

## מאגר מעבדות לשפות התכנות החדשות

ניתן להשתמש בחומרים לצורך הוראה בלבד.

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

ללא קבלת אישור מד"ר תמר פז.

המעבדה בקובץ זה מיועדת לתלמידים הלומדים מדעי המחשב בשפת התכנות ג'אווה והיא מותאמת לסביבת אקליפס.

המעבדה מיועדת לשיעורי המעבדה והיא מבוססת על שיטת ההוראה לפיה הלימוד של כל נושא חדש יפתח בהתנסות אישית במעבדה. לאחריה, יבוא דיון כיתתי, שבעקבותיו ייפתרו משימות שונות.

המעבדה מתרכזת בלימוד נושא מרכזי אחד:

### היכרות עם עצמים (יצירת עצמים וביצוע פעולות על עצמים)

במעבדה זו נעשה שימוש במחלקה Bucket. הרעיון למחלקה Bucket לקוח מתוך: המרכז להוראת המדעים, האוניברסיטה העברית, עיצוב תכנה מבוסס עצמים, 2005. אבל,

מחלקה Bucket במעבדות אלו איננה זהה למחלקה Bucket של האוניברסיטה העברית. מצורף קובץ מחלקה Bucket המתאים למעבדה הנוכחית.

## יצירת עצמים וביצוע פעולות על עצמים

## משימה 1 – חלק א'

פתחו מחלקה חדשה בתוך הפרויקט BucketProject והקלידו בה את המחלקה הבאה. שימרו, הריצו ורשמו את הפלט:

```
public class FirstProgram
{
    public static void main(String[] args)
    {
        Bucket b1;
        b1 = new Bucket (12);
        System.out.println("b1: "+b1.toString());
    }
}
```

המשתנים בהם עסקנו עד כה הכילו ערך מטיפוס נתונים.

הצהרנו עליהם כך: **שם המשתנה הטיפוס שהמשתנה יכול** למשל, `int number;` גם `b1` הוא משתנה וההצהרה עליו היא כמו ההצהרה על המשתנים בהם פגשנו עד כה. אבל, `b1` הוא משתנה שהטיפוס שהוא יכול יהיה הפניה לעצם מטיפוס `Bucket` משמעות ההוראה: `Bucket b1;` היא הצהרה על משתנה בשם `b1` שיכול להכיל הפניה לעצם מטיפוס `Bucket` (דלי). משמעות ההוראה: `b1 = new Bucket (12);` היא: **יצירת עצם** מטיפוס `Bucket` (דלי) והצבת הפנייה אליו בתוך המשתנה `b1`.

## משימה 1 – חלק ב'

אפשר לאחד את הגדרת המשתנה `b1 (Bucket b1);` עם יצירת הדלי החדש והצבת הפניה אליו בתוך `b1 (b1=new Bucket b1(12);)`.

לשם כך מחקו את שתי ההוראות הנ"ל וכתבו במקומן את ההוראה: `Bucket b1 = new Bucket (12);` תזכורת: יש משמעות לאותיות גדולות וקטנות!!

- שימרו את המחלקה.
- הריצו פעם נוספת וודאו כי הפלט לא השתנה.

```
public class FirstProgram
{
    public static void main(String[] args)
    {
        Bucket b1 = new Bucket (12);
        System.out.println("b1: "+b1.toString());
    }
}
```

משמעות ההוראה: **Bucket b1= new Bucket (12);** היא הגדרת משתנה בשם `b1`, יצירת עצם מטיפוס `Bucket` (דלי) והצבת הפנייה אליו בתוך המשתנה `b1`.

**משימה 1 – חלק ג'**

ההוראה `System.out.println("b1: "+b1.toString());` היא הוראה להדפסת המחרוזת "b1: " ובצמוד אליה, הדפסת הערך המוחזר מהפעולה `.toString()`. הערך המוחזר מפעולה זו כולל את הערכים של התכונות של הדלי (Bucket) שהפניה אליו נמצאת בתוך `b1`.  
 מהפלט של הפעולה אנו רואים כי לדלי (Bucket) יש שתי תכונות:  
 תכונה ראשונה: הקיבולת (`capacity`) של הדלי (במקרה שלנו, 12)  
 תכונה שנייה: הכמות הנוכחית (`current amount`) שהדלי מכיל (במקרה שלנו \_\_\_\_\_).

