

## חומרים שהוכנו על-ידי משתתפי קורס מורים מובילים תשע"ד

ניתן להשתמש בחומרים לצורך הוראה בלבד.

**לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי**

ללא קבלת אישור מראש מצוות הפיתוח

## מערך שיעור בנושא: מערך של עצמים

**כתיבה ועריכה:**

**אוי גרינולד**

נקודות עיקריות בשיעור הקודם: כתיבת מחלקות שמהוות טיפוס נתונים וביניהן המחלקה Point שמהווה טיפוס נתונים מופשט שמתאר נקודה במישור. תכונות המחלקה הן שיעורי x,y של הנקודה מטיפוס ממשי.

שיעורי בית שניתנו לקראת השיעור הנוכחי: לסיים את כתיבת המחלקה Point כולל תכנית בדיקה לכל פעולות המחלקה ולהביא אותה על אמצעי אחסון.

הרציונל של מערך השיעור: הרחבת שימוש במערך ממערך של טיפוסים בסיסיים: שלם, ממשי, תו למערך של עצמים או יותר מדויק מערך אשר מכיל כל תא שלו הפנייה לעצם מטיפוס המחלקה.

השיעור הוא שיעור מעבדה.

מטרות השיעור:

- התלמיד ידע ליצור מערך של עצמים
- לאתחל כל תא במערך לעצם מטיפוס המחלקה
- לסרוק את המערך ולשלוף את התכונה של העצם שנמצא בתא מסוים במערך.
- יישום תבניות אלגוריתמיות שנלמדו על מערך על מערך של עצמים.

נקודות עיקריות בשיעור הנוכחי:

- יצירת מערך של עצמים, במקרה זה של נקודות
- להבין שגישה לתא במערך זו הפנייה לעצם מטיפוס נקודה
- להבין שיצירת מערך לא יוצרת ישירות הפנייה לנקודות מכל תא במערך, אלא מציבה null וכדי להציב נקודות בתאי המערך יש לאתחל כל נקודה ואז להציבה בתא המתאים.

נקודות עיקריות בשיעור העוקב:

- תרגול ויישום בעיות אחרות בשילוב יצירת מערך של עצמים והפעלת פעולות עליו
- יישום בעיות אחרות על אותו מבנה נתונים, כלומר על מערך של Point

קשיים צפויים:

- הבנת הנקרא של מסיחי המטלה
- זיהוי השגיאה שמציגה התוכנה
- הבחנה בין יצירת מערך של טיפוס המחלקה לבין יצירתו והחובה לאתחל כל תא במערך לעצם מהטיפוס

**תיאור מהלך השיעור (90 דקות)**

השיעור הוא שיעור מעבדה, התלמידים מורגלים לפתוח סביבה תומכת למידה moodle ואת התוכנה להקלדה והרצת תכניות בשפת התכנות.

