**אופרטורים וביטויים לוגיים**

אז מה למדנו עד כה?

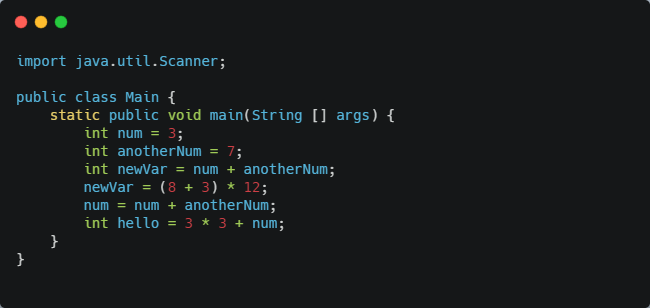
* צורת העבודה עם סביבת העבודה intellij.
* יצירת תכנית בסיסית בשפת JAVA.
* הגדרתם משתנים בתוכנית.

כעת נלמד על אופרטורים וביטויים לוגיים.

אם המשתנים מאפשרים לנו לשמור ערכים בזיכרון המחשב, **אופרטורים** מאפשרים לנו לשנות את הערכים האלו, לשחק איתם, ולעדכן אותם בהתאם לצורך.

**דוגמה לשימוש באופרטורים:**

ננתח את הריצה של הפרויקט, מלמעלה למטה, שורה אחר שורה.



* הגדרנו משתנה מסוג int, שמו הוא num **והשמנו** בתוכו את הערך **3**.
* הגדרנו משתנה נוסף מסוג int, שמו הוא anotherNum **והשמנו** בתוכו את הערך **7**.
* הגדרנו משתנה נוסף מסוג int, שמו הוא newVar והשמנו בתוכו את הערך של המשתנה num **ועוד** הערך של המשתנה anotherNum.  
  כלומר newVar יכיל את הערך 7+3 = **10**
* לאחר ביצענו שוב השמה לערך newVar, דרסנו את הערך הישן והשמנו ערך חדש:

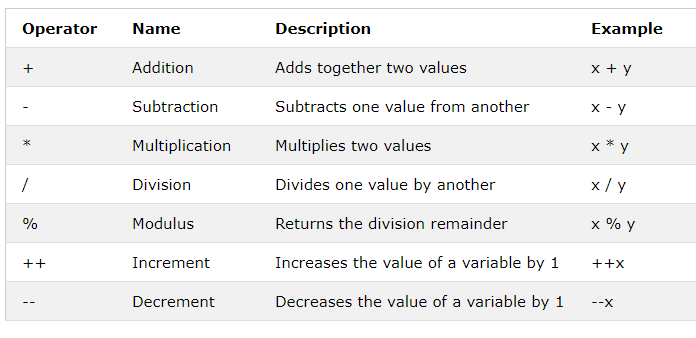
(8+3) \* 12 = **132**

* דרסנו את num בערך חדש num+anotherNum = 7+3 = **10**
* יצרנו משתנה חדש בשם hello, וביצענו לו השמה 3\*3 + num = **19**

**בסוף ריצת התוכנית**, ערכי המשתנים הסופיים יהיו:

|  |  |
| --- | --- |
| 10 | num |
| 7 | anotherNum |
| 132 | newVar |
| 19 | hello |

ישנם המון סוגים של אופרטורים בJava, המאפשרות לבצע פעולות מתמטיות שונות:

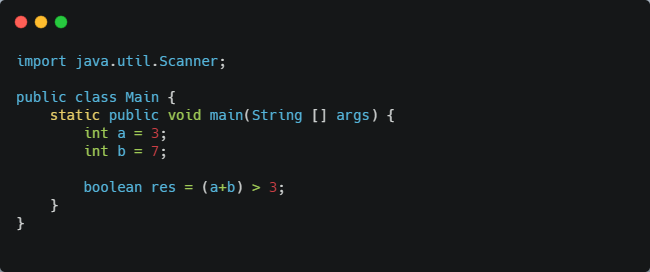


ניתן לקרוא עוד על אופרטורים בJava בלינק הבא:

<https://www.w3schools.com/java/java_operators.asp>

**ביטויים לוגיים:**

לעיתים נרצה להשוות בין משתנים מסוימים בצורות מסוימות, על כן נצטרך להשתמש בביטויים מסוימים שיקלו עלינו לביצוע ההשוואות.

