

Lesson 5 - Arrays

הנושאים הנלמדים:

מערכים חד מימדיים, מערכים דו מימדיים

הצורך במערכים

- קבל מהמשתמש 20 ציונים והדפס את אלו שגבוהים מהממוצע.
- קבל מהמשתמש 40 שמות והדפס אותם בסדר הפוך.

מה משותף לתרגילים הללו?

- כמות גדולה של נתונים שצריך לשמור (האם הגדרת 20 או 40 משתנים היא דבר הגיוני?).

מערך - הגדרה

- מערך הוא אוסף סדור של משתנים מאותו הסוג.
- בעת יצירת המערך הוא מקבל שם, גודל וטיפוס.
- המערכת ניגשת ל heap מוצאת מקום רציף, שומרת אותם, ומחזירה כתובת אל תוך השם.
- מערך הוא סטטי - גודלו נקבע בזמן אתחולו.
- כל איבר במערך הוא משתנה.

מערך - הגדרה

ככל משתנה למערך יש שני שלבים הצהרה ואיתחול

שלב 1 – הצהרה (הכרזה)

שם המערך [] <טיפוס>

למשל:

```
int[] numbers;  
double[] nums;
```

- הסוגריים המרובעים מציינים מערך.
- מוקצה מקום בזכרון עבור מצביע (כתובת) למערך.
- הכתובת תיווצר רק לאחר האיתחול וכרגע התוכן הוא null.

מערך - הגדרה

שלב 2 - אתחול

ההכרזה מגדירה שם וטיפוס ואינה מקצה תאים.
לאחר ההגדרה מקצים **מקום** בזיכרון באמצעות האופרטור `new` אשר
מקצה זיכרון מה `heap`.

`<name> = new <type>[<size>]`

`numbers = new int[20]; //array of integer in size 20`

`nums = new double[10]; //array of 10 doubles`

ניתן כמובן גם להגדיר ולאתחל באותה שורה (כמו משתנים):

`char[] chars = new char[30]; //array of 30 characters`

מערך - הגדרה

- כל תאי המערך (לאחר ה new) מאותחלים לערכי ברירת המחדל.
- int ל 0, double ל 0.0, boolean ל false, char ל \0, String ל null.
- איתחול על ידי הכנסת ערכים:

```
int[] nums = {1, 2, 4, 8, 19} ;
```

```
double[] numbers = {1.6, 2.8, 4.9, 8.6, 19.666} ;
```

```
String[] names = {"moshe", "dave", "yael", "yuri"} ;
```

- תוכנו יכול להשתנות. גודלו לא.

הגדרת מערך - זכרון

arr

null

שלב א' - הכרזה

```
int[] arr;
```

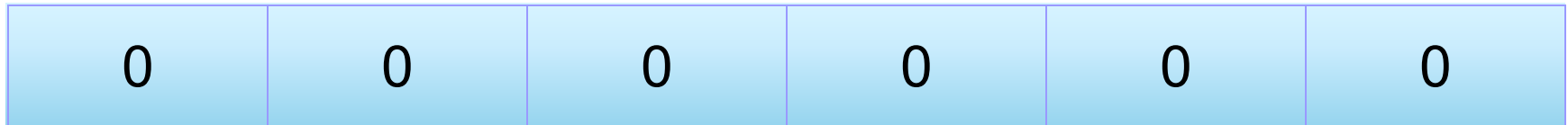
arr

כתובת

שלב ב' - איתחול

```
arr = new int[6];
```

heap



מערך - אינדקס

התא הראשון במערך הוא באינדקס 0 והתא האחרון נמצא באינדקס של גודל המערך פחות אחד. למשל:

```
int[] nums = {1, 2, 4, 8, 19} ;
```

יוצרת מערך בעל 5 תאים הנראה כך, לכל תא יש אינדקס וערך:

index	0	1	2	3	4
Value	1	2	4	8	19

מערך - אינקדס

**אינדקסים משמשים לקריאה וכתיבה ערכים לתוך/מ המערך.
למשל:**

`nums[3] = 0;`

1	2	4	0	19
---	---	---	---	----

`nums[0] = nums[4] = 5;`

5	2	4	0	5
---	---	---	---	---

x

6

`int x = nums[1] + nums[2]`

**פנייה למיקום שאינו קיים (`nums[5]`) שגיאת זמן הריצה – קריסה!
בזמן הקומפילציה השגיאה לא תתגלה!**

מערך - גודל

מערך - טיפוס מיוחד - יש מאפיין פנימי שמתאר את אורכו.

```
<שם המערך>.length;
```

חשוב: גישה לתכונת האורך על מערך שהוגדר אבל טרם
אותחל - שגיאת זמן ריצה - קריסה.

