

## Лекция 4

# Операции и выражения

---

Преподаватель Палехова Ольга Александровна,  
кафедра О7 БГТУ «Военмех»

# Выражение

---

**Выражение** – синтаксически и семантически допустимая последовательность операндов и операций, позволяющая вычислить **значение**.

## Операнды:

- константы,
- переменные,
- выражения.



Выражение всегда имеет значение.

**Простейшее** выражение – один **операнд**.

Все **операнды** в выражении имеют **совместимые типы** – типы, допускающие неявные преобразования к одному общему типу.

# Операции языка Си

Приоритет	Знаки операций	Ассоциативность
1	(), [], ., ->	Слева направо
2	+, -, &, *, !, ~, ++, --, sizeof, (тип)	Справа налево
3	*, /, %	Слева направо
4	-, +	Слева направо
5	<<, >>	Слева направо
6	<=, <, >=, >	Слева направо
7	==, !=	Слева направо
8	&	Слева направо
9	^	Слева направо
10		Слева направо
11	&&	Слева направо
12		Слева направо
13	?:	Слева направо
14	=, *=, /=, %=, +=, -=, <<=, >>=, &=, ^=,  =	Справа налево
15	,	Слева направо

# Классификация операций

---

По синтаксису:

- **унарные** (применяются к одному операнду, форма записи – *префиксная* (знак операции перед операндом));
- **бинарные** (применяются к двум операндам, форма записи – *инфиксная* (знак операции между операндами));
- **тернарные** (применяются к трем операндам, форма записи – инфиксная).

По семантике:

- **арифметические** операции,
- **побитовые** операции,
- операции **присваивания**,
- операции **отношения**,
- **логические** операции,
- операции над указателями,
- другие.

# Арифметические операции

Мультипликативные:

- $*$  (умножение),
- $/$  (деление),
- $\%$  (деление по модулю (получение остатка от деления))

Аддитивные:

- $+$  (сложение),
- $-$  (вычитание).

```
int a = 3, b = 5;  
a / 2  → 1 тип int  
a / b  → 0 тип int  
a % b  → 3  
a / 2. → 1.5 тип double
```



Тип результата зависит от типа операндов.



Как поделить  $a$  на  $b$  и получить вещественный результат?

```
(double) a / b → 0.6 тип double
```

операция приведения типа

# Порядок выполнения операций

---

Выражения обычно вычисляются слева направо в соответствии с приоритетом операций, хотя порядок вычисления операндов стандартом не определен.

$$z = \frac{5a + c}{ab} (b - c)$$

1 2 4 3 6 5

$$z = (5 * a + c) / (a * b) * (b - c)$$

$$x = \frac{5c^2 - d(a + b)}{(c + d)(d - 2a)}$$

1 2 5 4 3 10 6 9 8 7

$$x = (5 * c * c - d * (a + b)) / ((c + d) * (d - 2 * a))$$

# Арифметические преобразования

---

Если типы операндов в выражении не совпадают, то операнды приводятся к общему типу, который также является и типом результата.

Основные правила преобразований:

- приведение типа осуществляется к типу операнда с бóльшим множеством значений;
- если один из операндов *float* или *double*, то оба преобразуются в *double*;
- если один из операндов принадлежит типу *int*, а другой - *unsigned int*, то оба преобразуются в *unsigned int*;
- если один из операндов принадлежит типу *signed char*, а другой - *unsigned char*, то оба преобразуются в *int*.

# Побитовые операции

Побитовые операции применяются только к целочисленным операндам.

- $\sim$  (побитовое отрицание, поразрядная инверсия);
- $\gg, \ll$  (сдвиг);

$5 \ll 2 \rightarrow 20$

00000101

00010100

Сдвиг влево на X  
эквивалентен  
умножению на  $2^X$

$5 \gg 2 \rightarrow 1$

00000101

00000001

Сдвиг вправо на X  
эквивалентен  
делению на  $2^X$

- $\&$  (побитовое и);
- $\wedge$  (побитовое исключающее или);
- $|$  (побитовое или).

$5 \& 6 \rightarrow 4$

00000101

00000110

00000100

$5 \& 2 \rightarrow 0$

$5 \wedge 6 \rightarrow 3$

00000101

00000110

00000011

$5 \wedge 2 \rightarrow 7$

$5 | 6 \rightarrow 7$

00000101

00000110

00000111

$5 | 2 \rightarrow 7$



# Операции присваивания

---



Операции присваивания изменяют значение операнда.

К группе присваивания относятся:

- простое присваивание

`= ;`

- расширенное присваивание

`*=, /=, %=, +=, -=, <<=, >>=, &=, ^=, |= ;`

- операции **инкремента** и **декремента**

`++, --`

Операндом для операции инкремента/декремента и левым операндом для операций присваивания может являться только **lvalue (леводопустимое выражение)** – выражение, обозначающее некоторую область памяти, значение в которой можно изменить.

# Расширенные операции присваивания

---

простое присваивание	расширенное присваивание
$a = a + b$	$a += b$
$a = a - b$	$a -= b$
$a = a * b$	$a *= b$
$a = a / b$	$a /= b$
$a = a \% b$	$a \% = b$
$a = a << b$	$a << = b$
$a = a \& b$	$a \& = b$
$a = a * (b + c)$	$a *= b + c$

# Преобразования типов при присваивании

---

Значение правой части присвоения приводится к типу левой части, который и является типом результата.

Возможна потеря данных:

- значения типа *int* или *long int* преобразуются в *short int* или в *char* путем отбрасывания старших разрядов;
- значения типа *float* или *double* преобразуются в *int* отбрасыванием дробной части; если полученное при этом значение нельзя представить в заданном целочисленном типе, то результат не определен;
- если значение преобразуется из *int* во *float* и оно находится в допустимом диапазоне, но представляется в новом типе неточно, то результатом будет одно из двух значений нового типа, ближайших к исходному.

# Инкремент и декремент

простое присваивание	префиксная форма записи	постфиксная форма записи
<b>a = a + 1</b>	<b>++a</b>	<b>a++</b>
<b>a = a - 1</b>	<b>--a</b>	<b>a--</b>



Значение выражения инкрементации или декрементации зависит от формы его записи.

Если выражение записано в **префиксной** форме, то его значением является значение операнда **после** его изменения, а если в выражение записано в **постфиксной** форме, то его значением является значение операнда **до** его изменения.

Значение операнда в любом случае изменяется на единицу.

```
int a = 3, b = 5;  
a += ++b;
```

→ b = 6, a = 9

```
int a = 3, b = 5;  
a += b++;
```

→ b = 6, a = 8

# Операции отношения

К операциям отношения относятся:

- отношения  
 $<, <=, >, >=$  ;
- сравнения на равенство  
 $==, !=$

```
int a = 3, b = 5;  
a < b   → 1 тип int  
a >= b  → 0 тип int  
a == b  → 0  
a != b  → 1
```

!

Результатом выражения отношения будет число:  
1 – выражение истинно, 0 – выражение ложно.

$x \in (a, b)$

~~$a < x < b$~~

$a < x == x < b$

синтаксически верное  
выражение,  
семантическая ошибка

!

Вещественные числа сравнивать на равенство некорректно. Модуль их разности сравнивают с допустимой для конкретной задачи погрешностью.