[](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-csrc&ds=true&dsyoff=20px&dsblur=46px&wc=true&wa=false&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=1x&wm=false&code=import%2520java.util.Scanner%253B%250A%250Apublic%2520class%2520Main%2520%257B%250A%2520%2520%2520%2520static%2520public%2520void%2520main(String%2520%255B%255D%2520args)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520a%2520%253D%25203%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520b%2520%253D%25207%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520boolean%2520res%2520%253D%2520(a%252Bb)%2520%253E%25203%253B%250A%2520%2520%2520%2520%257D%250A%257D%250A)

נשים לב שהגדרנו שני משתנים, a,b עם הערכים 3 ו-7 בהתאמה.

לאחר מכן לתוך המשתנה res ביצענו השמה של תוצאת החישוב הבא:

boolean res = (a+b) > 3;

בפקודה זו, נבדוק האם הסכום של ערכי המשתנים a ו-b **גדולים** מהערך 3. השתמשנו באופרטור ‘>’ השקול למונח 'גדול מ'.

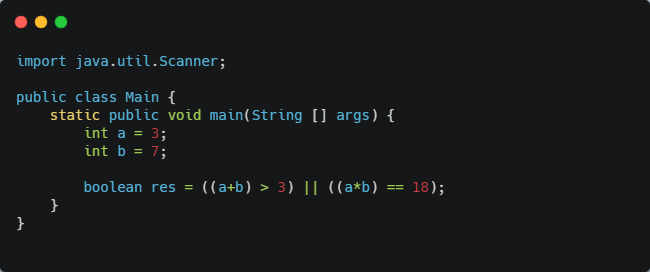
היות וערכם של a,b הם 3,7 אזי תוצאת ביטוי לוגי זה יהיה **אמת.**אכן, סכומם של a ושל b גדול מ 3.

לאחר ריצת השורה הנוכחית, המשתנה res יכיל את הערך הבוליאני **true**.

מה נעשה במידה ונרצה לבדוק כמה תנאים ביחד?

בדיוק בשביל זה יש לנו בשפת JAVA **אופרטורים לוגיים.**

אופרטורים לוגיים מאפשרים לנו לבצע פעולות לוגיות כגון 'או' ו- 'וגם' על הביטויים הלוגיים שלנו, וע"י כך ליצור שרשור של תנאים לוגיים מורכבים.

[](https://carbon.now.sh/?bg=rgba(178%2C192%2C204%2C0.98)&t=seti&wt=none&l=text%2Fx-csrc&ds=true&dsyoff=20px&dsblur=46px&wc=true&wa=false&pv=0px&ph=0px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=1x&wm=false&code=import%2520java.util.Scanner%253B%250A%250Apublic%2520class%2520Main%2520%257B%250A%2520%2520%2520%2520static%2520public%2520void%2520main(String%2520%255B%255D%2520args)%2520%257B%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520a%2520%253D%25203%253B%250A%2520%2520%2520%2520%2520%2520%2520%2520int%2520b%2520%253D%25207%253B%250A%250A%2520%2520%2520%2520%2520%2520%2520%2520boolean%2520res%2520%253D%2520((a%252Bb)%2520%253E%25203)%2520%257C%257C%2520((a*b)%2520%253D%253D%252018)%253B%250A%2520%2520%2520%2520%257D%250A%257D%250A)

נשים לב לקטע קוד המופיע לעיל:

השימוש באופרטור ' || ' שקול למונח 'או'.

ניתן לראות כי ישנם שני תנאים לוגיים.

