структуры

- Имена полей в структуре должны различаться.
- Имена элементов разных структур могут совпадать.
- Элементом структуры может быть другая структура.

```
struct pets { char name[10];  
                int age };  
struct boy { char name[10];  
                int age;  
                pets pet };
```

Поля структуры

- Открытого типа – `public`;
- Закрытого типа – `private`.

Тип *struct* является видом класса и обладает всеми свойствами классов. Все поля класса *struct* по умолчанию открыты (*public*), но их можно переопределить как закрытые (*private*).

Объявление переменных

Объявление переменной (то есть объект) типа book:

struct book library;

создает объект типа ***struct book***, под который выделяется 110 байт памяти.

Объявление переменных

Одновременно с объявлением структуры можно объявить одну или несколько переменных:

```
struct book  
{  
    char title[81];  
    char author[13];  
    int year;  
    int page;  
    float price;  
} library , catalog[10], *plibrary;
```

```
struct  
{  
    char title[81];  
    char author[13];  
    int year;  
    int page;  
    float price;  
} library;
```

Доступ к членам структуры

Элементы структуры в памяти запоминаются последовательно, в том порядке, в котором они объявляются.

- Структурную переменную можно инициализировать в операторе описания подобно массиву:

```
struct book library = {"Язык C++,  
Страуструп", 1990, 500, 1000 };
```

Доступ к членам структуры

- Доступ к элементам структуры осуществляется с помощью операции точка:

```
library.author="Павловская"; // явная  
инициализация
```

```
gets (library.author); //ввод значения  
strcpy(library.author, "Шилдт");
```

Размер структуры

Используется функция `sizeof()`:

Синтаксис:

`sizeof(имя структуры)`

Пример:

```
int size_library = sizeof(library);  
// size_library = 110 байт
```

Копирование структур

- Функция `memcpy`:

memcpy (адрес назначения, адрес источника, размер);

book library2;

memcpy(& library2, & library, sizeof(book));

- Присваивание одной структуры другой:

library2 = library;

Массивы структур

- Описание массива структур аналогично описанию любого другого массива:

struct book catalog[10];

- Каждый элемент массива *catalog* представляет собой структуру типа *book*.

Доступ к элементу массива

Для доступа к элементу массива используется индекс, который присоединяется к имени массива:

catalog[2].title

catalog[4].price

catalog[2].title[5] //6 элемент

символьного массива в 3-й структуре

Вложенные структуры

Элементом структуры может быть другая структура.

```
struct myfile {  
    char name[10];  
    char ftype[4];  
    int    ver;  
};  
struct dir {  
    struct myfile f;  
    int          size;  
} my_f [100];
```

Вложенные структуры

Шаблон для вложенной структуры должен располагаться перед определением фактической структурной переменной в рамках другой структуры.

my_f [0].size //элемент size 1-ой структуры

my_f [2].f.ver //элемент ver вложенной структуры f в 3-й структуре my_f

Структуры и функции

Для передачи информации о структуре внутри функции используются следующие способы:

- Использование в качестве фактического аргумента элемента структуры.
- Использование в качестве фактического аргумента адреса структуры.
- Использование в качестве фактического аргумента самой структуры.

Структуры и функции

Каждый член структуры передается функции:

func(library.title[81]); / передается символьное значение title[81] */*

func2(library.year); / передается целое значение year */*

func3(library.price); / передается значение с плавающей точкой price */*

func4(library.title); / передается адрес строки s */*

Структуры и функции

Когда в качестве аргумента функции используется структура, то для передачи целой структуры используется обычный способ вызова по значению.

```
void f1(struct book parm);  
int main() {  
    struct book library;  
    library.year = 1000;  
    f1(library);  
    return 0; }
```

Указатели на структуры

Указатели на структуры используются:

- структура передается функции с помощью ссылки;
- создаются структуры с динамическими данными, работающие на основе динамического размещения.

Объявим указатель на структуру:

```
struct book *pst;
```

Адрес структуры *library* можно присвоить указателю *pst*:

```
pst = &library;
```

```
ps= &catalog[0];
```

```
p=&my_f[0];
```

Указатели на структуры

Доступ к полям структуры через указатель осуществляется с помощью операции косвенного доступа \rightarrow

$pst \rightarrow page = 100;$

$pst \rightarrow page == library.page$

$(ps+i) \rightarrow page == catalog[i].page$

Для вложенных структур:

$my_f[0].f.ver == p \rightarrow (f.ver)$

Спасибо за внимание