מערכים ולולאת for

```
int[] array = new int[20];
```

```
//Using for loop to populate the array
```

```
for(int i=0;i<array.length;i++) {  
    System.out.println("Please enter number " + (i+1));  
    array[i] = scanner.nextInt();  
}
```

קבלת ערכים למערך

```
//Using for loop to print the array
```

```
System.out.println("The original array is:");  
for(int i=0;i<array.length;i++) {  
    System.out.print "["+i+"]="+array[i] + " ";  
}
```

הדפסת מערך

מערכים ולולאת for

תרגיל: קבל מהשתמש 5 מספרים שלמים ושמור אותם במערך. לאחר מכן הדפס את המספר המקסימלי.

```
Scanner sc = new Scanner(System.in);
int[] numbers = new int[5];

for(int i=0;i<numbers.length;i++) {
    System.out.println("Please enter number " + (i+1));
    numbers[i] = sc.nextInt();
}

int max = numbers[0];
for(int i=1;i<numbers.length;i++) {
    if(numbers[i]>max)
        max = numbers[i];
}

System.out.println("The max number is " + max);
```

מערכים ולולאת for

הזזה מעגלית – הגדר מערך בגודל 5 וקבל לתוכו נתונים מהמשתמש, לאחר מכן בצע הזזה מעגלית של המספרים כך שכל תא יכנס לתא העוקב לו והתא האחרון לתא הראשון.

לדוגמה עבור המערך:

1	2	4	8	19
---	---	---	---	----

נקבל את המערך הזה:

19	1	2	4	8
----	---	---	---	---

מערכים ולולאת for

```
Scanner scanner = new Scanner(System.in);
int[] array = new int[5];

//Using for loop to populate the array
for(int i=0;i<array.length;i++) {
    System.out.println("Please enter number " + (i+1));
    array[i] = scanner.nextInt();
}

//Circular movement
int temp = array[array.length-1];
for(int i=array.length-2;i>=0;i--) {
    array[i+1] = array[i];
}
array[0] = temp;

System.out.println("The result array is:");
for(int i=0;i<array.length;i++) {
    System.out.print("[ "+i+" ]"+"="+array[i] + " ");
}
```

Values & references

השמה בין משתנים פרימיטיביים - העתקת ערכים (values) - התוכן של משתנה מועתק לתוך משתנה אחר שנמצא בכתובת אחרת:

```
int x=9;  
int y=x;  
x=1;
```

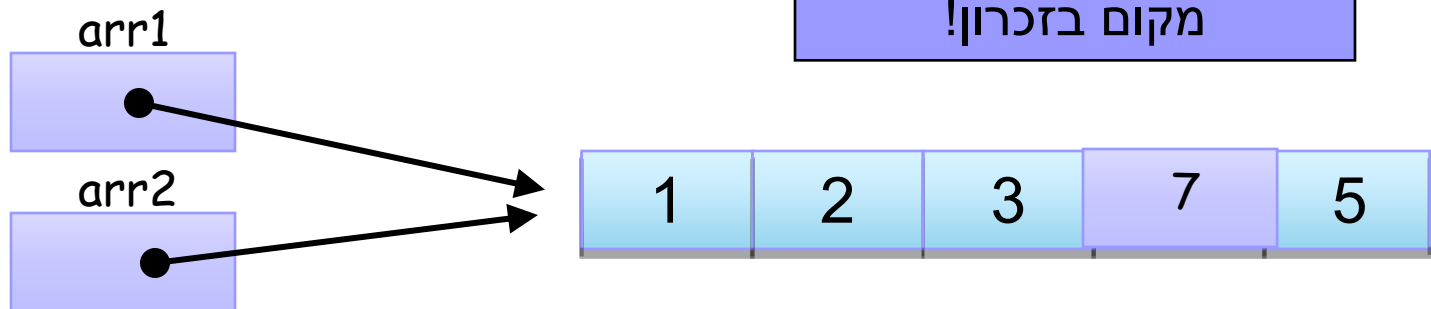


כל שינוי במשתנה אחד לא ישפיע על המשתנה השני כי הם שמורים במקומות שונים בזכרון.

