

תקציר הוראות השפה - Java

טיפוסי משתנים בסיסיים (פרימיטיביים) בשפת java:

int	(בתחום $\pm 2,147,483,648$)	מספרי שלם
long		מספרי שלם ארוך
double		מספר ממשי
char		תו
String		מחרוזת

<p>static final ; ערך-הקבוע = שם-הקבוע טיפוס-נתונים</p> <p>static final int N = 10;</p> <p>static final double X = 2.34;</p> <p>Integer.MAX_VALUE // הערך השלום הגדול ביותר</p> <p>Integer.MIN_VALUE // הערך השלום הקטן ביותר</p>	<p>קבועים (של המחלקה, static) :</p> <p>נהוג לרשום שמות קבועים באות גדולה</p> <p>קבועים בשפה:</p>
--	--

<p>int a, b;</p> <p>double x, y;</p>	הגדרת משתנים:
--	----------------------

מבנה התכנית בשפת Java:

<pre>import java.util.Scanner; public class שם-המחלקה { public static void main(String[] args) { הגדרת המשתנים; Scanner input - new Scanner (System.in); // אובייקט הקלט הוראה; הוראה; } }</pre>	<p>שם לב להזחות identification (קפיצת tab של תחילת השורה)</p>
--	---

הערות

// הערת שורה	
/* הערה */	הערת קטע הנמשכת על פני כמה שורות
	*/

ב- java יש חשיבות לגודל אות (case sensitive)

•

קלט/פלט:

System.out.print ("מחרוזת"); System.out.println ("מחרוזת");	פלט מחרוזת
System.out.print (שם-משתנה); System.out.print (שם-משתנה + " " + שם-משתנה); System.out.println (שם-משתנה); System.out.println (שם-משתנה + " " + שם-משתנה);	הדפסת תוכן משתנה
System.out.print (שם-משתנה + " מחרוזת "); System.out.print (" מחרוזת " +שם-משתנה+ " מחרוזת"); System.out.println (שם-משתנה + " מחרוזת "); System.out.println ("מחרוזת"+שם-משתנה+" מחרוזת");	הדפסת משתנים ומחרוזות
int a = input.nextInt();	קלט מספר שלם
double x = input.nextDouble();	קלט מספר ממשי
char tav = input.next().charAt(0);	קלט מספר תו
String str = input.next (); //קלט עד תו הרווח הראשון// String str = input.nextLine (); //קלט שורה	קלט מחרוזת

<ul style="list-style-type: none"> • בכל הוראת קלט ניתן לקלוט בדיוק משתנה אחד. • לפני כל הוראת קלט יש להציג בקשה : System.out.print (" → הקש מספר שלם"); int a = input.nextInt();	דגשים
---	-------

הוראות השמה:

תו	מחרוזת	מספר	
char ch = ' '; // תו רווח	String str = "" ; // מחרוזת ריקה	int a = 0;	אתחול בהגדרה
ch = 'b' ;	str = "hello";	a = 5;	הצבת קבוע

פעולות חישוביות:

int a, b, c; double x, y, z;	נתונים המשתנים הבאים :
c = a + b;	חיבור
c = a - b;	חיסור
c = a * b;	כפל

<p>שלם / שלם ← שלם</p> <p>ממשי / ממשי ← ממשי</p> <p>שלם / ממשי ← ממשי</p> <p>שלם / שלם (double) ← ממשי</p>	<p>חילוק:</p>
<p>$z = x / y;$ // לפחות אחד המשתנים המשתתפים בפעולה החישוב חייב להיות ממשי</p> <p>$z = (\text{double})a / b;$</p>	<p>חילוק בממשיים</p> <ul style="list-style-type: none"> המרה מפורשת
<p>$c = a / b;$</p>	<p>חילוק בשלמים (מנה)</p>
<p>$c = a \% b;$</p>	<p>שארית</p>
<ul style="list-style-type: none"> המרה - ניתן להמיר משלם לממשי ולהיפך בהמרה מפורשת (casting). 	

קיצורים:

<p>$a ++;$</p>	<p>הגדלה עצמית</p>
<p>$a --;$</p>	<p>הקטנה עצמית</p>

הוראות השפה:

<p>שווה $==$</p>	<p>שונה $!=$</p>	<p>סוגי יחס (קרא משמאל לימין)</p>
<p>גדול מ- $>$</p>	<p>גדול או שווה $>=$</p>	
<p>קטן מ- $<$</p>	<p>קטן או שווה $>=$</p>	

בלוק הוראות:

<pre>{ בלוק הוראות }</pre>

<pre>if (a > 5) b = b + 1;</pre>	<p>(פסוק-לוגי) if ביצוע;</p>	<p>אם ... (הוראה יחידה)</p>
<pre>if (a == 3) { c = b * 2; System.out.println (c + " : התוצאה"); }</pre>	<p>(פסוק-לוגי) if { הוראה; הוראה; }</p>	<p>אם ... (בלוק הוראות)</p>

