

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: **פעולות (שיטות)**

המעבדה נכתבה עבור תלמידים שלא מכירים עצמים ולכן כל הפעולות נכתבות עם `static`.
כדאי לבקש מהתלמידים להוסיף טענות כניסה וטענות יציאה.

פעולות (שיטות)

במעבדה זו נלמד להעשיר את השפה ולהוסיף לה פעולות כרצוננו.

```
import java.util.Scanner;
public class First
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        double n1 , n2;
        int result;
        System.out.println ("enter two numbers ");
        n1 = input.nextDouble();
        n2 = input.nextDouble();
        result = numPositive (n1,n2);
        System.out.println ("num positive= "+result );
    }
    private static int numPositive (double a , double b)
    {
        int count = 0;
        if (a>0 && b>0)
            count =2;
        else
            if (a>0 || b>0)
                count =1;
        return count;
    }
}
```

משימה 1 – חלק א'

הקלידו והריצו את המחלקה הבאה. כאשר תתבקשו, הקלידו שני מספרים כרצונכם.

השלימו:

- הפעולה הראשית (main) קולטת מהמשתמש _____ מספרים ומציגה כפלט את מספר המספרים החיוביים (0 או 1 או 2).
- במחלקה מוגדרת פעולה נוספת בשם **numPositive**. הפעולה **numPositive** מקבלת שני נתונים מטיפוס מספר ממשי ומחזירה את _____

- הפעולה **numPositive (double a , double b)** היא החתימה של הפעולה **numPositive**.
- החלק **(double a , double b)** מסמן כי בעת ההפעלה של הפעולה יש לתת לה _____ ערכים מטיפוס **double**.
- המילה **int** לפני שם הפעולה **numPositive** מסמנת כי הפעולה תחזיר ערך מטיפוס _____

- המילה **private** מציינת כי הפעולה **numPositive** היא פעולה פרטית של המחלקה **First** ואי אפשר לזמן אותה ממחלקות אחרות.
- במשמעות המילה **static** נדון מאוחר יותר.
- ההוראה **return count** היא הוראת החזרה. תפקידה לדאוג שהפעולה תחזיר ערך.
- בכל פעולה חייבים לתת הוראת החזרה:

; הצרך המוחזר על-ידי הפעולה return

```
import java.util.Scanner;
public class First
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        double n1 , n2;
        int result;
        System.out.println ("enter two numbers ");
        n1 = input.nextDouble();
        n2 = input.nextDouble();
        result = numPositive (n1,n2);
        System.out.println ("num positive= "+result );
    }
    private static int numPositive (double a , double b)
    {
        int count = 0;
        if (a>0 && b>0)
            count=2;
        else
            if (a>0 || b>0)
                count=1;
        return count;
    }
}
```

משימה 1 – חלק ב'

a ו-b הם הנתונים עליהם פועלת הפעולה. נתונים כאלה נקראים **פרמטרים פורמליים** משתמשים בהם רק בפעולה numPositive והפעולה הראשית main לא מכירה אותם!!

כאשר קוראים לפעולה (מזמנים את הפעולה) משתמשים בנתונים שהפעולה הקוראת (הפעולה המזמנת) מכירה.

הנתונים הללו נקראים **פרמטרים אקטואליים** (כי הם הערכים האקטואליים כרגע!). במקרה שלנו הערך האקטואלי של הפרמטר a הוא הערך של המשתנה _____, והערך האקטואלי של הפרמטר _____ הוא הערך של המשתנה n2

הקריאה (הזימון) לפעולה היא כמו הזימון לפעולות הקיימות בשפה: (נתונים) שם הפעולה
ובמקרה שלנו, הזימון לפעולה הוא באמצעות ההוראה: _____

נבדוק כעת אילו ערכים יכולים לשמש כפרמטרים אקטואליים. בעמודה הימנית של הטבלה הבאה מופיעות 5 הוראות. הקלידו בנפרד כל אחת מהן במקום השורה המודגשת בפעולה הראשית, הריצו את המחלקה ומלאו את הטבלה. שימו לב: **חלק מההוראות הן שגויות!**

הסבר שלכם	פלט של הפעולה	ערכים אקטואליים		הוראה לביצוע	
		a	B		
				result = numPositive (n1 , 100);	1
				result = numPositive (n2 , n1);	2
				result = numPositive (-75 , 100);	3
				result = numPositive (count, n2);	4
				result = numPositive (a , b);	5

