

## Язык программирования С

B.Г.Тетерин – Microsoft Solution Developer (Visual C++) teterin@specialist.ru



модуль 3

ОПЕРАТОРЫ (ОПЕРАЦИИ)

## Модуль 3. Операторы

- Операторы присваивания
- Арифметические и комбинированные операторы
- Отношения и логические операторы
- Битовые, адресные и прочие операторы
- Приоритеты и ассоциативность операторов
- Трассировка программы

#### Выражения. Операнды. Операторы.

- Выражение формула для вычисления значения.
  - Синтаксически выражение строится из операндов и операторов (знаков операций, англ. operators).
- Операнд аргумент операции, т.е. значение, участвующее в вычислении.
  - В качестве операндов в выражениях используются идентификаторы, константы и другие выражения (возможно, заключенные в скобки)
- Оператор (операция) вычисление, выполняемое над значениями операндов.
  - В зависимости от числа операндов операторы подразделяются на:
    - одноместные (унарные),
    - двуместные (бинарные),
    - многоместные операторы (в С имеется один тернарный оператор).
  - В зависимости от положения операндов относительно символа оператора:
    - префиксные, например, -*x*,
    - инфиксные, например, **a b**,
    - постфиксные, например, x--.
- Вычисление выражений выполняется в соответствии с приоритетами и ассоциативностью операторов (операций).

#### Оператор присваивания.

- Синтаксис: адресное выражение (Lvalue) = выражение (Rvalue)
- Порядок вычисления:
  - вычисляются выражения
  - сравниваются типы левой и правой части
  - при совпадении типов выполняется присваивание
  - при несовпадении типов выполняется приведение типа правой части к типу левой части, и затем присваивание
- Результатом оператора является присвоенное значение
  - Результат может участвовать в более сложном выражении:

$$x = y = z = 1.5$$

Приоритет	оритет Лексемы Тип операции		Ассоциативность	
15	=	Простое присваивание	Справа налево	

#### Арифметические операторы

• Арифметические вычисления:

```
      a + b
      сложение

      a - b
      вычитание

      a * b
      умножение

      a / b
      деление

      a % b
      остаток от деления (для целых типов)

      -a
      изменение знака

      +a
      унарный плюс
```

- Перед выполнением бинарных операций операнды приводятся к одному типу и результат имеет тот же тип.
- Явное приведение типа:

(тип)а (double)а

Приоритет Лексемы		Тип операции	Ассоциативность	
2	+ -	Унарные	Справа налево	
3	(тип)	Приведение типа	Справа налево	
4	* / %	Мультипликативные	Слева направо	
5	+ -	Аддитивные	Слева направо	

#### Комбинированные операторы

• Инкремент и декремент:

$$a + + a$$
  $a + + a = a + 1$   
 $a - - a$   $a - - a$   $a = a - 1$ 

Комбинирование с присваиванием:

```
      a += b означает
      a = a + b

      a -= b означает
      a = a - b

      a *= b означает
      a = a * b

      a /= b означает
      a = a / b

      a %= b означает
      a = a % b
```

Приоритет	Лексемы	Тип операции	Ассоциативность	
1	постфиксные ++ и —	Доступ	Слева направо	
2	префиксные ++ и —	Унарные	Справа налево	
15	= *= /= %= += -=	Простое и комбинированное присваивание	Справа налево	

#### Операторы отношения

• Операторы сравнения:

**a** > **b** больше

a >= b больше или равно

**a** < **b** меньше

a <= b меньше или равно

a == b равно

**a** != **b** не равно

- Операторы сравнения возвращают одно из значений:
  - 1 (true)
  - 0 (false)

Приоритет	Лексемы	Тип операции	Ассоциативность
7	< > <= >=	Отношения	Слева направо
8	== !=	Равенство	Слева направо

#### Логические операторы

• Логические операторы:

! a отрицание

a && b конъюнкция (логическое «И»)

a || b дизъюнкция (логическое «ИЛИ»)

а	b	&&	11
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

• Условный оператор:

a ? b : c

Пример: a > b ? a:

Приоритет	Лексемы	Тип операции	Ассоциативность	
2	!	Унарные	Справа налево	
12	&&	Логическое «И»	Слева направо	
13	П	Логическое «ИЛИ»	Слева направо	
14	? :	Условное выражение	Справа налево	

#### Битовые операторы

• Операнды должны иметь целый тип и рассматриваются как цепочки битов.