לעצם יש תכונות

**משימה 1 – חלק ד'**

הוסיפו לפעולה `main` את ההוראה:

`b1.addAmount(8);`

שימרו את הפעולה, הריצו והשלימו:

\_\_\_\_\_ הקיבולת של הדלי נשארה

\_\_\_\_\_ הכמות הנוכחית בדלי היא:

```
public class FirstProgram
{
    public static void main(String[] args)
    {
        Bucket b1 = new Bucket (12);
        b1.addAmount(8);
        System.out.println("b1: "+b1.toString());
    }
}
```

**משימה 1 – חלק ה'**

הוסיפו לפעולה את ההוראה: `b1.addAmount(3);`

שימרו את הפעולה, הריצו והשלימו:

```
public class FirstProgram
{
    public static void main(String[] args)
    {
        Bucket b1 = new Bucket (12);
        b1.addAmount(8);
        b1.addAmount(3);
        System.out.println("b1: "+b1.toString());
    }
}
```

משמעות ההוראה: `b1.addAmount (num);` היא: \_\_\_\_\_ `num`  
 לכמות הנוכחית (`current amount`) שהדלי מכיל.

משמעות ההוראה `b1.addAmount(3)`; היא הפעלת הפעולה `addAmount` על העצם שההפניה אליו נמצאת במשתנה `b1`. כדי שהפעולה `addAmount` תוכל לפעול, צריך לתת לה ערך. 3 הוא הערך הניתן לפעולה. במילים אחרות, `addAmount` היא פעולה, 3 הוא הערך האקטואלי שניתן לפרמטר שלה. הפעולות שהכרנו עד כה יכלו לקבל ערכים ויכלו להחזיר ערך. גם הפעולות הנוכחיות יכולות לקבל ערכים ויכולות להחזיר ערך. ובנוסף, הפעולות הנוכחיות גם פועלות על עצם כלשהו!

## משימה 2 – חלק א'

הקלידו את המחלקה הבאה, שימרו, הריצו, ורישמו מה היא מבצעת. אל תתארו את האלגוריתם, רק רישמו באופן כללי מה מבצעת הפעולה הראשית.

```
import java.util.Scanner;
public class SecondProg
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.println("enter capacity");
        double capacity = input.nextDouble();
        Bucket b1 = new Bucket (capacity);
        System.out.println ("enter amount to fill");
        double amount = input.nextDouble();
        b1.addAmount (amount);
        System.out.println("b1: "+b1.toString());
    }
}
```

## משימה 2 – חלק ב'

שנו את הפעולה כך שתקלוט מהמשתמש קיבולות של שני דליים, תיצור שני דליים בהתאם לקיבולות שנקלטו, תמלא אותם בכמויות כרצונכם, ותציג כפלט את התכונות והערכים של שני הדליים.

- שימרו את המחלקה, הריצו ובידקו שהתקבל הפלט המבוקש.
- בידקו גם מה קורה כאשר מנסים למלא את הדלי מעבר לקיבולת שלו.

## משימה 2 – חלק ג'

שנו את הפעולה כך שתמלא את הדלי בעל הקיבולת הגדולה יותר בכמות השווה לקיבולת של הדלי השני. ואת הדלי השני הפעולה תמלא בכמות השווה להפרש שבין שתי הקיבולות. למשל אם יקלטו הקיבולות 10 ו-8, הפעולה תיצור שני דליים. הראשון יהיה בעל קיבולת 10 וימולא בכמות של 8, והשני יהיה בעל קיבולת 8 וימולא בכמות של 2. לסיום, הפעולה תציג כפלט את התכונות והערכים של שני הדליים.

- שימרו, הריצו ובידקו שהתקבל הפלט המבוקש.

**משימה 3**

כתבו פעולה שתיצור 3 דליים (first, second, third) בעלי קיבולת 100 כל אחד. לאחר מכן, הפעולה תקלוט שלוש כמויות למילוי, ותמלא את first בכמות הגדולה ביותר, את second בכמות האמצעית בגדלה ואת third בכמות הקטנה ביותר. לסיום, הפעולה תציג כפלט את התכונות והערכים של שלושת הדליים. הוסיפו לפעולה מסננת קלט שתוודא כי כל הכמויות למילוי קטנות או שוות ל-100. במידה והתקבלה כמות למילוי הגדולה מ-100, יש לבקש מהמשתמש כמות אחרת למילוי.

יש להשתמש בפעולות Math.min ו-Math.max

- הקלידו, שימרו, הריצו ובידקו שהתקבל הפלט הרצוי.