Values & references

השם של מערך הוא בעצם כתובת (reference) למקום אחר בזכרון
לכן בהשמה בין מערכים איננו מעתיקים את הערכים שלהם אלא אנו
מייצרים מצביע נוסף לאותו מקום בזכרון.

```
int[] arr1 = {1,2,3,4,5};  
int[] arr2 = arr1;  
arr2[3] = 7;
```



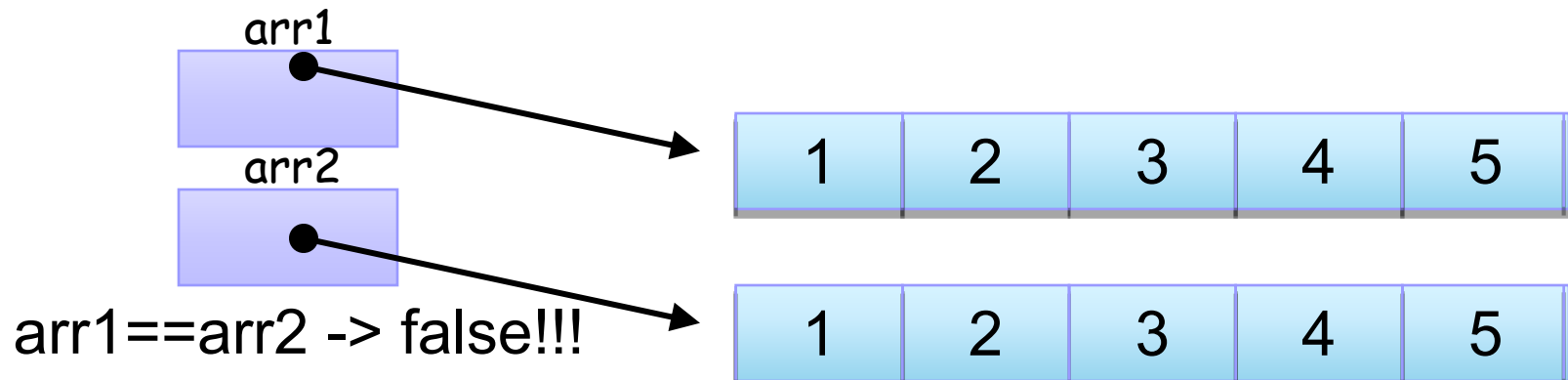
Values & references

השוואה בין פרימיטיבים - נבדקים הערכים (values)
השוואה בין מערכים - נבדקות הכתובות (reference).

כך שהשוואה בין מערכים אינה משווה תכנים! היא רק בודקת אם הם מצביעים לאותו מקום בזכרון.

```
int[] arr1 = {1,2,3,4,5};
```

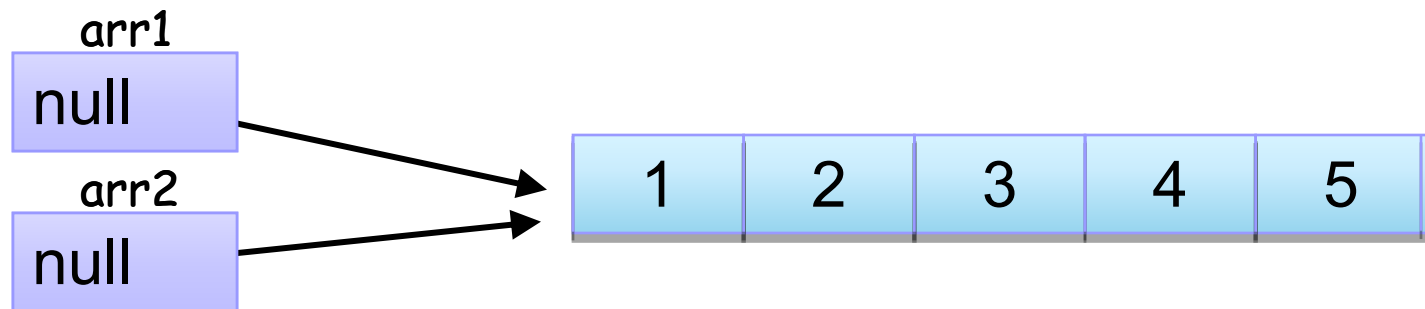
```
Int[] arr2 = {1,2,3,4,5};
```



References

אני יכול שיהיה לי יותר ממצביע אחד למערך, אך מה קורה למערך
שאף אחד אינו מצביע עליו?
למשל, מה יקרה כאן:

```
int[] arr1, arr2;  
arr1 = arr2 = new int[5];  
arr1 = arr2 = null;
```



המנקה זכרון *Garbage Collector* של ג'אווה שומר טבלה עם מספר מצביעים לכל זכרון שהוקצה ב *heap* כאשר יש זכרון עם מספר אפס של מצביעים, אין איך לגשת אליו והוא משחרר אותו אוטומטית.

