

Программирование для начинающих

Пятница, 26 июля 2019 года

Обработка: побитовые операторы

Операторы поддержки обработки, которые выполняют побитовые и битовые операции над целочисленными типами.

Поддержка обработки ниже операторов

оператор	Описание
&	Немного мудрый И
	Немного мудрый ИЛИ
^	Немного мудрый Эксклюзив ИЛИ
~	Битовый паттерн
<<	Выполнить сдвиг влево
>>	Выполнить сдвиг вправо
>>>	Нулевое заполнение вправо сдвиг

Немного И И (&), ИЛИ (|)

Bit1	Bit2	&		^
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Operators.pde

```
байт a = 8 ;  
байт b = 9 ;  
  
println ( "a & b is" + ( a & b ));  
println ( «a | b is» + ( a | b ));  
println ( "a ^ b is" + ( a ^ b ));
```

Вывод

a & 6 8

a | 6 9

a ^ 6 = 1

объяснение

a = 0000 1000

b = 0000 1001
a | b = 0000 10001

a = 0000 1000
b = 0000 1001
a ^ b = 0000 0001

Примечание:

Побитовые операторы применимы только к целочисленным значениям, таким как байты, короткие и т. Д., Они не применимы к плавающим, двойным, логическим значениям.

~: Инвертировать битовый шаблон

Инверсионный оператор изменяет значение бита на ноль, если оно равно единице, и заменяет значение бита на единицу, если оно равно нулю.

пример

байт a = 7;

в двоичном формате 0 0 0 0 0 1 1 1
~ (a) в двоичном формате: 1 1 1 1 1 0 0 0
~ (a) представляет -8 в двоичном формате.

Operators.pde

```
byte byteVar = 7 ;  
  
println ( «Значение переменной байта равно» + byteVar );  
println ( «Значение байтовой переменной после инверсии» + ~ ( byteVar ));
```

Вывод

Значение байтовой переменной равно 7

Значение байтовой переменной после инверсии равно -8

Бит мудрый сдвиг влево (<<)

Оператор сдвига влево просто сдвигает все биты влево на один бит, т. Е. Просто удваивает значение.

Синтаксис:

переменная << n

сдвиньте переменную влево на «n» количество раз.

Переменная << n равна переменной = переменная * 2 степень n

```
int var = 2;  
var << 1 = 2 * 2 мощность 1 = 4  
var << 2 = 2 * 2 мощность 2 = 8  
var << 9 = 2 * 2 мощность 9 = 1024
```

```
println ( «Значение переменной байта равно» + byteVar );
println ( «После сдвига на 1 бит влево» + ( byteVar << 1 ));
println ( «После сдвига на 2 бита влево» + ( byteVar << 2 ));
println ( «После сдвига на 3 бита влево» + ( byteVar << 3 ));
println ( «После сдвига на 4 бита влево» + ( byteVar << 4 ));
println ( «После сдвига на 5 бит влево» + ( byteVar << 5 ));
println ( «После сдвига на 6 бит влево» + ( byteVar << 6 ));
println ( «Значение переменной байта равно» + byteVar );
```

Вывод

Значение байтовой переменной равно 2

После сдвига 1 бит влево 4

После сдвига осталось 2 бита 8

После сдвига осталось 3 бита 16

После сдвига осталось 4 бита 32

После сдвига осталось 5 битов 64

После сдвига осталось 6 битов 128

Значение байтовой переменной равно 2

Бит мудрый правый сдвиг (>>)

Оператор сдвига вправо просто сдвигает все биты на один бит вправо, т.е. он просто наполовину (делит на 2) значение

Синтаксис:

переменная >> n

сдвиньте переменную вправо на «n» количество раз.

Переменная >> n равна переменной = переменная / 2 степени n

int var = 128;

var >> 1 = 128 / (2 степени 1) = 64

var >> 2 = 128 / (2 степень 2) = 32

Operators.pde

```
int intVar = 128 ;

println ( «Значение переменной int равно» + intVar );
println ( «После сдвига на 1 бит вправо» + ( intVar >> 1 ));
println ( «После сдвига на 2 бита вправо» + ( intVar >> 2 ));
println ( «После сдвига на 3 бита вправо» + ( intVar >> 3 ));
println ( «После сдвига на 4 бита вправо» + ( intVar >> 4 ));
println ( «После сдвига на 5 бит вправо» + ( intVar >> 5 ));
println ( «После сдвига на 6 бит вправо» + ( intVar >> 6 ));
println ( «После сдвига на 6 бит вправо» + ( intVar >> 7 ));
println ( «Значение переменной int равно» + intVar );
```

Вывод

Значение переменной int равно 128

После сдвига 1 бит вправо 64

После сдвига 2 бит вправо 32

После смещения 6 бит вправо 1
Значение переменной int равно 128

Бит мудрый без знака вправо Shift (>>>)

сдвигает все биты на один бит вправо и делает самый левый бит на ноль.

Operators.pde

```
int intVar = - 128 ;  
  
println ( «Значение переменной int равно» + intVar );  
println ( «После сдвига на 1 бит вправо» + ( intVar >>> 1 ));
```

Вывод

Значение переменной int равно -128
После сдвига 1 бит вправо 2147483584

объяснение

-128 в форме дополнения 2

11111111 11111111 11111111 10000000

>>> сдвигает один бит вправо и заменяет самый левый бит на ноль

так (-128 >>> 1), как показано ниже

01111111 11111111 11111111 11000000

[Предыдущая](#)

[Следующая](#)

[Главная](#)

Кришна в 10:27

доля

Без комментариев:

[Оставить комментарий](#)



[Домой](#)



[Просмотр веб-версии](#)

При поддержке [Blogger](#) .