**משימה 4**

כתבו פעולה שתיצור 4 דליים בעלי קיבולת 100 כל אחד, ותמלא אותם ב"סדרה חשבונית". הפעולה תקלוט מהמשתמש את הכמות למילוי של הדלי הראשון, ואת ההפרש בין הדליים. למשל אם הכמות למילוי שתתקבל עבור הדלי הראשון תהייה 10 וההפרש שיתקבל יהיה 8 אז בדלי הראשון הפעולה תמלא 10, בשני 18, בשלישי 26, וברביעי 34. הפעולה תציג כפלט את התכונות והערכים של הדליים. יש להוסיף לפעולה מסננת קלט שתוודא כי הכמות למילוי של הדלי הרביעי לא תעלה על הקיבולת שלו (100). במידה, הכמות למילוי תהייה גדולה מהקיבולת, יש לבקש מהמשתמש כמות אחרת למילוי.

- הקלידו, שימרו, הריצו ובידקו שהתקבל הפלט הרצוי.

**עצם מורכב מתכונות שמתארות את המצב של העצם, ומפעולות שניתן לבצע עליו.**

עד כה ראינו כי אפשר ליצור עצם מטיפוס דלי (Bucket) ואפשר להוסיף לו כמות רצויה באמצעות הפעולה \_\_\_\_\_ . בהמשך, נכיר פעולות נוספות שאפשר לבצע על עצם מטיפוס דלי.

**משימה 5 – חלק א'**

```
public class AddAmountRemoveAmount
{
    public static void main(String[] args)
    {
        Bucket b1= new Bucket (12);
        b1.addAmount(6);
        System.out.println("b1: "+b1.toString ());
        b1.removeAmount(4);
        System.out.println("b1: "+b1.toString ());
    }
}
```

- הקלידו, שימרו והריצו את המחלקה הבאה.
- השלימו:

משמעות ההוראה:  
 b1.removeAmount(num); היא:  
 num \_\_\_\_\_ מהכמות הנוכחית  
 (current amount) שהדלי מכיל.

**משימה 5 – חלק ב'**

שנו את הפעולה האחרונה כך שתוכלו לענות על השאלה: מה קורה אם הכמות לריקון גדולה יותר מהכמות הנוכחית בדלי?

הקלידו, שימרו, הריצו והשלימו:

אם הכמות לריקון גדולה יותר מהכמות הנוכחית בדלי אז \_\_\_\_\_

הפעולות addAmount ו- removeAmount פועלות על עצם מטיפוס Bucket (דלי). שתיהן צריכות לקבל ערך כדי לפעול. במשימה הבאה נכיר פעולות שלא צריכות לקבל ערך בכדי לפעול (פעולות שלא מקבלות אף פרמטר).

**משימה 6**

• הקלידו, שימרו והריצו את המחלקה הבאה.

```
public class AddAmountRemoveAmount2
{
    public static void main(String[] args)
    {
        Bucket b1= new Bucket (12);
        b1.addAmount(6);
        System.out.println(b1.toString ());
        if (b1.isEmpty())
            System.out.println("b1 is empty");
        else
            System.out.println("b1 is not empty");
        b1.emptyAll();
        if (b1.isEmpty())
        {
            System.out.println("now b1 is empty");
            System.out.println("b1: "+b1.toString ());
        }
    }
}
```

• השלימו:

הפעולה emptyAll לא מקבלת אף ערך

ו\_\_\_\_\_ (כן / לא) מחזירה ערך.

הפעולה מרוקנת את הכמות הנוכחית מהדלי עליו היא פועלת.

הפעולה isEmpty מקבלת ערך \_\_\_\_\_

ומחזירה ערך מטיפוס \_\_\_\_\_

הפעולה מחזירה את הערך true אם \_\_\_\_\_

\_\_\_\_\_

ומחזירה את הערך \_\_\_\_\_ אם \_\_\_\_\_

\_\_\_\_\_

## משימה 7

כתבו פעולה שתבצע:

1. תקלוט מהמשתמש קיבולת של דלי ותיצור דלי עם הקיבולת שנקלטה.
  2. תקלוט מהמשתמש כמות למילוי ותמלא את הדלי בכמות שנקלטה.
  3. תרוקן את הדלי "לאט לאט". כל פעם תרוקן ליטר אחד ותציג כפלט את מצב הדלי (כלומר, לאחר ריקון של כל ליטר, הפעולה תציג כפלט את תכונות הדלי ואת ערכיהן). התהליך ימשיך עד שהדלי יהיה ריק לגמרי.
- הקלידו, שימרו, הריצו ובידקו שהתקבל הפלט הרצוי.

ההפעלה של כל הפעולות הפועלות על עצם מטיפוס דלי היא אחידה: ***pe*** ***הפעולה.b1*** למשל, \_\_\_\_\_.