בונוס – מערכים דו מימדיים

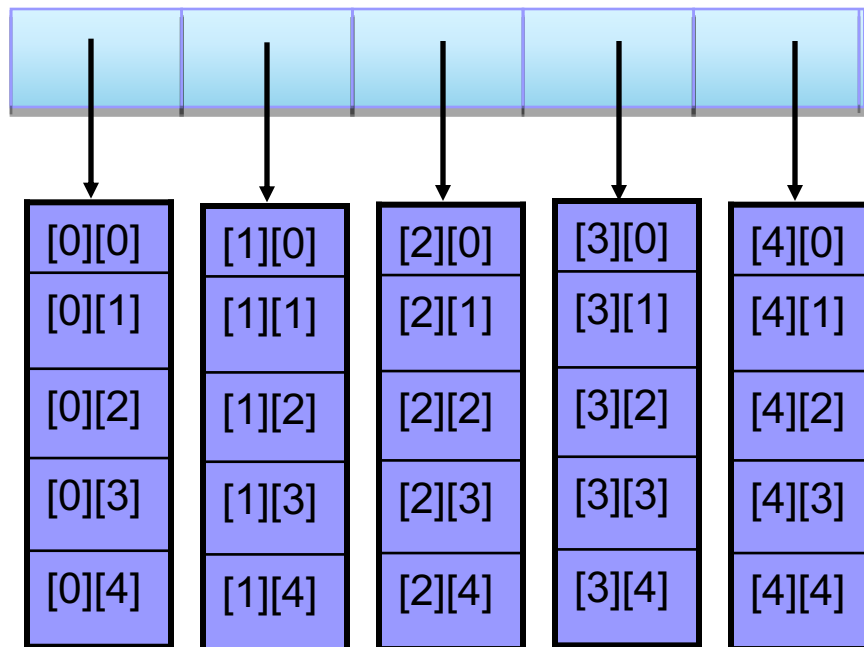
מערכים דו מימדיים

- המערך אותו הכרנו נקרא מערך חד-מימדי
- מערך דו מימדי הוא מערך שיש לו שני אינדקסים
- ניתן להסתכל עליו כעל טבלה עם שורות ועמודות
- $arr[i][j]$ – שורה i ו- j עמודה

<code>arr[0][0]</code>	<code>arr[0][1]</code>	<code>arr[0][2]</code>
<code>arr[1][0]</code>	<code>arr[1][1]</code>	<code>arr[1][2]</code>
<code>arr[2][0]</code>	<code>arr[2][1]</code>	<code>arr[2][2]</code>

מערכים דו מימדיים בזכרון

- בזכרון שמורים המערכים הדו מימדיים כמערך של מערכים.
- כלומר ישנו מערך (נניח השורות) שבו כל תא בעצם מצביע למערך אחר (נניח העמודות).
- זו הסיבה שלא חייבים אותו מספר עמודות בכל שורה.



אתחול המערך הדו מימדי

- ניתן להגדיר מערך דו מימדי המאותחל לערכי ברירת המחדל
 - `int[][] nums = new int[10][5];`
- ניתן גם להגדיר ולאתחיל מערך דו מימדי באמצעות הכנסת ערכים ראשוניים (מספר העמודות בכל שורה לא קבוע).
 - `Int[][] nums = {{3,6},{6,8,9},{3}};`
 - `String[][] strs = {"mosh","John","Dave"}, {"Yael","riki"}`

תכונות המערך הדו מימדי

■ ניתן לקבל את מספר השורות במערך באופן הבא:

■ `<שם המערך>.length`

■ ניתן לקבל את מספר העמודות בכל שורה באופן הבא:

■ `<שם המערך>[אינדקס השורה].length`

מספר העמודות בשורה ה $0 < \text{matrix}[0].length$

קבלת מערך דו-מימדי והדפסתו

```
Scanner scanner = new Scanner(System.in);

final int SIZE = 10;

int[][] matrix = new int[SIZE][SIZE];
for(int row = 0; row < matrix.length; row++) {
    for(int col = 0; col < matrix[row].length; col++) {
        System.out.println("Enter number for matrix in row " +
            (row+1) + " and column " + (col+1));
        matrix[row][col] = scanner.nextInt();
    }
}

System.out.println("The matrix you entered is");
for(int row = 0; row < matrix.length; row++) {
    for(int col = 0; col < matrix[row].length; col++) {
        System.out.print(matrix[row][col] + "\t");
    }
    System.out.println();
}
```