מסקנה: (מחקו את המיותר)**פרמטר אקטואלי יכול להיות:**

משתנה של הפעולה הקוראת (הפעולה המזמנת).
 משתנה שהוגדר בתוך הפעולה המזומנת (כמו count).
 ערך שאינו משתנה (למשל 999).

```
import java.util.Scanner;
public class First
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        double n1 , n2;
        System.out.println ("enter two numbers ");
        n1 = input.nextDouble();
        n2 = input.nextDouble();
        while (numPositive (n1,n2)> 0)
        {
            System.out.println ("enter two numbers ");
            n1 = input.nextDouble();
            n2 = input.nextDouble();
            System.out.println ("num positive =" +
                numPositive (n1,n2));
        }
    }
    private static int numPositive (double a , double b)
    {
        int count = 0;
        if (a>0 && b>0)
            count=2;
        else
            if (a>0 || b>0)
                count=1;
        return count;
    }
}
```

משימה 2 – חלק א'

מכיוון שפעולה מחזירה ערך, כאשר מזמנים פעולה (קוראים לפעולה) צריך להגיד מה לעשות עם הערך המוחזר. **שנו את הפעולה הראשית בלבד** כך שתקבל המחלקה הבאה.

- שימרו, הריצו והשלימו:

ניתן לבצע עם הערך המוחזר על-ידי

פעולה, כל מה שניתן לבצע עם משתנה

מהטיפוס שמוחזר על-ידי הפעולה.

למשל, אפשר להציב אותו במשתנה אחר,

אפשר _____ אותו,

ואפשר לבדוק את ערכו באמצעות

משימה 2 – חלק ב'

הקלידו בנפרד כל אחת משתי ההוראות הבאות במקום הקטע המודגש.

שימרו, הריצו ומלאו את הטבלה. שימו לב: **שתי ההוראות שגויות!**

הוראה לביצוע	הוראה שגויה – הסבר שלכם	
return numPositive (n1 , n2);		1
numPositive (n1,n2) = result;		2

משימה 3

```

import java.util.Scanner;
public class Order
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        double n1 , n2 , n3;
        int count = 0;
        for (int i=1; _____; _____)
        {
            System.out.println ("enter three numbers");
            n1 = _____;
            n2 = input.nextDouble();
            n3 = input.nextDouble();
            if (between (n1 , n2 , n3))
                count++;
        }
        System.out.println ("there was " + count + " order three");
    }

    private static boolean between (double a , double b , double c)
    {
        return (a<b _____ b<c);
    }
}

```

לפניכם שלד של מחלקה שהפעולה הראשית בה קולטת 10 שלשות של מספרים ומונה את מספר השלשות "המסודרות". שלשה מסודרת היא שלשה שבה המספר הראשון הוא הקטן ביותר והמספר האחרון הוא הגדול ביותר. למשל, השלשה 3 7 9 היא שלשה מסודרת, והשלשה 9 10 8 איננה שלשה מסודרת.

- השלימו את המחלקה.
- הריצו ובדקו שהתקבל הפלט הרצוי.
- השלימו: הפעולה between

מקבלת שלושה ערכים מטיפוס _____, הפעולה מחזירה ערך מטיפוס _____

הערך שפעולה מחזירה יכול להיות מכל אחד מהטיפוסים שהכרנו בשפה: `int` , `double` , `char` , `boolean`

במחלקה האחרונה, הפעולה between מזומנת (נקראת) _____ פעמים.

בכל המקרים, הזימון לפעולה between נעשה באמצעות ההוראה _____.

השם של הפרמטר הפורמלי הראשון הוא `a` ונכנס אליו הערך של המשתנה _____.

השם של הפרמטר הפורמלי השני הוא _____ ונכנס אליו הערך של המשתנה _____.

השם של הפרמטר הפורמלי השלישי הוא _____ ונכנס אליו הערך של המשתנה _____.