<pre> if (a != 5) b = b + 1; else b = b - 1; </pre>	<pre> if (פסוק-לוגי) ביצוע1; else ביצוע2; </pre>	<p>אם... אחרת... (הוראה יחידה)</p>																					
<pre> if (a > 3) { System.out.print ("הקש מספר →"); b = input.nextInt(); } else { b = a; c = c + 1; } </pre>	<pre> if (פסוק-לוגי) { הוראה1; הוראה2; } else { הוראה3; הוראה4; } </pre>	<p>אם... אחרת... (בלוק הוראות)</p>																					
<pre> switch (a) { case 1 : System.out.println ("one"); case 2 : System.out.println ("two"); break ; case 3 : case 4 : case 5 : System.out.println ("five"); case 6 : System.out.println ("six"); break ; default: System.out.println ("bigger then six"); } </pre> <p style="text-align: right;">הפלט :</p> <table border="1" data-bbox="178 1339 632 1415"> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>אחר</td> </tr> <tr> <td>one</td><td>two</td><td>five</td><td>five</td><td>five</td><td>six</td><td>bigger then six</td> </tr> <tr> <td>two</td><td></td><td>six</td><td>six</td><td>six</td><td></td><td></td> </tr> </table>	1	2	3	4	5	6	אחר	one	two	five	five	five	six	bigger then six	two		six	six	six			<pre> switch (משתנה) { case 1-ערך : ביצוע-1; break; case 2-ערך : ביצוע-2; break; case 3-ערך : הוראה ; הוראה; break; case 4-ערך : ביצוע-4; break; default : ביצוע-5; break; } </pre>	<p>ברירת החלטה</p> <ul style="list-style-type: none"> המשתנה הנבדק הוא מטיפוס סדור: מספר שלם או תו ההוראות מתבצעות עד ה- break ניתן לבחון רק ערכים בדידים
1	2	3	4	5	6	אחר																	
one	two	five	five	five	six	bigger then six																	
two		six	six	six																			

קשרים לוגיים

&&	וגם
	או
!	לא
<pre> if (a > 3 && (b == 5 c != 0)) ... </pre>	

משתנה בוליאני:

<p>השמה למשתנה בוליאני:</p> <pre>boolean b = (num1 > num2);</pre>	
<pre>boolean b; if (num1 > num2) b = true ; else b = false ;</pre>	<p>ביטויים שקולים:</p>
<pre>boolean found = ... ; // true או false ;</pre>	
<pre>if (found) ... if (found == true) ...</pre>	<p>ביטויים שקולים</p>
<pre>if (! found) ... if (found == false) ...</pre>	<p>ביטויים שקולים</p>

לולאות:

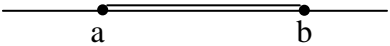
<pre>for (קידום-אינדקס ; ערך-סיום <= אינדקס ; ערך-תחילי = אינדקס) { הוראה; הוראה; } ----- int a, sum = 0; for (i = 1 ; i <= 5 ; i++) { System.out.print(" מספר שלם → "); a = input.nextInt (); sum = sum + a; }</pre>	<p>לולאת for (לולאה עולה)</p>
<pre>for (הקטנת-אינדקס ; ערך-סיום >= אינדקס ; ערך-תחילי = אינדקס) { הוראה; הוראה; } ----- double x, sum = 0; for (i = 5 ; i >= 1 ; i --) { System.out.print (" מספר ממשי → "); x = input.nextDouble (); sum = sum + x; } System.out.println (sum + " : סכום המספרים");</pre>	<p>לולאת for (לולאה יורדת)</p>

<pre> while (ביטוי-לוגי) { הוראה; הוראה; } ----- int sum = 0; System.out.print (" מספר ראשון → "); int a = input.nextInt (); while (a != 999) { sum = sum + a; System.out.print (" עוד מספר → "); a = input.nextInt (); } System.out.println (sum + " : סכום המספרים"); </pre>	<p>לולאת while</p>
--	---------------------------

פונקציות מתמטיות:

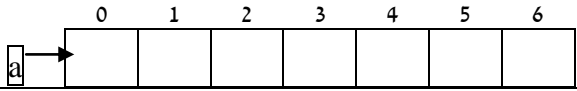
דוגמא	תחביר	מתקבל	מוחזר	טיפוס הערך
<code>int b = Math.abs(a); double y = Math.abs(x);</code>	<code>Math.abs(x)</code>	שלם ממשי	שלם ממשי	ערך מוחלט $ x $
<code>if (Math.sqrt (x) > y) ...</code>	<code>Math.sqrt (x)</code>	שלם ממשי	ממשי	שורש ריבועי \sqrt{x}
<code>double p = Math.pow (x, b);</code>	<code>Math.pow (x, y)</code>	שלם ממשי	ממשי	חזקה x^y
<code>System.out.println ((int)Math.round(7.853));</code>	<code>(int) Math.round (x)</code>	ממשי	שלם ארוך	עיגול לשלם הקרוב ($4 \leftarrow 3.67$)
<code>if (x == (int)x) ... ;</code>	<code>(int) x</code>	ממשי	שלם	החלק השלם ($3 \leftarrow 3.67$)
<code>int small = Math.min (12, a);</code>	<code>Math.min(x, y)</code>	שלם ממשי	שלם ממשי	הערך הקטן מבין השניים
<code>int big = Math.max (12, a);</code>	<code>Math.max(x, y)</code>	שלם ממשי	שלם ממשי	הערך הגדול מבין השניים
<code>int range = y - x + 1; // תחום המספרים</code> <code>int num = (int)(Math.random() * range) + x;</code> דוגמה : מספר שלם אקראי בין 20 - 50 (כולל): התחום : $31 \leftarrow 50 - 20 + 1$				מספר אקראי שלם בין x ו- $y - x$ ($x < y$)
<code>int num = (int)(Math.random() * 31) + 20;</code>				