* התנאי השמאלי בודק האם הסכום של a ושל b גדול מ 3.
* התנאי הימני בודק האם המכפלה של a ושל b שווה בדיוק ל- 18.   
  אופרטור '==' בודק השוואה, כלומר, האם שני ערכים שווים זה לזה.
* התנאי הימני אכן מתקיים. 7+3 = 10 - אכן גדול מ-3.
* התנאי השמאלי אינו מתקיים. 3\*7 = 21 - שונה מ-18.

מכאן, הקומפיילר מחשב את התוצאה הכוללת של הביטוי הלוגי בצורה הבאה:

((a+b) > 3) || ((a\*b) == 18)

((3+7) > 3) || ((3\*7) == 18)

((10) > 3) || ((21) == 18)

true || false

תוצאת חישוב ביטוי לוגי 'או' של ערך אמת או שקר, תחזיר **אמת**.

ולכן, התוצאה הסופית של הביטוי הלוגי שלנו היא **אמת.**

true

בסוף הריצה, המשתנה res יכיל את הערך true.

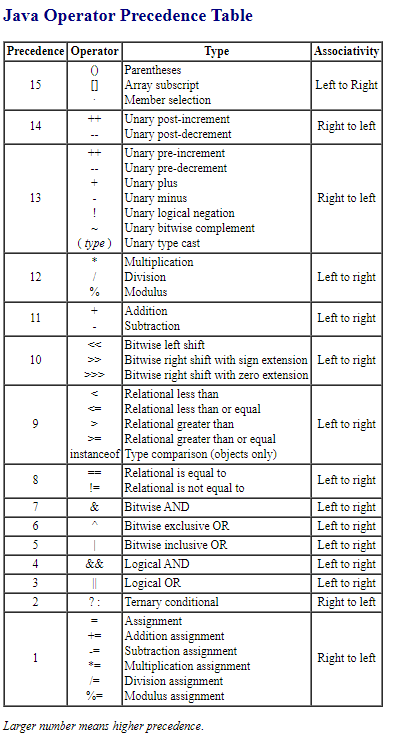
אופרטור נפוץ ונוסף בביטויים לוגיים מורכבים הוא האופרטור && - "וגם".

firstCondition && secondCondition

אופרטור זה בודק האם שני התנאים (מצד ימין ומצד שמאל) מתקיימים.  
 במידה וכן מחזיר אמת, אחרת מחזיר שקר.

**סדר קדימויות**

ממש כמו בשיעורי חשבון בחטיבה, גם בשפת Java יש סדר קדימויות לביטויים.

[](http://www.cs.bilkent.edu.tr/~guvenir/courses/CS101/op_precedence.html)

נוכל לבצע שימוש חכם בסוגריים ובסוגים השונים של הביטויים הלוגיים שלנו, על מנת לבצע את הפקודות הנדרשות בסדר אותו נרצה לבדוק.

**תרגול**

1. כתוב תוכנית המבצעת את הפעולה הבאה:

מגדירה שני משתנים מסוג int, a ו- b. יש לאתחל אותם בערכים כלשהם.

לאחר מכן יש ליצור 4 משתנים. בשמות:

**int first**

**double second**

**boolean third**

**boolean fourth**

את המשתנה first נאתחל בצורה הבאה:  
 a פחות b ועוד 8

את המשתנה second נאתחל בצורה הבאה:

ראשית נחשב את a בחזקת b.  
 את התוצאה של חישוב זה נכפול ב-3.  
 לאחר מכן נוסיף למספר שיצא 8 ונחסר ממנו 14.  
 לבסוף, נחלק הכל ב-2  
   
 \*חקרו באינטרנט כיצד נבצע חישוב חזקה

את המשתנה third נאתחל בצורה הבאה:

במידה והמשתנה first גדול או שווה ל 0 נאתחל אותו לfalse.  
 אחרת, נאתחל אותו לtrue.

את המשתנה fourth נאתחל בצורה הבאה:  
 במידה והמשתנה third אמת וגם המספר first הוא זוגי, אזי נאתחל באמת, אחרת שקר.