

Operators

Intro

Problem

Как выполнять какие-то действия над данными?

Solutions

Операторы

Operators

Operators

- **Операторы** - это специальные символы
- которые сообщают компилятору о том, что вы хотите выполнить операцию с некоторым набором **operands/операнд:**
 - переменных
 - констант
 - литералов

Operators

Операторы бывают:

- **унарные** - выполняются над одним операндом
- **бинарные** - выполняются над двумя операндами
- **тернарные** - выполняются над тремя операндами

Математические операторы

Математические операторы

Обозначение	Пример	Название	Описание
+	a + b	Оператор сложения	Складывает два числа.
-	c - d	Оператор вычитания	Вычитает второе число из первого.
*	a * t	Оператор умножения	Умножает два числа.
/	a / b	Оператор деления	Делит первое число на второе.
%	c % d	Остаток от деления	Вычисляет остаток от деления первого числа на второе.

Математические операторы

Обозначение	Пример	Название	Описание
-	-a	Одинарный минус	Меняет знак переменной на обратный.
+	+a	Одинарный плюс	Ничего не меняет.

Examples

Выражение	Ответ	Описание
$19 / 10$	1	Если делить нацело 19 на 10, получим 1 и 9 в остатке.
$19 \% 10$	9	Если делить нацело 19 на 10, получим 1 и 9 в остатке.
$2 / 5$	0	Если делить нацело 2 на 5, получим 0 и 2 в остатке.
$16 \% 2$	0	Если делить нацело 16 на 2, получим 8 и 0 в остатке.

Операторы сравнения

Операторы сравнения

Обозначение	Пример	Название	Описание
<	$a < b$	Меньше	Проверяет, что a меньше b .
<=	$c \leq d$	Меньше или равно	Проверяет, что c меньше или равно d .
>	$a > b$	Больше	Проверяет, что a больше b .
>=	$c \geq d$	Больше или равно	Проверяет, что c больше или равно d .
==	$i == j$	Равно	Проверяет, что i равно j .
!=	$a != 0$	Неравно	Проверяет, что a неравно 0 .

Логические операторы

Логические операторы

Обозначение	Пример	Название	Описание
&&	a && b	AND – «и»	Результат выражения true , только когда и a и b равны true .
	c d	OR – «или»	Результат выражения true , если a или b равно true . Оба или хотя бы один.
!	!a	NOT – «не»	Результат выражения true , только если a равно false .

Логический оператор **&&**

A	B	A && B
true	true	true
true	false	false
false	true	false
false	false	false

Логический оператор ||

A	B	A B
true	true	true
true	false	true
false	true	true
false	false	false

Логический оператор !

A	!A
true	false
false	true

Побитовые операторы

Побитовые операторы

Обозначение	Пример	Название	Описание
&	a & b	AND – «и»	Побитовое «И»
	c d	OR – «или»	Побитовое «ИЛИ»
~	~a	NOT – «не»	Побитовое «НЕ»
^	a ^ b	XOR – «исключающее или»	Побитовое «ИСКЛЮЧАЮЩЕЕ ИЛИ»

Побитовый оператор ^

A	B	$A \wedge B$
true	true	false
true	false	true
false	true	true
false	false	false

Examples

Пример	Числа в битовом виде	Ответ в битовом виде	Ответ
5 & 3	00000101 & 00000011	00000001	1
7 & 2	00000111 & 00000010	00000010	2
5 9	00000101 00001001	00001101	13
5 ^ 9	00000101 ^ 00001001	00001100	12
~9	~00001001	11110110	246

Операторы сдвига

Операторы сдвига

Обозначение	Пример	Название	Описание (что делает)
>>	a >> b	сдвиг вправо	Сдвигает биты числа a , на b разрядов вправо.
<<	c << d	сдвиг влево	Сдвигает биты числа c , на d разрядов влево.
>>>	a >>> 2	сдвиг вправо с заполнением нулем	Сдвигает биты числа a , на 2 разряда вправо.

Examples

Пример	Числа в битовом виде	Ответ в битовом виде	Ответ
10 >> 1	00001010 >> 1	00000101	5
10 >> 2	00001010 >> 2	00000010	2
10 << 1	00001010 << 1	00010100	20
10 << 2	00001010 << 2	00101000	40

Examples

Выражение	Результат	Описание
10001010 >> 1	11000101	Отрицательное число остается отрицательным. Новые разряды заполняются 1, для отрицательных чисел.
10001010 >> 2	11100010	Аналогично
10001010 >> 3	11110001	Аналогично
10001010 >>> 1	01000101	Отрицательное число перестает быть отрицательным. Новые разряды заполняются 0, для отрицательных чисел.
10001010 >>> 2	00100010	Аналогично

10001010 >>> 3

00010001

Аналогично

Операторы присваивания

Операторы присваивания

Оператор	Аналогичен
<code>a += b</code>	<code>a = a + b</code>
<code>a -= b</code>	<code>a = a - b</code>
<code>a *= b</code>	<code>a = a * b</code>
<code>a %= b</code>	<code>a = a % b</code>
<code>a = b</code>	<code>a = a b</code>
<code>a &= b</code>	<code>a = a & b</code>

Операторы инкремента и декремента

Операторы инкремента и декремента

Оператор	Пример	Описание
++ (постфиксный)	a++	Увеличивает число на 1.
++ (префиксный)	++b	Увеличивает число на 1.
-- (постфиксный)	d--	Уменьшает число/переменную на 1.
-- (префиксный)	--i	Уменьшает число/переменную на 1.

Инкремент

```
int a = 8;  
int b;
```

```
b = a++; // постфиксный инкремент
```

```
b = a; // аналог постфиксного инкремента  
a = a + 1;
```

```
b = ++a; // префиксный инкремент
```

```
a = a + 1; // аналог префиксного инкремента  
b = a;
```


Декремент

```
int a = 8;  
int b;
```

```
b = a--; // постфиксный декремент
```

```
b = a; // аналог постфиксного декремента  
a = a - 1;
```

```
b = --a; // префиксный декремент
```

```
a = a - 1; // аналог префиксного декремента  
b = a;
```

Остальные операторы

Остальные операторы

Запись	Пример	Описание
<code>()</code>	<code>(a + b) * c</code>	Скобки повышают приоритет выполнения. Сначала выполняется то, что в скобках.
<code>[]</code>	<code>c [i] = c [i + 1]</code>	Получение элемента массива по индексу.
<code>.</code>	<code>int n = a.length</code>	«оператор точка» – получение переменных и методов у объекта.

Приоритет операций

Приоритет операций

```
( ) [ ] .  
++ -- + - ~ !  
* / %  
+ -  
<< >> >>>  
< > <= >= instanceof  
== !=  
&  
^  
|  
&&  
||  
? : (тернарный оператор)  
= += -= *= /= %= &= ^= |= <<= >>= >>>=
```