• הסבירו מדוע ההוראה: `if (between (n1 , n2 , n3)) count++;` היא הוראה תקינה _____

משימה 4

א. הפעולה multi אמורה לקבל שני מספרים חיוביים ושלמים ולהחזיר את תוצאת הכפל שלהם בלי להשתמש בפעולת הכפל (*). בפעולה נפלה שגיאה אחת. תקנו אותה.

```
private static int multi (int a , int b)
{
    int i , result;
    result = 0;
    for (i= 1; i<=a; i++)
        result = result + a;
    return result;
}
```

i ו- result הם משתנים מקומיים רק של הפעולה multi

אפשר להצהיר בפעולה על משתנים שיהיו רק שלה.

משתנים אלו אינם מוכרים על-ידי הפעולה הראשית והם נקראים משתנים מקומיים של הפעולה!

ב. כתבו פעולה ראשית שקולטת מספרים חיוביים ושלמים (לפחות מספר אחד), ומציגה כפלט את מכפלתם. קליטת הנתונים תפסק עם קליטת המספר 0 (מספר זה לא יוכפל במספרים האחרים). למשל אם ייקלטו המספרים (משמאל לימין): 0 1 6 2 3 , הפעולה תציג כפלט 36.

השתמשו בפעולה multi

משימה 5

```
private static char kod (int num1 , int num2)
{
    char result;
    if (num1 > num2)
        result = 'p';
    else
        if (num1 < num2)
            _____ ;
        else
            _____ ;
    return result;
}
```

א. הפעולה kod מקבלת שני מספרים שלמים ומחזירה תו. אם המספר הראשון שהתקבל גדול מהשני, הפעולה תחזיר את התו p, אם הראשון קטן מהשני, הפעולה תחזיר n, ואם הם שווים, יוחזר התו z.

ב. כתבו פעולה ראשית שמגרילה 8 זוגות של מספרים בתחום שבין -100 לבין 100. הפעולה תציג כפלט שורה של תווים בהתאם למספרים שהוגרלו. למשל אם המספרים שהוגרלו הם (משמאל לימין):

12 12 -99 9 -8 9 -9 0 90 100 -8 7 -56 56 0 12 אז הפעולה תציג כפלט: pznnpnpz

זיכרו: יש חשיבות לסדר הופעת הפרמטרים האקטואליים.

משימה 6

כתבו פעולה ראשית שפולטת עבור כל אחד מהמספרים 1-5 את העצרת שלו.

הפלט יראה כך:

```
1 factorial is 1
2 factorial is 2
3 factorial is 6
4 factorial is 24
5 factorial is 120
```

כתבו תחילה פעולה fact שמקבלת מספר ומחזירה את העצרת שלו.

משימה 7

כתבו פעולה ראשית שקולטת מספרים חיוביים שלמים ופולטת עבור כל מספר את סכום ספרותיו ואת מספר ספרותיו. קליטת הנתונים תפסק כאשר ייקלט מספר שלילי.

כתבו תחילה שתי פעולות:

sumDigit שמקבלת מספר ומחזירה את סכום ספרותיו.

numDigit שמקבלת מספר ומחזירה את מספר ספרותיו.

משימה 8

א. כתבו פעולה שמקבלת מספר שלם חיובי. הפעולה תחזיר את הערך true אם יש במספר לפחות ספרה זוגית אחת, ותחזיר false אם אין במספר אף ספרה זוגית.

ב. כתבו פעולה ראשית שתקלוט 10 מספרים שלמים וחיוביים. הפעולה תודיע בכמה מהמספרים יש לפחות ספרה זוגית אחת. יש להשתמש בפעולה שכתבתם בסעיף א'.

משימה 9

בכל המחלקות שכתבנו עד כה, היתה רק הוראת זימון אחת לפעולה (שבחלק מהמקרים, התבצעה מספר פעמים) והפעולה שימשה לחלוקת המשימה לתת משימות. בבעיה הבאה נגדיר פעולה ונזמן אותה ממקומות שונים בפעולה הראשית.

תלמידי שכבת י' יצאו לטיול שנתי. נכתוב פעולה שתקלוט את מספר התלמידים בשכבה ותציג כפלט את מספר האוטובוסים שיש להזמין (בכל אוטובוס ניתן להסיע לכל היותר 50 תלמידים), ואת מספר החדרים

```
private static int numBus (int num)
{
    int order ;
    if (num % 50 == 0 )
        order = _____
    else
        order = (num / 50) + 1;
    return order;
}
```

שיש להזמין באכסניה (בכל חדר יכולים ללון 6 תלמידים לכל היותר).

