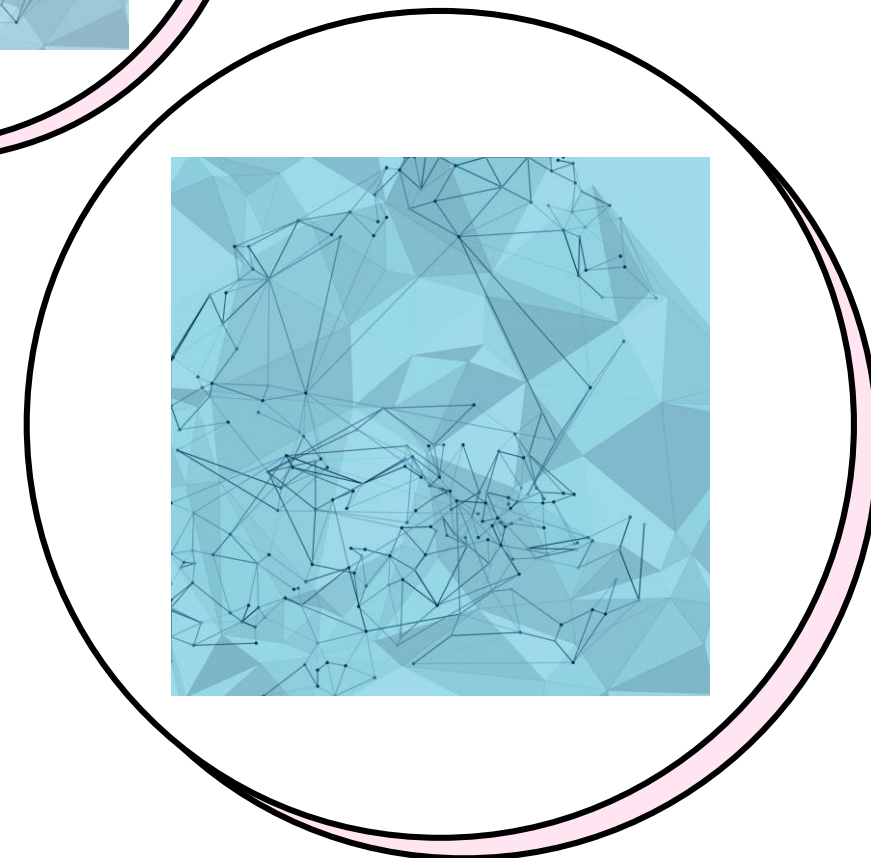
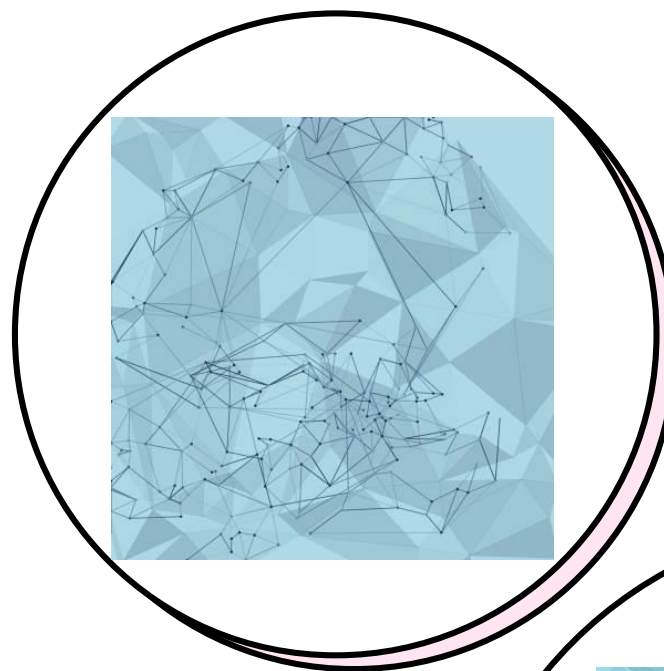


# תרגול 3

חומר תיאורטי +  
תרגילים





## מערכים דו-מימדיים

- בשבוע שעבר דיברנו על מערכים חד-מימדיים, תזכורת:

```
int[] arr = new int[10];    //a new array named arr with 10 int cells.
```

כעת, נרחיב זאת למערך דו מימדי כלומר, מערך הבנוי ממערכים חד-מימדיים.

דוגמאות:

```
int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7,} }; // it's possible to create a 2D  
array in Java where the subarrays (the 1D arrays) have different lengths.
```

```
String[][] foo = new String[5][6];
```

```
String[][] myStrings = { {"This", "is"}, {"Tirgul", "3"} };
```

ניתן לרשום את "3" כאיבר בתוך המערך של המחרוזות כיוון שכפי שכבר למדנו מחרוזת היא

רצף של תווים (char) ומחרוזת עם תו יחיד זה אכן רצף של תווים באורך 1.////



# המשך מערכים דו-מימדיים

➤ מה יודפס בקטע קוד הבא:

```
String[][] arr = { {"Silly", "Mango", "Art"}, {"Rabbit", "Tiger"} };  
for (int i = 0; i < arr.length; i++) {  
    for (int j = 0; j < arr[i].length; j++) {  
        System.out.print(arr[i][j].charAt(0));  
    }  
}
```





# לולאת For each

- כבר דיברנו על 3 סוגים של לולאות (for, while, do while), כעת נדבר על סוג רביעי שהוא יותר קצר ובמקרים מסוימים נוח יותר לשימוש.

```
int arr[] = { 10, 50, 60, 80, 90 };
```

```
for (int element : arr) // why can we write it without {}?
```

```
System.out.print(element + " ");
```

הקוד עובר על כל האיברים במערך ללא צורך באינדקסים, כלומר עובר איבר איבר משמאל למערך ועד לסופו.





