Java

*תקציר הוראות השפה ב-*

בשפת :java

טיפוסי משתנים בסיסיים (פרימיטיבים)

|  |  |
| --- | --- |
| **int** (±2,147,483,648 בתחום) | מספרי שלם |
| **long** | מספרי שלם ארוך |
| **double** | מספר ממשי |
| **char** | תו |
| String | מחרוזת |
|  | |
| ; ערך-הקבוע = שם-הקבוע טיפוס-נתונים **final static**  **static final int** N = 10;  **static final double** X = 2.34;  הערך השלום הגדול ביותר // Integer.MAX\_VALUE  הערך השלום הקטן ביותר // Integer.MIN\_VALUE | **קבועים** (של המחלקה, :)static  נהוג לרשום שמות קבועים באות  גדולה  **קבועים בשפה:** |
|  | |
| **int** a, b;  **double** x, y; | **הגדרת משתנים:** |

מבנה התכנית בשפת :Java

|  |
| --- |
| **import** java.util.Scanner;  שם-המחלקה **class public**  {  **public static void** main(String[] args)  {  ;הגדרת המשתנים  Scanner input - **new** Scanner (System.in); // הקלט אובייקט  ;הוראה  שים לב להזחות identification ;הוראה  (קפיצת tab של תחילת השורה) }  } |
| **הערות** |
| הערת שורה **//** |
| הערת קטע **/\* \*/** הערה **/\*** הנמשכת על פני כמה שורות  **\*/** |

(case sensitive)

 ב- **java** יש חשיבות לגודל אות

קלט/פלט:

|  |  |
| --- | --- |
| System.out.print ("מחרוזת"); System.out.println ("מחרוזת"); | פלט מחרוזת |
| System.out.print (משתנה-שם);  );שם-משתנה + " " + שם-משתנה( System.out.print  System.out.println (משתנה-שם); System.out.println (משתנה-שם + " " + משתנה-שם); | הדפסת תוכן משתנה |
| "); מחרוזת " + שם-משתנה( System.out.print "); מחרוזת " +שם-משתנה+ " מחרוזת(" System.out.print  "); מחרוזת " + שם-משתנה( System.out.println ");מחרוזת+"שם-משתנה"+ מחרוזת(" System.out.println | 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 |
| קלט עד תו הרווח הראשון// (); input.next = str String  String str = input.nextLine (); // שורה קלט | קלט מחרוזת |
|  | |
|  בכל הוראת קלט ניתן לקלוט בדיוק משתנה אחד.   לפני כל הוראת קלט יש להציג בקשה:  System.out.print ("שלם מספר הקש  ");  **int** a = input.nextInt(); | דגשים |

הוראות השמה:

|  |  |  |  |
| --- | --- | --- | --- |
| **תו** | **מחרוזת** | **מספר** |  |
| **char** ch = ' '; // רווח תו | מחרוזת ריקה // ; "" = str String | **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 |
|  |  |

|  |  |
| --- | --- |
| שלם / שלם  שלם  ממשי / ממשי  ממשי שלם / ממשי  ממשי  שלם / שלם(double)  ממשי | חילוק: |
| לפחות אחד המשתנים המשתתפים // y; / x = z  בפעולה החישוב חייב להיות ממשי //  z = (double)a / b; | חילוק בממשיים   המרה מפורשת |
| c = a / b; | חילוק בשלמים (מנה) |
| c = a % b; | שארית |
|  המרה - ניתן להמיר משלם לממשי ולהיפך בהמרה מפורשת .(casting) | |
| **קיצורים**: | |
| a ++ ; | הגדלה עצמית |
| a -- ; | הקטנה עצמית |

הוראות השפה:

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

בלוק הוראות:

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

|  |  |  |  |
| --- | --- | --- | --- |
| **if** (a != 5)  b = b + 1;  **else**  b = b – 1; | | )פסוק-לוגי( **if**  ;ביצוע1  **else**  ;ביצוע2 | **אם... אחרת ...**  (הוראה יחידה) |
| **if** (a > 3) | | )פסוק-לוגי( **if** | **אם... אחרת ...** |
| {  System.out.print ("מספר הקש  "); | | {  ;הוראה1 | (בלוק הוראות) |
| b = input.nextInt(); | | ;הוראה2 |  |
| } | | } |  |
| **else** | | **else** |  |
| { | | { |  |
| b = a; | | ;הוראה3 |  |
| c = c + 1; | | ;הוראה4 |  |
| } | | } |  |
| **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") ;  } |  | )משתנה( **switch**  {  ;ביצוע1- : ערך1- **case**  **break**; **case** 2-ערך : 2-ביצוע;  **break**; ;הוראה : ערך3- **case**  ;הוראה  **break**; **case** 4-ערך : 4-ביצוע;  **break**; **default** : 5-ביצוע;  **break**;  } | **ברירת החלטה**   המשתנה הנבדק  הוא מטיפוס  סדור: מספר שלם או תו  ההוראות  מתבצעות עד ה-  break |
|  | הפלט: | ניתן לבחון רק ערכים בדידים |

|  |  |
| --- | --- |
| **קשרים לוגיים** | |
| **&&** | **וגם** |
| **||** | **או** |
| **!** | **לא** |
| **if** ( a > 3 **&&** (b = = 5 **||** c != 0) ) … | |