שלבי השיעור	פירוט	הערות
פתיחה	<p>- בעבר למדתם ליצור טיפוסים חדשים, כלומר בנייתם מחלקה שממנה ניתן ליצור עצמים. <u>שאלה</u>: הזכירו מי המחלקות שכתבתם? מה זה טיפוס חדש?</p> <p>- בעבר למדתם ליצור מערך מטיפוס בסיסי ולבצע אלגוריתמים מסוימים על המערך. <u>שאלה</u>: ציינו תבניות אלגוריתמיות שאתם מכירים ותבניות אלגוריתמיות שיישמתם על מבנה מערך. - היום נלמד לבנות מערך של עצמים.</p>	<p>לרענן ולהזכיר ידע קודם כדי לחבר למעבדה זו. אין להתעכב יותר מידי</p>
הסבר והדגמה ראשונית	<p>- לפתוח את הסביבה התכנותית ולהדגים בתוכנה איך יוצרים מערך של עצמים. 1. המורה או תלמיד מקליד: <code>Point [] arrPoints = new Point[3];</code> הסבר: יצרנו מערך בגודל 3 מטיפוס Point 2. המורה או התלמיד מקליד: <code>for (int index=0; index&lt;arrPoints.length; index++) System.out.println("point "+(index+1)+"-"+arrPoints[index]);</code> 3. שאלה: מה התקבל? מה קרה? 4. הסבר שלא מספיק ליצור את המערך בעזרת המילה השמורה new. חייבים לאתחל כל תא במערך לעצם מטיפוס המחלקה 5. שאלה: איך נאתחל כל תא במערך לעצם מטיפוס point? 6. נשמע את עצות התלמידים ונקליד דוגמה: <code>arrPoints[0] = new Point(1,1.5); arrPoints[1] = new Point(-1,0); arrPoints[2] = new Point(5,2.5);</code> 7. נעתיק את שורות ההדפסה של המערך שוב ונראה את הפלט. 8. נזכיר שהעובדה שיכולנו לראות את הפלט של הנקודות זה בזכות העובדה שלמחלקה Point יש פעולה toString</p>	<p>המחשב של המורה מחובר למקרן. כך שתלמידים רואים את מה שהמורה או תלמיד נבחר מקליד בתוכנה. במידה ואין מקרן, המוקה מבקשת מהתלמידים להקליד ושואלת מה התקבל. השיעור מתבצע כדיאלוג בין המורה לתלמידים תוך הדגמה וסיכומי ביניים.</p>
עבודה עצמית מול המחשב	<p>עבודה על פי דף עבודה: <b>מערך של עצמים מטיפוס Point</b> התלמידים עובדים על פי ההנחיות בדף העבודה, מקלידים, מריצים ובודקים את התוצרים. המורה עוברת בין התלמידים, מסייעת ועוזרת. התלמידים מסייעים אחד לשני. אם המורה נתקלת בבעיה שצצה בין יותר מ 2-3 תלמידים, היא עוצרת את השיעור ומסבירה לעל התלמידים.</p>	<p>דף העבודה מצורף בהמשך</p>
שיעורי בית	<p>לסיים את המעבדה ולהגיש ב moodle לבדיקה והערכת המורה. צורת ההגשה מסמך טקסט או וורד הכולל: קוד גיאווה, תיעוד לעל קטע תכנית או פעולה, דוגמאות של פלטים</p>	

## דף עבודה: מערך של עצמים מטיפוס Point

מטרת המעבדה: עבודה עם מערך של עצמים  
 יישום תבניות אלגוריתמיות כגון: מקסימום, מינימום, חיפוש ומיון על מערך של עצמים  
 כתובת פעולות static זימונם בפעולה הראשית  
 שימוש במחלקה המתמטית Math  
 טיפול בזוגות נתונים במערך

### תרגיל 1: בניית מערך של נקודות והדפסתו

(1) צור פרויקט חדש בשם **ArrayPoints** והעתק אליו את המחלקה Point שיצרת בשיעורים קודמים.

(2) צור מחלקה **ArrayPoints** הכוללת פעולה ראשית.

(3) בפעולה הראשית צור מערך בגודל 3 מטיפוס Point

```
Point [] arrPoints = new Point[3];
```

(4) הוסף פעולה להדפסת תוכן אברי המערך.

```
public static void print(Point[] arr)
{
    // טענת כניסה: הפעולה מקבלת מערך של נקודות
    // הפעולה מדפיסה את אברי המערך
}
}
```

(5) זמן את פעולה ההדפסה של מערך הנקודות בפעולה הראשית.  
 מה התקבל?

#### מסקנה:

כאשר יוצרים מערך של עצמים בכל תא במערך יש null .  
 כדי שכל תא במערך יכיל הפנייה לעצם מטיפוס המערך, יש ליצור את העצם ולהציב בתא במערך.

(6) הוסף פעולה אשר תקבל בפרמטר את מספר הנקודות, n, כלומר את גודל המערך.

הפעולה תקלוט מהמשתמש n זוגות של ערכים שמהווים שיעורי x,y של הנקודה.

