

# Structures

# Понятие структуры

Помимо стандартных типов, программист может создавать свои собственные составные типы данных - структуры.

**Структура — это множество, состоящее из одного или более объектов, возможно имеющих различные типы, объединённых под одним именем.** Конструкция `struct` — один из основных строительных блоков данных в языке. Он предоставляет удобный способ объединения различных элементов, связанных между собой логической связью. Рассмотрим особенности работы со структурами на примере:

# Пример работы со структурами

<https://git.io/vbr8Q>

# Особенности структур

- Описание структуры начинается со служебного слова **struct**, за которым может следовать необязательное имя, называемое именем типа структуры. Это имя типа структуры используется в дальнейшем для создания конкретного объекта
- За именем типа структуры идёт заключенный в фигурные скобки список элементов структуры, с описанием типа каждого элемента (элементом структуры может быть переменная, массив или структура). Элементы структуры отделяются друг от друга точкой с запятой

# Особенности структур

- За правой фигурной скобкой, закрывающей список элементов, может следовать список объектов. Например, оператор `struct date {...} x,y,z;` определяет переменные `x,y,z` в качестве структур описанного типа и приводит к выделению памяти
- Описание структуры, за которым не следует список объектов, не приводит к выделению памяти - только определяет форму структуры. Если описание снабжено именем типа, то это имя типа может быть использовано позднее при определении объектов структур

# Особенности структур

- Структуру можно инициализировать, поместив вслед за её определением список инициализаторов для её компонентов, заключенный в фигурные скобки. В программе таким образом инициализирована структура:  
**date my\_birthday = {20,7,1981,"July"};**
- Обращение к определённому члену структуры производится с помощью конструкции вида:  
**<имя структуры>.<имя элемента>**
- Структуры могут вкладываться одна в другую, но самовложение структур запрещено!

# Пример вложенности

```
struct date {  
    int day; // День  
    char month[10]; // Месяц  
    int year; // Год  
};
```

```
struct person {  
    char name[50]; // Имя, фамилия, отчество  
    char address[10]; // Домашний адрес  
    int salary; // Зарплата  
    date birthdate; // Дата рождения.  
    date hiredate; // Дата поступления на работу  
};
```

# Списковая инициализация

```
struct Date
{
    unsigned int day;
    unsigned int month;
    int year;
    char weekday[10]; // название дня недели
};
```

```
Date today {6, 7, 2013, "Saturday"}; // знак = можно пропустить
Date yesterday {}; // все поля установлены в 0
// а ещё все байты weekday - тоже по нулям!
```

# Использование вложенности

Если определить **struct person Nick;** то `Nick.birthdate.month` будет обозначать месяц рождения. Операция доступа к элементу структуры «.» вычисляется слева направо.

# Действия над структурами

- Доступ к элементу структуры с помощью операции «.»
- Доступ к элементу структуры по указателю с помощью операции «->»
- Определение адреса структуры с помощью операции «&»
- Присваивание структуры как единого целого
- Передача структуры в качестве параметра функции и возвращение структуры в результате работы функции

# Операция sizeof

В языке C существует специальная унарная операция **sizeof**, которая возвращает размер своего операнда в байтах. Операндом операции sizeof может быть любое выражение:

```
sizeof(Выражение);
```

Результат операции sizeof имеет тип int.

# Размер объектов структур

Может показаться, что размер структуры равен сумме размеров её компонентов. Это не так. Вследствие выравнивания объектов разной длины в структуре появляются безымянные «дыры». Так, например, если переменная типа `char` занимает один байт, а `int` — четыре байта, то для

```
struct Test {  
    char c;  
    int i;  
};
```

может потребоваться 8 байт, а не 5.

Правильное значение возвращает операция `sizeof`.

# Практика

Создать структуру ВИДЕОКАТАЛОГ со следующими полями:

- Название фильма
- Режиссер
- Жанр (может быть одновременно несколько)
- Рейтинг популярности

Реализовать следующие возможности:

- Поиск по названию
- Поиск по жанру
- Поиск по режиссеру
- Самый популярный фильм в жанре
- Показ всех записей и добавление