א. הפעולה numBus מקבלת את מספר

התלמידים ומחזירה את מספר

האוטובוסים הדרוש.

• השלימו את הפעולה.

```
private static int numRooms (int num)
{
}
}
```

ב. הפעולה numRooms תקבל את מספר התלמידים ותחזיר את מספר החדרים הדרוש. השלימו את הפעולה.

ג. הפעולות numBus ו- numRooms שונות זו מזו בגורם אחד (מלבד בשמן כמובן). מהו? _____

```
private static int numItem (int num , int capacity)
{
  int order ;
  if (num % capacity == _____ )
    order = (num / capacity);
  else
    order = _____ ;
  return order;
}
```

ד. הפעולה numItem היא פעולה כללית שמקבלת מספר התלמידים וקיבולת (50 במקרה של אוטובוסים, ו-6 במקרה של חדרים). הפעולה מחזירה את מספר הפריטים הדרוש. השלימו את הפעולה.

ה. כתבו כעת את המחלקה המלאה (השתמשו בפעולה numItem). שימרו, הריצו ובדקו כי התקבלו התוצאות הרצויות.

ו. שנו את המחלקה כך שבנוסף למספר התלמידים בשכבה, המחלקה גם תקלוט מהמשתמש את מספר המקומות באוטובוס ואת מספר המיטות בחדר. הפעם הפעולה הראשית צריכה לבצע את המשימות הבאות (את האלגוריתם הבא):

1. לקלוט את מספר התלמידים.
2. לקלוט את מספר המקומות באוטובוס.
3. לקלוט _____
4. לחשב את מספר האוטובוסים הדרוש.
5. לחשב את מספר החדרים הדרוש.
6. להציג כפלט את _____ ואת מספר החדרים.

שימרו, הריצו ובדקו כי התקבלו התוצאות הרצויות.

משימה 10

כתבו מחלקה שקולטת רביעיות של מספרים ומבצעת את הפעולות הבאות:
 אם המספר הראשון ברביעייה הוא 1 אז המחלקה תציג כפלט את הממוצע של המספר השני והמספר השלישי.
 אם המספר הראשון ברביעייה הוא 2 אז המחלקה תציג כפלט את הממוצע של המספר השני והמספר הרביעי.
 אם המספר הראשון ברביעייה הוא 3 אז המחלקה תציג כפלט את הממוצע של המספר השלישי והמספר הרביעי.
 קליטת הנתונים תיפסק כאשר המספר הראשון ברביעייה לא יהיה 1 או 2 או 3

יש להשתמש בפעולה אחת לפחות בנוסף לפעולה הראשית.

משימה 11 – חלק א'

שימו לב לזימון המיוחד לפעולה func2

הקלידו, הריצו את המחלקה ורשמו מה היא מבצעת:

```
public class Stam
{
    public static void main(String[] args)
    {
        System.out.println (func2 ( ));
    }

    private static double func1 (double a)
    {
        return (a+1);
    }

    private static double func2 ()
    {
        double x;
        x = Math.random ();
        return (x + func1 (x));
    }
}
```

מסקנות (מחקו את המיותר):

1. פעולה יכולה / לא יכולה לזמן פעולה אחרת שהוגדרה במחלקה.
2. פעולה חייבת / לא חייבת לקבל פרמטרים.
3. (השלימו): כאשר פעולה לא מקבלת פרמטרים:

- א. בחתימה של הפעולה, במקום רשימת הפרמטרים, רושמים _____
- ב. בעת הזימון לפעולה, צריך להוסיף _____ לאחר שם הפעולה.

השלימו: משמעות ההוראה: `Math.random ()`; היא זימון של הפעולה _____.

פעולה זו כן / לא (מחקו את המיותר) מקבלת ערכים.

ומשמעות ההוראה _____ הוא זימון של הפעולה `nextInt`.

גם פעולה זו לא מקבלת אף ערך. הפעולה מחזירה מספר שלם שהמשתמש הקליד.

משימה 11 – חלק ב'

- שנו את המחלקה כך שהפעולה func2 תופיע לפני הפעולה func1.
- הריצו את המחלקה.

```
public class Stam
{
    public static void main(String[] args)
    {
        System.out.println (func2 ());
    }

    private static double func2 ()
    {
        double x;
        x = Math.random ();
        return (x + func1 (x));
    }

    private static double func1 (double a)
    {
        return (a+1);
    }
}
```

מסקנה (השלימו):

חשיבות לסדר הגדרת הפעולות.

