



# Java Starter

Логічні та побітові операції

# Java Starter

Після уроку обов'язково



Повторіть цей урок у форматі відео на [ITVDN.com](http://itvdn.com)

Доступ можна отримати через керівництво вашого навчального центру



Перевірте, як Ви засвоїли цей матеріал на [TestProvider.com](http://testprovider.com)

## Логічні та побітові операції

# AND - &&

## Кон'юнкція

**Кон'юнкція** (від лат. *conjunction* – союз, зв'язок) – логічна операція, яка за своїм застосуванням максимально наближена до союзу «і».

операнд1 && операнд2 = результат

Таблиця істинності для операції кон'юнкції двох логічних виразів:

true && true = true      true && false = false

false && false = false      false && true = false

# AND - &

## Побітове «|»

Побітове I – це бінарна операція, дія якої еквівалентна застосуванню логічного I до кожної пари бітів, які стоять на одинакових позиціях у двійкових представленнях операндів.

Таблиця істинності для операції побітового «I» (кон'юнкції) значень

$$1 \text{ } \& \text{ } 1 = 1$$

$$0 \text{ } \& \text{ } 0 = 0$$

$$1 \text{ } \& \text{ } 0 = 0$$

$$0 \text{ } \& \text{ } 1 = 0$$

$$\text{result} = 255 \text{ } \& \text{ } 1 = 1$$

$$\begin{array}{r} & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \& 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \quad \& \quad \begin{array}{r} 255 \\ 1 \\ \hline 1 \end{array}$$

# OR - ||

## Диз'юнкція

**Диз'юнкція** (*лат. disjunctio – роз'єднання*) – логічна операція, яка за своїм застосуванням максимально наближена до союзу «або» в сенсі «або те, або це, або обидва відразу».

операнд1 || операнд2 = результат

Таблиця істинності для операції диз'юнкції двох логічних виразів:

true		true	=	true	true		false	=	true
------	--	------	---	------	------	--	-------	---	------

false		false	=	false	false		true	=	true
-------	--	-------	---	-------	-------	--	------	---	------

## Побітове «АБО»

Побітове «АБО» – це бінарна операція, дія якої еквівалентна застосуванню логічного АБО доожної пари бітів, які стоять на однакових позиціях у двійкових представленнях операндів.

Таблиця істинності для операції побітового «АБО» (диз'юнкції) значень

$$1 \mid 1 = 1$$

$$0 \mid 0 = 0$$

$$1 \mid 0 = 1$$

$$0 \mid 1 = 1$$

$$\text{result} = 2 \mid 1 = 3$$

$$\begin{array}{r} 0000\ 0010 \\ 0000\ 0001 \\ \hline 0000\ 0011 \end{array} \quad \begin{array}{r} 2 \\ 1 \\ \hline 3 \end{array}$$

# XOR - ^

## Виключене АБО

**Виключене АБО** (логічне додавання, строга диз'юнкція) – булева функція та логічна операція. Результат виконання операції є дійсним лише за умови, якщо один операнд є дійсним, а інший хибним.

операнд1 ^ операнд2 = результат

Таблиця істинності для операції диз'юнкції двох логічних виразів:

**true ^ true = false      true ^ false = true**

**false ^ false = false      false ^ true = true**

## Побітове «Виключне АБО»

Побітове виключне АБО – це бінарна операція, дія якої еквівалентна застосуванню логічного виключеного АБО до кожної пари бітів, які стоять на однакових позиціях у двійкових представленнях операндів.

Таблиця істинності для операції побітового «Виключного АБО» значень

$$1 \wedge 1 = 0$$

$$0 \wedge 0 = 0$$

$$1 \wedge 0 = 1$$

$$0 \wedge 1 = 1$$

$$\text{result} = 3 \wedge 1 = 2$$

$$\begin{array}{r} \wedge \quad 00000011 \\ \wedge \quad 00000001 \\ \hline 00000010 \end{array} \qquad \qquad \qquad \begin{array}{r} \wedge \quad 3 \\ \wedge \quad 1 \\ \hline 2 \end{array}$$

# NOT - !

## Заперечення

**Заперечення** в логіці – унарна операція над судженнями, результатом якої є судження (у певному сенсі) «протилежне» вихідному.

$! \text{ операнд} = \text{результат}$

Таблиця істинності для операції заперечення логічного виразу:

$!\text{false} = \text{true}$

$!\text{true} = \text{false}$

*Запереченнем істини є брехня, а запереченнем брехні – істина.*

## Побітове заперечення

Побітове заперечення (НЕ) – це унарна операція, дія якої еквівалентна застосуванню логічного заперечення до кожного біта двійкового уявлення операнда.