הפעולה תחזיר מערך בגודל n עם נקודות.

```
public static Point[] build(int n)
```

(7) זמן בפעולה הראשית את פעולה הבנייה של מערך הנקודות ואת פעולת הדפסת מערך הנקודות

```
int n;
System.out.println("Enter number of Points");
n = in.nextInt();
Point[] arr = ArrayPoints.build(n);
ArrayPoints.print(arr);
```

**תרגיל 2:** מציאת הערך המקסימלי את הנקודות ,  
הדפסת הנקודות בעל שיעור  $y$  גדול ביותר וכמה נקודות כאלו יש.

(8) הוסף פעולה למציאת הערך הגבוה ביותר של שיעור  $y$  מבחן הנקודות

```
public static double maxValue(Point[] pArr)
{// טענת כניסה : מערך של נקודות
//טענת יציאה: הערך המקסימלי של גובה נקודה}
}
```

(9) הוסף שורות קוד להדפסת הנקודות בעלי שיעור  $y$  מקסימלי וכמה נקודות כאלו יש.  
חובה לזמן את הפעולה שהוספת בסעיף (8)

**תרגיל 3:** מציאת הנקודות שעל הישר  $x+y=10$

(10) הוסף פעולה או שורות קוד בפעולה הראשית להדפסת הנקודות אשר נמצאות על הישר  $x+y=10$  .  
אם כתבת פעולה זמן אותה הפעולה הראשית.

**תרגיל 4:** מרחקים של נקודות סמוכות

(11) כתוב פעולה או הוסף שורות קוד לפעולה הראשית אשר יבצעו:  
יצירת מערך של מספרים ממשיים אשר יכילו את המרחקים של נקודות סמוכות במערך .  
הדפס את המרחקים בין הנקודות.

מה גודל מערך המרחקים?  
איך חישבת מרחק בין שתי נקודות ?

בהצלחה, Evi

```
Array of Points:
(-2.0,10.0)
(5.0,5.0)
(-3.0,10.0)
(8.0,2.0)
(3.5,6.5)
maximum level is: 10.0
Points in height 10.0
(-2.0,10.0)
(-3.0,10.0)
There are 2 points in level 10.0
Points on line x+y=10
(5.0,5.0)
(8.0,2.0)
(3.5,6.5)
distances between successive points
distance between point 1 and point 2 is: 8.602325267042627
distance between point 2 and point 3 is: 9.433981132056603
distance between point 3 and point 4 is: 13.601470508735444
distance between point 4 and point 5 is: 6.363961030678928
```

דוגמה להרצה

**תרגילים נוספים:**

- (1) מיין את מערך הנקודות על פי שיעור  $x$  מקטן לגדול
- (2) כתוב פעולה המקבלת מערך של Point ממיון על פי שיעור  $x$ . במערך יש  $n$  נקודות ממוינות. גודל המערך הוא  $n+1$ . הפעולה מקבלת נקודה נוספת. הפעולה מוסיפה את הנקודה למערך כך שיישאר ממוין. לאחר הוספת הנקודה במערך יש  $n+1$  נקודות ממוינות.
- (3) כתוב פעולה אשר מקבלת מערך נקודות . הפעולה מחזירה 'אמת' אם כל הנקודות נמצאות על ישר מקביל לציר  $x$ , 'שקר' – אחרת. חשוב מה זה אומר שהנקודות נמצאות על ישר מקביל לציר  $x$  ?