מספר אקראי - שימוש באובייקט Random

<code>import java.util.Random;</code>	לפני התכנית:
<code>Random rnd = new Random();</code>	בתוך גוף בתכנית: יצירת עצם:
<code>x = rnd.nextInt (n);</code>	מספר שלם בתחום 0 עד n (לא כולל) $0 \leq x < n$
<code>x = rnd.nextInt (n) + 1;</code>	מספר שלם בתחום 1 עד n (כולל) $1 \leq x \leq n$
<code>n = b - a + 1;</code> <code>x = rnd.nextInt (n) + a;</code>	מספר שלם בתחום a עד b (כולל) 

פעולות

<code>static void procName (פרמטרים)</code> { }	פעולה שאינה מחזירה ערך (מוחזר ערך void)
<code>static void aaa (int a, int b, double x)</code> { int i; : }	
<code>static funcName (פרמטרים) טיפוס- מוחזר</code> { : return ערך להחזרה; }	פעולה המחזירה ערך
<code>static double sum (int a, int b, double x)</code> { double total ; total = a + b + x; return total; }	

<code>arr = new</code> טיפוס-נתונים <code>[10]</code> טיפוס-נתונים	מערך חד-ממדי
<code>int [] a = new int [7] ;</code> <code>double [] x = new double [N];</code> <code>char [] arrChar = new char [25];</code>	הגדרה
<code>x[2] = 2 * a[0] ;</code>	פנייה לתא
	
<code>for (int i = 0 ; i < a.length ; i ++)</code> <code> a[i] = i;</code> גודל המערך הוא <code>a.length</code> מספר התא האחרון במערך הוא <code>a.length - 1</code>	סריקת מערך

<code>arr = new</code> טיפוס-נתונים <code>[5][7]</code> טיפוס-נתונים	מערך דו-ממדי (מטריצה)
<code>int [][] mat = new int [6][7] ;</code> <code>double [][] x = new double [N][M];</code>	
סכום התא ה"ראשון" והתא ה"אחרון" במערך בגודל 6 שורות ו-7 עמודות: <code>mat[3][4] = mat [0][0] + mat [5][6];</code>	פניה לתא
<code>mat.length</code> \Rightarrow מספר השורות במטריצה <code>mat[i].length</code> \Rightarrow מספר העמודות בשורה <code>i</code> <code>for (int i = 0 ; i < mat.length ; i++)</code> <code> for (int j = 0 ; j < mat[i].length ; j++)</code> <code> mat [i][j] =</code>	סריקת המערך

String str ; שיים ♥ : String באות גדולה	מחרוזות
str = "bla-bla-bla" ; str = new String ("bla-bla-bla") ;	מחרוזות ב-Java היא אובייקט
String str1 = input.next(); קלט מילה בודדת String str2 = input.nextLine(); קלט משפט	קלט
פעולות שאינן משנות את המחרוזת	
num = str.length();	אורך
str = str1 + str2 + "aaa" ;	שרשור
boolean equal = str1.equals(str2) ; int n = str1.compareTo (str2) ; str1 > str2 ⇔ n > 0 str1 < str2 ⇔ n < 0 str1 == str2 ⇔ n = 0	השוואה
ch = str.charAt (i) ; 0 ≤ i < str.length()	תו i במחרוזת
int place = str.indexOf (ch) ; // חיפוש הראשון	חיפוש תו
int place = str.indexOf (subStr); // חיפוש הראשון	חיפוש תת-מחרוזת
הערה : indexOf מחזיר תמיד את הראשון מתחילת המחרוזת. אם לא נמצא יוחזר -1	
int place = str.indexOf (ch, fromPlace); // חיפוש ממקום int place = str.indexOf (subStr, fromPlace);	חיפוש המופע הבא :
int place = str.lastIndexOf (ch); // חיפוש מהסוף int place = str.lastIndexOf (subStr);	חיפוש מהסוף :
int place = str.lastIndexOf (ch, from); // חיפוש מהסוף, ממקום int place = str.lastIndexOf (subStr, from);	