כלומר, פעולה יכולה לזמן גם פעולות

שמוגדרות

משימה 12

- המחלקה הבאה קולטת מספר טבעי (num) ומספר ממשי נוסף (extra), ופולטת בשורה אחת num פעמים את המספר הנוסף extra.
- הקלידו והריצו את המחלקה.
- כאשר תתבקשו, הקלידו מספר טבעי ומספר נוסף כרצונכם.

במחלקה מוגדרת פעולה בשם **line**.

השלימו:

- לפעולה line יש שני פרמטרים:

count מטיפוס _____

מטיפוס _____

- לפעולה line יש משתנה מקומי אחד:

מטיפוס _____

```
import java.util.Scanner;
public class OneLine
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int num;
        double extra;
        System.out.println ("enter number");
        num = input.nextInt();
        System.out.println ("enter extra number");
        extra = input.nextDouble();
        line (num, extra);
    }
    private static void line (int count, double n)
    {
        int i ;
        for (i=1; i<=count; i++)
            System.out.print (n+ " ");
        System.out.println ("");
    }
}
```

שימרו את המחלקה. נזדקק לה במשימה הבאה

במה שונה הפעולה line מהפעולות שהכרנו עד כה (שימו לב לחתימה של הפעולה ולזימון של הפעולה)

מסקנות

1. (מחקו את המיותר): פעולה **חייבת** / **לא חייבת** להחזיר ערך.

2. (השלימו): כאשר פעולה לא מחזירה ערך:

א. בחתימה של הפעולה, במקום הערך המוחזר, רושמים: _____

ב. בגוף הפעולה אין הוראת _____ ערך.

ג. בעת הזימון לפעולה, לא צריך לעשות כלום עם ערך מוחזר.

פעולה כזו שלא מחזירה ערך נקראת **פרוצדורה**

3. **גם main היא פעולה!!!**

main היא פעולה ראשית שמופעלת "באופן אוטומטי" עם הפעלת המחלקה.

החתימה `public static void main(String[] args)` מציינת כי:

main היא פעולה ש **לא** / **כן** מחזירה ערך. (מחקו את המיותר)

הערך שהפעולה main מקבלת הוא `args` מטיפוס `String[]` ועל כך נרחיב בהמשך.

המלה `public` מציינת כי main היא פעולה פומבית שניתן לפנות אליה גם ממחלקות אחרות.

משימה 13

```
import java.util.Scanner;
public class OneLine
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int num;
        double extra;
        System.out.println ("enter number");
        num = input.nextInt();
        System.out.println ("enter extra number");
        extra = input.nextDouble();
        line (num, extra);
    }
    private static void line (int count, double num)
    {
        int i ;
        for (i=1; i<=count; i++)
            System.out.print (num + " ");
        System.out.println ("");
    }
}
```

נשנה כעת את הפעולה line ממשימה 12 כך שהשם של הפרמטר יהיה num ולא n שימו לב, אין שום שינוי אחר במחלקה! ובפרט גם במחלקה main יש משתנה בשם num. מה תהיה תגובת המחשב כאשר נריץ עכשיו את המחלקה?

• הריצו את המחלקה.

• בדקו מהו הפלט

• הסבירו מדוע התקבל פלט זה?

משימה 14 – חלק א'

• מה יהיה הפלט של המחלקה הבאה?

```
public class Sod
{
    public static void main(String[] args)
    {
        a(35);
    }
    private static void a (int number)
    {
        System.out.println ("a starts with "+ number);
        b (1000);
        b (number);
        System.out.println ("a ends with "+ number);
    }
    private static void b (int number)
    {
        System.out.println ("in b number = "+ number);
    }
}
```

• הקלידו, הריצו ובדקו מהו הפלט

• הסבירו מדוע התקבל פלט זה?

```
public class Sod
{
    public static void main(String[] args)
    {
        a(35);
    }
    private static void a (int number)
    {
        System.out.println ("a starts with "+ number);
        b (1000);
        b (number);
        System.out.println ("a ends with "+ number);
    }
    private static void b (int number)
    {
        number = 80;
        System.out.println ("in b number = "+ number);
    }
}
```

משימה 14 – חלק ב'

• הוסיפו לפעולה b את השורה
number=80;

• אין שינוי בשאר ההוראות במחלקה!
מה יהיה הפלט של המחלקה עכשיו?