## פתרון תרגילים 4 – 1

```

import java.util.Scanner;
/**
 * @author EVI
 * @version 8/1/14
 * (a) build Array of Points
 * (b) find maximum value of function described by these points
 * (c) prints all points with maximum y-coordinate
 * (d) search point in array with sum of coordinate 10
 * (e) create array of double with distances between points
 * (f) sort array by x-coordinate
 * (g) add point to sorted array by x coordinate
 * (i) check if all points are on a line parallel to x-axis
 */
public class ArrayPoints
{
    static Scanner in = new Scanner(System.in);

    /**
     * exercise 1
     * @param n: integer
     * build Array of n Points
     * @return Array of n Points
     */
    public static Point[] build(int n)
    {
        Point[] arr = new Point[n];
        double x,y;
        for (int k=0; k<n; k++)
        {
            System.out.println("Enter x and y coordinates");
            x = in.nextDouble();
            y = in.nextDouble();
            arr[k] = new Point(x,y);
        }
        return arr;
    }
    /**
     * exercise 1
     * prints array of PPoints
     * @param arr : Point[]
     */
}

```

```

public static void print(Point[] arr)
{
    System.out.println("Array of Points:");
    for (int k=0; k<arr.length; k++)
    {
        System.out.println(arr[k]);
    }
}
/**
 * exercise 2
 * @param pArr: Point[]
 * @return maximum value of y-coordinate
 */
public static double maxValue(Point[] pArr)
{
    double max = pArr[0].getY();
    for (int i=1; i<pArr.length; i++)
    {
        if (pArr[i].getY(>)max)
            max = pArr[i].getY();
    }
    return max;
}
/**
 * prints Points with y-coordinate = value
 * and how many of them exist
 * @param pArr : Point[]
 * @param value : y-coordinate
 */
public static void pointsInY(Point[] pArr, double value)
{
    int count = 0;
    System.out.println("Points in height "+value);
    for (int i=0; i<pArr.length; i++)
    {
        if (pArr[i].getY()==value)
        {
            System.out.println(pArr[i]);
            count++;
        }
    }
    System.out.println("There are " + count+ " points in level " + value);
}

```



```

}
/**
 * exercise 3
 * prints point on the line x+y=10
 * @param pArr: Point[]
 */
public static void isOnline(Point[] pArr)
{
    System.out.println("Points on line x+y=10");
    for (int i=0; i<pArr.length; i++)
    {
        if (pArr[i].getX()+pArr[i].getY() == 10)
            System.out.println(pArr[i]);
    }
}
/**
 * exercise 4
 * create array of distances between 2 following points
 * and prints it
 * @param pArr: Point[]
 */
public static void distances(Point[] pArr)
{
    double[] distArr = new double[pArr.length-1];
    double dx, dy, dist;
    for (int i=1; i<pArr.length; i++)
    {
        dx = pArr[i].getX()-pArr[i-1].getX();
        dy = pArr[i].getY()-pArr[i-1].getY();
        dist = Math.sqrt(Math.pow(dx, 2)+Math.pow(dy, 2));
        distArr[i-1] = dist;
    }
    System.out.println("distances between successive points");
    for (int k=0; k<distArr.length; k++)
    {
        System.out.println("distance between point "+(k+1)+" and point "+(k+2)
            +" is: " + distArr[k]);
    }
}
}

```

```

//== check ==
public static void main(String[] args)
{

    //Part A: create array of Points
    Point [] arrPoints = new Point[3];
    // part B: prints the array
    for (int index=0; index<arrPoints.length; index++)
        System.out.println("point "+(index+1)+": "+arrPoints[index]);
    // part C: initializes each cell in array
    arrPoints[0] = new Point(1,1.5);
    arrPoints[1] = new Point(-1,0);
    arrPoints[2] = new Point(5,2.5);
    // part D: prints again
    for (int index=0; index<arrPoints.length; index++)
        System.out.println("point "+(index+1)+": "+arrPoints[index]);

    //--- exercise 1
    int n;
    System.out.println("Enter number of Points");
    n = in.nextInt();
    Point[] arr = ArrayPoints.build(n);
    ArrayPoints.print(arr);

    //--- exercise 2
    double max = ArrayPoints.maxValue(arr);
    System.out.println("maximum level is: " + max);
    ArrayPoints.pointsInY(arr, max);

    //--- exercise 3
    ArrayPoints.isOnLine(arr);

    //--- exercise 4
    ArrayPoints.distances(arr);
    }// end main
}

```