Таблиця істинності для операції побітового «Заперечення НЕ» значень

$$\sim 0 = 1$$

$$\sim 1 = 0$$

$$\sim 1 = -2$$

$$\sim \begin{array}{r} 00000001 \\ \hline 11111110 \end{array} \sim \begin{array}{r} 1 \\ -2 \end{array}$$

# Двійкове число

## Зміна знаку

Щоб змінити знак числа, необхідно виконати його заперечення та отриманий результат збільшити на 1.

Формула зміни знака числа

$$\sim (+N) + 1 = -N$$

$$\sim (-N) + 1 = +N$$

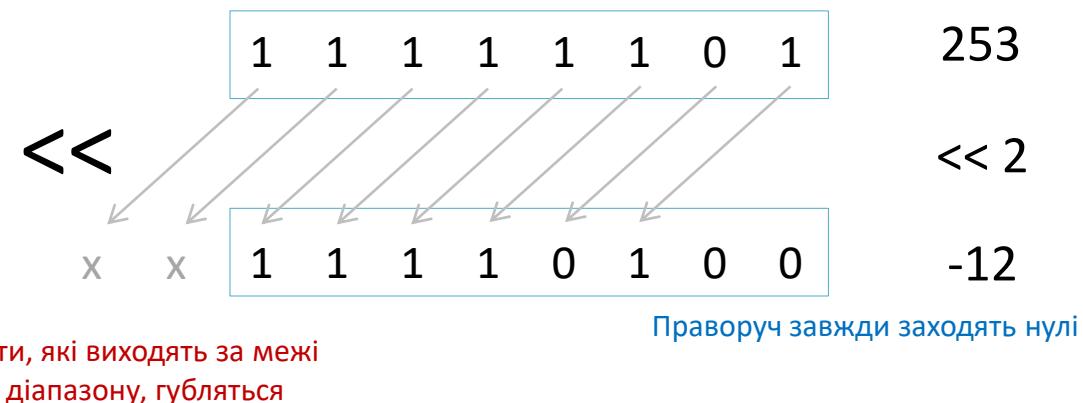
$$\begin{array}{r} \sim 00000001 \\ \sim 11111110 \\ + 00000001 \\ \hline 11111111 \end{array} \quad \begin{array}{r} 1 \\ -2 \\ + 1 \\ \hline -1 \end{array}$$

# Logical shifts - <<, >>

## Логічні зсуви числа

Під час логічного зрушення значення останнього біта за напрямком зсуву втрачається (копіюючись у біт перенесення), а перший набуває нульового значення.

$$253 \ll 2 = -12$$

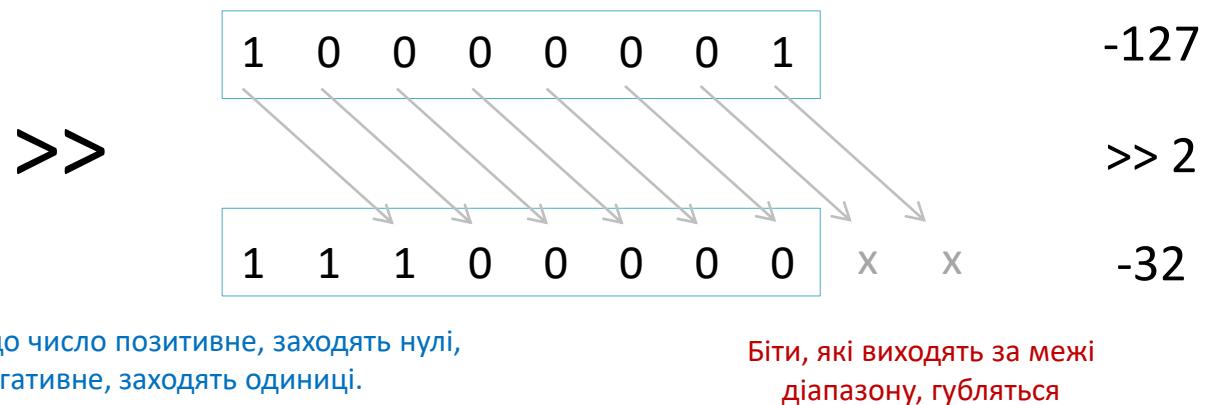


# Logical shifts - <<, >>

## Логічні зсуви числа

Під час логічного зрушення значення останнього біта за напрямком зсуву втрачається (копіюючись у біт перенесення), а перший набуває нульового значення.

$$-127 \gg 2 = -32$$



## Q&A

Інформаційний відеосервіс для розробників програмного забезпечення