# פונקציות

- ❖ פונקציה היא אוסף הוראות לביצוע שמממשות רעיון משותף.
- ❖ המוטיבציה לשימוש בפונקציות היא כדי שהקוד יהיה קריא, מודולרי(מחולק לקטעי קוד קטנים יותר) ויהיה ניתן לשימוש חוזר בכך שבמקום לכתוב את אותו קוד כמה וכמה פעמים אז נוכל לרכז זאת בפונקצייה שתבצע פעולה כלשהי ונקרא לה בכל פעם שנצטרך לבצע את הפעולה.
- ❖ הפונקציה מוגדרת כך: הצהרה, מימוש, שימוש(דוגמאות בעמוד הבא).
- ❖ אם הפונקצייה אמורה להחזיר משהו אזי נשתמש במילת "return" כדי ש-מתי שנקרא לפונקצייה אז נקבל ערך חזרה כלשהו, אם הפונקצייה מוגדרת כvoid אזי היא לא תחזיר כלום.





## פונקציות - דוגמאות

```
public class Main {  
    public int addNumbers(int num1, int num2) {// שלב ההצהרה  
        return num1 + num2; // שלב המימוש  
    }  
  
    public static void main(String[] args) { // void doesn't return anything  
        int sum = addNumbers(5, 3); // שלב השימוש  
        System.out.println("The sum is " + sum); // Outputs "The sum is 8"  
    }  
}
```





## פונקציות - המשך

- בדוגמה הקודמת הגדרנו את הפונקצייה בצורה רגילה ולא כסטטית (static), מילה שמורה זו קשורה לנושא "תכנות מונחה עצמים" וכעת אתם לא אמורים לדעת אותה אבל, נסביר ברגל אחת למה הכוונה:
- פונקצייה סטטית מאפשרת למחלקות חיצוניות לקרוא לאותה פונקציה מבלי ליצור אובייקט (תלמדו בהמשך (של המחלקה).
- חשוב לדעת שגם אם לא הבנתם לגמרי למה הכוונה אין טעם להילחץ, נחזור לזה עוד כמה שבועות.





# unit tests – טסטים

- הטסטים (בדיקות) מאפשרות לנו לבדוק חלקים קטנים של הקוד (לרוב פונקציות שהגדרנו) באופן עצמאי, כדי לוודא שהם עובדים כמצופה.
- דוגמאות:

```
import static org.junit.Assert.assertEquals;
import org.junit.Test;
public class SimpleTest {
    @Test
    public void testAddNumbers() { // testing the “addNumbers” function we defined 2 slides ago
        int a = 2;
        int b = 3;
        int sum = addNumbers(a, b);
        assertEquals(5, sum); // compare 5 to the result of the function(sum)
    }
}
```







# תרגילים

**שאלה 1:** כתבו פונקציה שמקבלת מערך של מספרים שלמים ומחזירה את סכום של איברי המערך: (אם משתמשים בלולאות אז להשתמש בלולאת for each)

```
public static int arraySum(int [] arr)
```

דוגמה: אם איברי המערך הם: 1,2,3,4,5 אז הסכום שווה 15. בדוק את הפונקציה.

**שאלה 2:** כתבו פונקציה שמקבלת מערך של מספרים שלמים ומחזירה את ממוצע של איברי המערך:

```
public static double arrayMean(int [] arr)
```

דוגמה: אם איברי המערך הם: 1,2,3,4,5 אז הממוצע שווה 3. בדוק את הפונקציה.





## עוד תרגילים

**שאלה 3:** כתבו פונקציה שמקבלת מערך של מספרים שלמים ורושמת איברים בסדר הפוך באותו מערך:

```
public static void reverse(int [] arr)
```

דוגמה: קלט – [1,2,3,4,5] פלט – [5,4,3,2,1]

**שאלה 4:** כתבו פונקציה שמקבלת מטריצה (מערך דו-מימדי) ומחשבת את סכום המטריצה:

```
public static int sum(int[][] array)
```





## עוד לא סיימנו עם התרגילים

שאלה 5: כתבו פונקציה שמקבלת מטריצה והופכת כל איבר באינדקס ההפוך לו כלומר, עבור איבר  $a_{i,j}$  במערך הנתון היא תעביר אותו ל  $a_{j,i}$  במערך החדש או במילים אחרות, שמה את השורות של המטריצה כעמודות ואת העמודות כשורות.

```
public static int[][] transpose(int[][] array)
```

מי שהצליח את התרגיל יכול לקחת לעצמו יום כיף מתי שילמדו "שיחלוף" בקורס אלגברה לינארית 1 (תפגשו את הפעולה בהמשך הסמסטר).





# תרגילים אחרונים

כתבו טסטים מתאים לשאלות 1 ו-2.  
דוגמה להתחלת הטסט לשאלה 1:

```
@Test
```

```
public void testArraySum()
```

**(שאלת אתגר):** כתבו פונקציה שמקבלת מערך של מספרים שלמים וממיינת אותו כך שהמספרים הזוגיים יהיו בתחילת המערך, והמספרים האי-זוגיים יהיו בסוף המערך.

```
public static void evenOddSort(int [] arr)
```

דוגמה: קלט – 1,2,3,4,5 פלט – 2,4,3,1,5

