

Семинар №2

ФАКИ 2019

Бирюков В. А.

October 2, 2019

Тип данных, который принимает 2 значения: **False** и **True** (либо 0 и 1).

В языке C роль типа данных boolean часто играет тип int:

int	bool
0	False
любое другое значение	True

Возвращают тип bool

!	не
	или
&&	и

- ① `()`, `[]`
- ② `++`, `--`, `+`, `-` (унарные), `sizeof`
- ③ `*`, `/`, `%`
- ④ `+`, `-`
- ⑤ `>`, `<`, `<=`, `>=`
- ⑥ `==`, `!=`
- ⑦ `&`, `|`, `&&`, `||`
- ⑧ `=`, `+=`, и т.д.

Приоритет операторов C подробнее:

ru.cppreference.com/w/c/language/operator_precedence

Управляющие конструкции

Оператор ветвления: if else

Базовые управляющие конструкции

if, if else

```
if ( x < 5 )  
    printf("%d\n", x);  
  
if (x < 5) {  
    x++;  
    printf("%d\n", x);  
}
```

Базовые управляющие конструкции

if, if else

```
if ( x < 5 ) {  
    printf("Less than 5\n");  
}  
else if ( x == 5 ) {  
    printf("Equal to 5\n");  
}  
else {  
    printf("More than 5\n");  
}
```


Базовые управляющие конструкции

Тернарный оператор :?

```
expression ? expression : expression
```

```
z = (x>0) ? x : -x
```

```
min = (x<y) ? x : y
```

Циклы: while, do while, for

```
int i = 0;

while (i < 3)
{
    i++;
    printf("%d ", i);
}
```

Напечатает 1 2 3

```
int i = 0;  
  
do  
{  
    i++;  
    printf("%d ", i);  
} while (i < 3);
```

Напечатает 1 2 3

```
for (int i = 0; i < 3; ++i) {  
    printf("%d ", i);  
}
```

Напечатает 1 2 3

В условии циклов может стоять любой оператор:

```
int i = 0;

while (i++ < 3)
    printf("%d ", i);

for (; ; i++) {
    printf("%d\n", i);
}
```

Другие управляющие
конструкции: break, continue,
switch

```
for(int i = 0; i < 10; ++i){  
    if(i == 6){  
        break;  
    }  
    printf("%d ", i);  
}
```



```
for(int i = 0; i < 10; ++i){  
    if(i == 6){  
        continue;  
    }  
    printf("%d ", i);  
}
```

Управляющие конструкции

Оператор выбора switch

```
switch(i) {  
    case 1:  
        printf("It's one!\n");  
        break;  
    case 2:  
        printf("It's two!\n");  
        break;  
    default:  
        printf("It's something else!\n")  
}
```

Оператор goto

- Оператор goto передает управление на оператор, помеченный меткой
- Оператор goto в языках высокого уровня является объектом критики, поскольку чрезмерное его применение приводит к созданию нечитаемого кода
- Использование goto в практике программирования на языке C настоятельно не рекомендуется

Стиль кода

```
// One line comment  
/* Another one */  
/* Multi-  
    line  
    comment*/
```

- Отступы. В программе должна быть структура. Количество отступов соответствует уровню вложенности. Уровень вложенности увеличивается внутри `if`, `for`, `while`, `do-while`, `switch`
- Каждый отступ - это ЛИБО `TAB`, ЛИБО `n` пробелов (лучше всего `n = 4`). Мешать их нельзя.
- Скобка `{` должна быть на следующей строке, под началом ключевого слова `if/for/`.

- Скобка } должна быть СТРОГО под соответствующей {. После неё не должно быть ничего, за исключением комментариев.
- Каждый оператор (особенно, содержащий ключевое слово) должен быть с новой строки, после оператора и знака ; не должно быть ничего, кроме комментариев.
- Пробелы должны быть ПОСЛЕ , и ;(в цикле for) , а до них они НЕ нужны.