• הקלידו, הריצו ובדקו מהו הפלט

• הסבירו מדוע התקבל פלט זה?

```
public class Sod
{
    public static void main(String[] args)
    {
        a(35);
    }
    private static void a (int number)
    {
        System.out.println ("a starts with "+ number);
        b (number);
        b (mispar);
        System.out.println ("a ends with "+ number);
    }
    private static void b (int mispar)
    {
        System.out.println ("in b mispar = "+ mispar);
    }
}
```

משימה 15

• מה יהיה הפלט של המחלקה הבאה?

• הקלידו ושימרו.

מופיעה הודעת שגיאה!

הסבירו מדוע התקבלה ההודעה.

משימה 16

```
import java.util.Scanner;
public class avarege
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        double a , b, m=0;
        System.out.println ("enter two numbers");
        a = input.nextDouble();
        b = input.nextDouble();
        average (a, b , m);
        System.out.println ("the average of "+a+" and "+b+" is "+m);
    }
    private static void average (double a, double b, double m)
    {
        m = (a+b)/ 2;
    }
}
```

- המחלקה הבאה אמורה לקלוט שני מספרים, לחשב את הממוצע ולפלוט הודעה מתאימה. המחלקה לא מבצעת את הנדרש. הסבירו מה הבעיה.

- תקנו את המחלקה.

- שימרו, הריצו ובידקו כי התקבל הפלט המבוקש.

משימה 17 – חלק א'

- כתבו פעולה ראשית שקולטת שני מספרים שלמים בתחום 1-9. הפעולה תציג כפלט את ההדפסה

88888	999	הבאה:
8888	99	
888	9	עבור הקלט 9, 3 יודפס:
88		
8		

הערות:

- כתבו מסוננת קלט לבדיקה כי המספרים הנקלטים הם בתחום הרצוי.
- כתבו תחילה פעולה שדואגת להדפסת שורה אחת (העזרו במשימה 12).

רמז: הביטוי לשינוי מונה הלולאה צריך להיות --

משימה 17 – חלק ב'

- הוסיפו לפעולה את ההוראות הדרושות כך שיוצגו כפלט הצירים הבאים:

88888	עבור הקלט 8, 5 יודפס:	999	עבור הקלט 9, 3 יודפס:
8888		99	
888		9	
88		9	
8		99	
8		999	
88			
888			
8888			
88888			

משימה 18

```

1 import java.util.Scanner;
2 public class Grades
3 {
4     static int highNum , highGrade;
5     public static void main(String[] args)
6     {
7         Scanner input = new Scanner(System.in);
8         int num , grade ;
9         System.out.println ("enter num");
10        highNum = input.nextInt();
11        System.out.println ("enter grade");
12        highGrade = input.nextInt();
13        for (int i = 2; i<11; i++)
14        {
15            System.out.println ("enter num");
16            num = input.nextInt();
17            System.out.println ("enter grade");
18            grade= input.nextInt();
19            high (num , grade);
20        }
21        System.out.println (highNum + " got the highest grade");
22    }
23    private static void high (int num , int grade)
24    {
25        if (grade > highGrade)
26        {
27            highGrade = grade;
28            highNum = num;
29        }
30    }
31 }

```

המחלקה הבאה קולטת 10 מספרי תלמיד וציונים. המחלקה פולטת את המספר של התלמיד בעל הציון הגבוה ביותר (מספור השורות הוא רק לצורך הדיון ואינו חלק מהמחלקה).

המשתנים highNum ו- highGrade מוגדרים בתחילת המחלקה. משתנים כאלה לא שייכים לפעולה מסוימת, אלא שייכים למחלקה כולה!

המילה **static** בשורה 4 מסמנת כי המשתנים המוגדרים בשורה זו שייכים למחלקה כולה.

השלימו:

▪ בשורות 9-12 קולטים את הנתונים של התלמיד הראשון.

את מספר התלמיד מציבים ב-

ואת הציון _____

מציבים ב- _____

▪ בלולאה שמתחילה בשורה 13 ,

קולטים את הנתונים של _____

ומזמנים את הפעולה high

שתפקידה _____

• מדוע highGrade ו- highNum לא יכולים להיות משתנים של הפעולה main וגם לא יכולים להיות משתנים של הפעולה high ?