~	побитовое отрицание	~a			
&	побитовая конъюнкция	a & b	a	b	^
	побитовая дизъюнкция	a   b	0	0	0
٨	побитовое XOR	a ^ b	0	1	1
<<	сдвиг влево	a << k, (k>=0)	J	1 1	1
>>	сдвиг вправо	a >> k, (k>=0)	× 1	0	1
<ul><li>Ko</li></ul>	мбинации с присваиванием	а &= b и т.п.	1	1	0

Приоритет	Лексемы	Тип операции	Ассоциативность	
2	~	Унарные	Справа налево	
6	<< >>	Битовый сдвиг	Слева направо	
9	&	Побитовое «И»	Слева направо	
10	٨	Побитовое исключающее «ИЛИ» Слева напр		
11	1	Побитовое «ИЛИ»	Слева направо	
15	<<= >>= &= ^=  =	Простое и комбинированное Справа на присваивание		

#### Операторы доступа и адресные операторы

- Операторы доступа
  - array [ index ] доступ к элементу массива
  - function ( arg\_list )доступ к значению функции
  - struct\_var . field\_name доступ к полю структуры
- Адресные операторы
  - & var
     вычисление адреса объекта
  - \* pointer
     разадресация указателя (доступ по указателю, доступ по адресу)
  - pointer\_to\_struct\_var->field\_name доступ к полю структуры по адресу

Приоритет	Лексемы	Тип операции	Ассоциативность	
1	[]()>	Доступ	Слева направо	
2	& *	Унарные	Справа налево	

#### Прочие операторы

- Размер объекта
  - sizeof(type)
  - sizeof expression
- Последовательное вычисление
  - expression1 , expression2

- размер типа в байтах
- размер типа результата выражения в байтах

- операция "запятая"

Приоритет	Лексемы	Тип операции	Ассоциативность
2	sizeof	Унарные	Справа налево
16	1	Последовательное вычисление	Слева направо

#### Приоритеты и ассоциативность операторов

Приоритет	Лексемы	Тип операции	Ассоциативность
1	[]()> постфиксные ++ и —	Доступ	Слева направо
2	префиксные ++ и — sizeof & * + - ~!	Унарные	Справа налево
3	(тип)	Приведение типа	Справа налево
4	* / %	Мультипликативные	Слева направо
5	+ -	Аддитивные	Слева направо
6	<< >>	Битовый сдвиг	Слева направо
7	< > <= >=	Отношения	Слева направо
8	== !=	Равенство	Слева направо
9	&	Побитовое «И»	Слева направо
10	٨	Побитовое исключающее «ИЛИ»	Слева направо
11	1	Побитовое «ИЛИ»	Слева направо
12	&&	Логическое «И»	Слева направо
13	11	Логическое «ИЛИ»	Слева направо
14	? :	Условное выражение	Справа налево
15	= *= /= %= += -= <<= >>= &= ^=  =	Простое и комбинированное присваивание	Справа налево
16	1	Последовательное вычисление	Слева направо

### Итоги

- В этом модуле Вы изучили:
  - Синтаксис 47 операторов (операций) языка С
  - Их приоритеты и ассоциативность в процессе грамматического разбора выражений
  - Семантические особенности вычисления выражений в языке С

## Вопросы?

■ В.Г.Тетерин – Microsoft Solution Developer (Visual C++)

• teterin@specialist.ru

www.specialist.ru



# ПРИЛОЖЕНИЕ ЗАДАЧИ

## Задачи

1. Ввести длину отрезка с указанием шкалы измерения (с – сантиметры, і – дюймы), напечатать результат в обеих шкалах: например,

```
при вводе «2i» напечатать «2"= 5.08 cm» при вводе «5.08c» напечатать «5.08 cm = 2"»
```

- 2. Ввести два числа, напечатать знак сравнения между ними
- 3. Ввести три числа, напечатать максимальное из них
- 4. Образуют ли цифры четырехзначного числа арифметическую прогрессию
- Введите экзаменационные оценки студента по 3 предметам и напечатайте его статус: «отличник», «хорошист», «троечник», «двоечник»

## Задачи

- 6. Ввести два значения времени в формате чч:мм:сс, проверить корректность ввода и напечатать знак сравнения между ними
- 7. Ввести номер года, напечатать количество дней в этом году (справка: год является високосным, если он делится на 4 и на 400 в том случае, если оканчивается на два нуля, например, 2000)
- 8. Ввести две даты в формате дд.мм.ггг, напечатать знак сравнения между ними
- 9. Ввести дату в формате дд.мм.гггг, проверить корректность ввода и напечатать следующую дату

## Задачи

- 10. К введенному числу (от 0 до 99) приписать слово «копеек» в правильном падеже
- 11. К введенному числу (от 0 до 1000) приписать слово «рублей» в правильном падеже
- 12. К введенному числу (от 0 до 10...0.99) приписать слова «рублей» и «копеек» в правильном падеже
- 13. Три числа являются длинами трех отрезков. Могут ли эти отрезки быть сторонами треугольника и, если да, то какой это будет треугольник (остро- прямо- или тупоугольный, равнобедренный, равносторонний и т.д.).
- 14. Даны две тройки чисел, являющихся длинами отрезков. Образует ли каждая тройка стороны треугольника и будут ли эти треугольники подобными