

# Алгоритмизация и программирование

Программирование на C/C++  
(ч.13 - структуры)

Беркунский Е.Ю., кафедра ИУСТ, НУК  
[eugeny.berkunsky@gmail.com](mailto:eugeny.berkunsky@gmail.com)  
<http://berkut.homelinux.com>

# Структуры – а что это?

Структура — это , некое объединение различных переменных (даже с разными типами данных), которому можно присвоить имя.

Например данные об объекте **Дом**:

- город (в котором дом находится),
- улица,
- количество квартир,
- интернет(проведен или нет)
- и т.д. в одной структуре.

В общем, можно собрать в одну совокупность данные обо всем, что угодно, точнее обо всем, что необходимо конкретному программисту.

# Объявление структуры

- Структура — это агрегатный тип данных, так как может содержать в себе разнотипные элементы.
- Синтаксис объявления структуры в C++ отличается от C.
- Версия C остается правильной и для C++.
- Следовательно, в C++ можно пользоваться двумя стилями объявления структур, а в языке C — только одной.

# Объявление структуры

```
struct Name  
{  
    type atrib;  
    // остальные элементы структуры  
} structVar1, structVar2, ...;
```

- **struct** — ключевое слово - начинает определение стр-ры
- **Name** — имя структуры
- **type** — тип данных элемента структуры
- **atrib** — элемент структуры
- **structVar1-2** — структурные переменные

# Объявление структуры

Так как структура это тип данных, то, для того, чтобы использовать этот тип данных, необходимо объявить структурную переменную, а в качестве типа данных указать имя структуры.

```
struct_name structVariable;
```

## Синтаксис объявления структуры в языке Си

```
typedef struct name  
{  
    type atrib1;  
    type atrib2;  
    // остальные элементы структуры...  
} newStructName structVar;
```

# Объявление структуры

Синтаксис объявления структуры в языке Си предполагает два варианта.

Первый, опустить ключевое слово `typedef`, при этом имя `newStructName` тоже не используется, и имя структуры, тогда обязательно необходимо при объявлении структуры использовать структурные переменные — `structVar`

```
struct name structVar;
```

Или использовать `typedef`, для объявления псевдонима структуры `newStructName`, псевдоним

```
newStructName structVar;
```

# Объявление указателя на структуру

В Си, если вы не используете `typedef` при определении структуры, то, в обязательном порядке необходимо использовать структурные переменные, между закрывающейся фигурной скобкой и точкой с запятой.

***В С++, этого не требуется.***

Чтобы объявить указатель на структуру, в С++ вы просто перед именем структурной переменной ставите символ указателя — `*`.



# Объявление указателя на структуру

**В C++:**

```
structName *structVar;  
// указатель на структуру structName
```

**В Си:**

```
newStructName *structVar;  
// newStructName должно быть объявлено с typedef
```

**или так, тоже для Си:**

```
struct name *structVar;
```

- Доступ к элементам структуры происходит с использованием символа «точка».
- Предположим, что у нас есть структурная переменная с именем **car** и у нее есть элемент с именем **speed**, к которому, мы сейчас получим доступ:

```
car.speed;
```

*Примечание: этот способ доступа к элементам структуры работает только в том случае, когда не используется указатель на структуру.*

## Доступ к элементам указателя на структуру

- Чтобы получить доступ к элементам структуры, через указатель на структуру, вместо оператора «точка», используйте оператор «стрелка» ->:

```
carPtr->speed;
```

# Структуры в C++

- Рассмотрим простой пример, который поможет познакомиться со структурами и покажет, как с ними работать.
- В этой программе мы создадим структуру, создадим объект структуры, заполним значениями элементы структуры (данные об объекте) и выведем эти значения на экран.

# Пример программы (1/2)

```
#include <iostream>
using namespace std;

struct building      //Создаем структуру!
{
    char owner[50]; //здесь будет храниться имя владельца
    char city[30];  //название города
    int amountRooms; //количество комнат
    float price;    //цена
};
```

# Пример программы (2/2)

```
int main()
{
    building apartment1;
    strcpy(apartment1.owner, "Ivan");
    strcpy(apartment1.city, "Nikolaev");
    apartment1.amountRooms = 2;
    apartment1.price = 2500;

    cout << "Владелец: " << apartment1.owner << endl;
    cout << "Квартира находится в городе: "
         << apartment1.city << endl;
    cout << "Количество комнат: "
         << apartment1.amountRooms << endl;
    cout << "Стоимость: " << apartment1.price
         << " $" << endl;
    return 0;
} // Весь код по ссылке http://code.re/7sy
```



# Демонстрація

<http://code.re/7sy>

# Дополнительные сведения о структурах в C++

Объект структуры можно объявить до функции `main()`. Это выглядело бы так:

```
struct building
{
    char owner[50]
    char city[30];
    int amountRooms;
    float price;
} apartment1; //объявление объекта типа building
```

Инициализировать структуру можно и таким способом:

```
building apartment1 = {"Ivan", "Nikolaev", 2, 2500};
```

... но так делают крайне редко



Структуры можно вкладывать в другие структуры. Например, так:

```
struct date // дата постройки
{
    char month[10]; // Месяц постройки дома
    int year; // Год
};

struct building
{
    char owner[50];
    char city[30];
    int amountRooms;
    float price;
    date built;
    //вкладываем одну структуру в определение второй
};
```

## После описания структуры можно использовать

```
void show(building object)
//функцию, которая принимает структуру, как параметр
{
    cout << "Владелец квартиры: "
          << object.owner << endl;
    cout << "Квартира находится в городе: "
          << object.city << endl;
    cout << "Количество комнат: "
          << object.amountRooms << endl;
    cout << "Стоимость: " << object.price
          << " $" << endl;
    cout << "Дата постройки: " << object.built.month
          << ' ' << object.built.year << endl;
}
```

Весь код примера можно взять здесь: <http://code.re/7sx>



НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ  
КОРАБЛЕБУДУВАННЯ  
ІМЕНІ АДМІРАЛА МАКАРОВА

# Демонстрація



Следует отметить, что функции могут так же возвращать структуры в результате своей работы. Например:

```
building Set()  
{  
    building object; // формирование объекта  
  
    //... код функции  
  
    return object;  
}
```



Спасибо!  
Вопросы?



# Алгоритмизация и программирование

Программирование на C/C++  
(ч.13 – структуры)

Беркунский Е.Ю., кафедра ИУСТ, НУК  
eugeny.berkunsky@gmail.com,  
<http://berkut.homelinux.com>