במידה והפעולה לא מחכה לערך נרשום: ***pe*** ***הפעולה.b1*** ( )

גם **toString** היא פעולה שפועלת על עצם מטיפוס דלי. וההפעלה שלה היא בדיוק כמו הפעלה של

הפעולות האחרות הפועלות על העצם דלי: ***pe*** ***הפעולה.b1*** ( )

הפעולה **toString** \_\_\_\_\_ (כנ /  $kf$ ) מקבלת ערך. הפעולה מחזירה מחרוזת הכוללת את השמות של התכונות של עצם מטיפוס דלי ואת הערכים שלהן עבור הדלי עליו היא מופעלת!

## משימה 8

פעולות נוספות שניתן להפעיל על עצם מטיפוס דלי הן: `getCurrentAmount` ו-`getCapacity`. שתי הפעולות הללו לא מקבלות שום ערך ושתיהן מחזירות מספר ממשי. כתבו פעולה כרצונכם שתבדוק מהו הערך המוחזר משתי הפעולות הללו. **רמז:** כיוון ששתי הפעולות מחזירות ערך, צריך לעשות משהו עם הערך המוחזר. אפשר להציב אותו במשתנה ממשי למשל כך: `double x = b1.getCapacity()`; (ואז כדי לדעת מהו הערך המוחזר צריך להדפיס את \_\_\_\_\_) ואפשר להדפיס ישירות את הערך המוחזר.

השלימו:

הפעולה `getCapacity` לא מקבלת שום ערך ומחזירה את \_\_\_\_\_

הפעולה `getCurrentAmount` לא מקבלת שום ערך ומחזירה את \_\_\_\_\_

הוראה ליצירת עצם חדש מטיפוס Bucket (דלי) והצבת הפנייה אליו בתוך משתנה

**Bucket b1 = new Bucket (12);** (הקובלת *fe הדלי*) *משתנה* **Bucket**

למשל, Bucket b1 = new Bucket (12);

פעולות הפועלות על עצם מטיפוס דלי:

פעולות מאחזרות: מחזירות את הערכים של התכונות של הדלי עליו הן פועלות:

**getCurrentAmount**, \_\_\_\_\_

פעולות נוספות: **addAmount**, **removeAmount**, **emptyAll**, \_\_\_\_\_,

**toString**

ההפעלה של פעולה: (צרך) *משתנה* *יש בו הפניה לצמט מטיפוס דלי*

יתכנו פעולות שלשם הפעלתן צריך לתת מספר ערכים ויתכנו פעולות שלא מחכות לשום ערך.

למשל, \_\_\_\_\_

כאשר פעולה מחזירה ערך (למשל, isEmpty) צריך לעשות משהו עם הערך המוחזר!

## משימה 9

כתבו פעולה שתקלוט מהמשתמש קיבולת של דלי ותיצור דלי עם הקיבולת שנקלטה.

יש להוסיף מסננת קלט שתדאג כי הקיבולת הנקלטת היא חיובית.

לאחר מכן, הפעולה תמלא את הדלי "לאט לאט". כל פעם תוסיף לדלי ליטר אחד ותציג כפלט את מצב

הדלי (כלומר, לאחר הוספה של כל ליטר, הפעולה תציג כפלט את התכונות של הדלי ואת ערכיהן).

התהליך ימשך עד שהדלי יהיה מלא. יש להשתמש בפעולה `getCurrentAmount`.

• הקלידו, שימרו, הריצו ובידקו שהתקבל הפלט הרצוי.

## משימה 10

כתבו פעולה שתקלוט מהמשתמש קיבולת של דלי (יש לדאוג שהקיבולת תהייה חיובית) ותיצור דלי עם

הקיבולת שנקלטה. לאחר מכן, הפעולה תקלוט מהמשתמש כמות למילוי (יש לדאוג שהכמות לא תהייה

שלילית), תוסיף לדלי את הכמות שנקלטה ותציג כפלט את מצב הדלי. התהליך ימשך עד שהדלי יהיה

מלא. יש להשתמש בפעולה `getCurrentAmount`

• הקלידו, שימרו, הריצו ובידקו שהתקבל הפלט הרצוי

לעצם יש תכונות, שניתן לשנות את ערכיהן במהלך ביצוע הפעולה. ויש פעולות שניתן לבצע

עליו.