משתנה בוליאני:

|  |  |
| --- | --- |
| השמה למשתנה בוליאני: | |
| boolean b = (num1 > num2);  ביטויים שקולים:  boolean b;  **if** (num1 > num2) b = true ;  **else**  b = false ; |  |
| boolean found = … ; // true או false ; | |
| ביטויים שקולים … (found) if … true) == (found if | if ( ! found ) … שקולים ביטויים if (found == false) … |

לולאות:

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

|  |  |
| --- | --- |
| )ביטוי-לוגי( **while**  {  ;הוראה ;הוראה  }  - - - - - - - - - - - - - - -  **int** sum = 0;  System.out.print ("ראשון מספר  ");  **int** a = input.nextInt ();  **while** (a != 999)  {  sum = sum + a;  System.out.print ("מספר עוד  "); a = input.nextInt ();  }  System.out.println (sum + " :המספרים סכום"); | **לולאת while** |

פונקציות מתמטיות:

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

- שימוש באובייקט Random

מספר אקראי

|  |  |
| --- | --- |
| **import** java.util.Random**;** | לפני התכנית:  בתוך גוף בתכנית: |
| Random rnd = **new** Random(); :עצם יצירת |
| x = rnd.nextInt (n); | מספר שלם בתחום 0 עד n  (לא כולל) n < x  0 |
| x = rnd.netxInt (n) + 1; | מספר שלם בתחום 1 עד n (כולל)  1  x  n |
| n = b - a + 1;  x = rnd.nextInt (n) + a; | מספר שלם בתחום a עד b (כולל)  a b |

פעולות

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



|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [10];טיפוס-נתונים **new** = arr [] טיפוס-נתונים | | | | | | | | | | | **מערך חד-ממדי** |
| **int** [] a = **new int** [7] ;  **double** [] x = **new double** [N];  **char** [] arrChar = **new char** [25]; | | | | | | | | | | | **הגדרה** |
| x[2] = 2 \* a[0] ; | | | | | | | | | | | **פנייה לתא** |
| 0 1 2 3 4 5 6 | | | | | | | | | | |  |
|  | | |  |  |  |  |  |  |  |  |
|  | a |  |
| **for** (**int** i = 0 ; i < a.length ; i ++) a[i] = i;  גודל המערך הוא .lengthשם-המערך  מספר התא האחרון במערך הוא 1 - length | | | | | | | | | | | **סריקת מערך** |
|  | | | | | | | | | | | |
| [5][7];טיפוס-נתונים **new** = arr [][] טיפוס-נתונים | | | | | | | | | | | **מערך דו-ממדי (מטריצה)** |
| **int** [][] mat = **new int** [6][7] ;  **double** [][] x = **new double** [N][M]; | | | | | | | | | | |  |
| סכום התא ה"ראשון" והתא ה"אחרון" במערך בגודל 6 שורות ו- 7 עמודות:  mat[3][4] = mat [0][0] + mat [5][6]; | | | | | | | | | | | **פניה לתא** |
| מספר השורות במטריצה  mat.length  מספר העמודות בשורה i  mat[i].length  **for** (**int** i = 0 ; i < mat.length ; i++)  **for** (**int** j = 0 ; j < mat[i].length ; j++) mat [i][j] = …. | | | | | | | | | | | **סריקת המערך** |

|  |  |
| --- | --- |
| שים :♥ String באות גדולה ; str String | **מחרוזות** |
| str = "bla-bla-bla" ;  str = **new** String ("bla-bla-bla") ; | מחרוזת ב- Java היא אובייקט |
|  |  |
| קלט מילה בודדת input.next(); = str1 String  String str2 = input.nextLine(); משפט קלט | קלט |
| **פעולות שאינן משנות את המחרוזת** | |
| num = str.length(); | אורך |
| str = st1 + 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) ; // הראשון חיפוש | חיפוש תו |
| **int** place **=** str.indexOf (subStr); // הראשון חיפוש | חיפוש תת-מחרוזת |
|  |  |
| הערה: indexOf מחזיר תמיד את הראשון מתחילת המחרוזת. אם לא נמצא יוחזר -1 | |
| **int** place = str.indexOf (ch, fromPlace); // ממקום חיפוש  **int** place = str.indexOf (subStr, fromPlace); | חיפוש המופע הבא: |
| **int** place = str.lastIndexOf (ch); // מהסוף חיפוש  **int** place = str.lastIndexOf (subStr); | חיפוש מהסוף: |
| **int** place = str.lastIndexOf (ch, from); // ממקום ,מהסוף חיפוש  **int** place = str.lastIndexOf (subStr, from); |  |