**משימה 11**

כתבו פעולה שתבצע:

1. תקלוט מהמשתמש קיבולת של דלי ותיצור דלי עם הקיבולת שנקלטה.
  2. תשאל את המשתמש האם ברצונו להוסיף כמות לדלי או להחסיר ממנה כמות.
  3. תקלוט מהמשתמש כמות למילוי או לריקון.
  4. תוסיף או תרוקן את הכמות מהדלי בהתאם לתשובת המשתמש בסעיף 2.
  5. תחזור על סעיפים 2-4 עד שהדלי יהיה ריק או שהכמות הנוכחית תהייה שווה לקיבולת של הדלי.
  6. תציג כפלט את המצב של הדלי.
- הקלידו, שימרו, הריצו ובידקו שהתקבל הפלט הרצוי

פעולות שפועלות על עצם מטיפוס דלי:

דוגמה להפעלה	ביצוע / הערך המוחזר	הערך שמקבלת	הפעולה
<code>c = b1.getCapacity()</code>	מוחזר הערך של התכונה <code>capacity</code> של הדלי עליו היא פועלת.	לא מקבלת ערך	<code>getCapacity</code>
<code>c=b1.getCurrentAmount()</code>	מוחזר הערך של התכונה <code>currentAmount</code> של הדלי עליו היא פועלת.	לא מקבלת ערך	<code>getCurrentAmount</code>
<code>b1.addAmount(10)</code>	לא מוחזר ערך. הפעולה מוסיפה את הקלט לכמות הנוכחית בדלי עליו היא פועלת. הפעולה לא תוסיף לדלי מעבר לקיבולת שלו. הנחה: הקלט איננו שלילי.	מספר ממשי	<code>addAmount</code>
<code>b1.removeAmount(8)</code>	לא מוחזר ערך. הפעולה מרוקנת את הקלט מהכמות הנוכחית בדלי עליו היא פועלת. אם הקלט גדול מהכמות הנוכחית, הפעולה תרוקן את כל הדלי. הנחה: הקלט איננו שלילי.	מספר ממשי	<code>removeAmount</code>
<code>b1.emptyAll()</code>	לא מוחזר ערך. הפעולה מרוקנת את הדלי עליו היא פועלת.	לא מקבלת ערך	<code>emptyAll</code>
<code>if (b1.isEmpty()) System.out.print ("empty");</code>	מוחזר <code>true</code> אם הדלי ריק, <code>false</code> אחרת.	לא מקבלת ערך	<code>isEmpty</code>
<code>s=b1.toString()</code>	מוחזרת מחרוזת הכוללת את השמות של התכונות של עצם מטיפוס דלי ואת הערכים שלהם עבור הדלי עליו היא מופעלת.	לא מקבלת ערך	<code>toString</code>

## משימה 12 – חלק א'

```
public class Composition
{
    public static void main(String[] args)
    {
        Bucket b1= new Bucket (12);
        b1.addAmount(6);
        Bucket b2= new Bucket (b1.getCapacity());
        b2.addAmount(b1.getCurrentAmount());
        System.out.println("b1: "+b1.toString());
        System.out.println("b2: "+b2.toString());
    }
}
```

עקבו אחר הפעולה שמשמאל ורשמו כאן מה יהיה הפלט שלה:

---



---

• הקלידו את המחלקה.

• הריצו והשלימו:

ניתן לבצע הרכבה של פעולות. למשל,

משמעות הביטוי:

b2.addAmount(b1.getCurrentAmount());

היא: הפעלת הפעולה addAmount על הדלי

שההפניה אליו נמצאת במשתנה \_\_\_\_\_ . הערך שהפעולה תקבל, יהיה הערך המוחזר

מהפעולה \_\_\_\_\_ המופעלת על דלי שההפניה אליו נמצאת

במשתנה \_\_\_\_\_ .

## משימה 12 – חלק ב'

בצעו את השינויים הבאים:

1. שנו את הכמות למילוי של הדלי שההפניה

אליו נמצאת ב b2 , בחרו כמות כרצונכם כך

שתהייה שונה מהכמות שבדלי שההפניה

אליו נמצאת ב b1.

2. לאחר שתי הוראות ההדפסה, הוסיפו את

ההוראות הדרושות כך שהכמות הנוכחית

תוחלף בין שני הדליים והדפיסו בשנית את

התכונות וערכיהן של שני הדליים.

רמז: יש צורך בדלי נוסף.

• שימרו, הריצו ובידקו שהתקבל הפלט

המבוקש.

```
public class Composition
{
    public static void main(String[] args)
    {
        Bucket b1= new Bucket (12);
        b1.addAmount(6);
        Bucket b2= new Bucket (b1.getCapacity());
        b2.addAmount( כתבו כאן כמות כרצונכם );
        System.out.println("b1: "+b1.toString());
        System.out.println("b2: "+b2.toString());
        הוסיפו כאן את ההוראות הדרושות
    }
}
```