• הקלידו, שימרו והריצו. כאשר תתבקשו, הקלידו מספרי תלמיד וציונים כרצונכם.

משתנה גלובלי הוא משתנה שמוגדר בתחילת המחלקה. כל הפעולות מכירות אותו, יכולות

להשתמש בו ויכולות גם לשנות אותו.

הגדרה של משתנה גלובלי: static משתנה טיפוס המשתנה

משימה 19

המחלקה שבעמוד הבא מציגה למשתמש 10 תרגילי חשבון. לכל תרגיל כזה, המחלקה מציגה את התרגיל, קולטת תשובה מהמשתמש, ומגיבה בהתאם לתשובה. לאחר הצגת 10 תרגילים, המחלקה שואלת את המשתמש האם ברצונו להמשיך ולתרגל. במידה ותשובת המשתמש חיובית, מוצגים בפניו 10 תרגילים נוספים. א. פיתחו את המחלקה MathEx, הריצו ובדקו שהיא אכן מבצעת את הנדרש. ב. השלימו:

- תפקיד הפעולה ten_exe הוא: _____
- הפעולה choose מחזירה את _____.
- הפעולה calculate מגרילה שני מספרים בתחום שבין _____ לבין _____, ומכניסה אותם למשתנים n_1 ו- n_2 . הפעולה מחזירה את תוצאת הפעולה operation עם שני המספרים שהוגרלו.
- הפעולה one_exe מקבלת שני פרמטרים: התוצאה הרצויה וסוג הפעולה המבוקשת. הפעולה מבצעת:
 - א. מציבה במשתנה _____ +, -, או * בהתאם לסוג הפעולה המבוקש.
 - ב. מדפיסה את התרגיל.
 - ג. _____.
 - ד. בודקת האם היא נכונה.
 - ה. מגיבה בהתאם.
- ג. מדוע המשתנים n_1 ו- n_2 מוכרחים להיות משתנים גלובליים? _____

ד. גם לפעולה הראשית (main) וגם לפעולה one_exe יש משתנה בשם answer. המשתנים הם מטיפוסים שונים ובכל זאת המחלקה תקינה. מדוע? _____

ה. אילו שינויים יש לערוך במחלקה אם נרצה להפוך את choose לפעולה שלא מחזירה ערך? _____

1. בצעו במחלקה את השינויים הבאים.
 - לאחר כל שינוי, הריצו ובדקו את המחלקה.
 - כדי למנוע עבודה עם מספרים שליליים, המחלקה תדאג שתמיד n_1 יהיה גדול מ- n_2 .
 - במידה והמשתמש נתן תשובה שגויה לתרגיל שהוצג בפניו, המחלקה תאפשר לו לתת תשובה נוספת. אם טעה גם בפעם השניה, המחלקה תציג כפלט הודעה מתאימה ותמשיך הלאה.
 - לפני כל סדרה של עשרה תרגילים, המחלקה תשאל את המשתמש באיזה תחום הוא מעוניין שיהיו המספרים המשתתפים בתרגילים, ותפעל בהתאם לבקשת המשתמש.
 - לשם ביצוע שינוי זה, יש להוסיף עוד פעולה!


```

import java.util.Scanner;
public class MathEx
{
static int n1 , n2;
static Scanner input=new Scanner (System.in);

    public static void main(String[] args)
    {
        boolean answer = true;
        while (answer )
        {
            ten_exe( );
            System.out.println ("do you want more
                exercises? Type true or false");
            answer = input.nextBoolean();
        }
        System.out.println("good bye");
    }
    private static void ten_exe ()
    {
        int result , i, operation;
        for (i=1; i<=10; i++)
        {
            operation = choose();
            result = calculate (operation);
            one_exe (result , operation);
        }
    }
    private static int choose ()
    {
        int operation;
        do {
            System.out.println ("choose operation: 1 for '+' 2 for '-' 3 for '*' ");
            operation = input.nextInt();
        }
        while (operation != 1 && operation != 2 && operation != 3);
        return operation;
    }
    private static int calculate (int operation)
    {
        int result;
        n1 = (int)(Math.random ()*10); n2 = (int)(Math.random ()*10);
        switch (operation)
        { case 1 : result = n1 + n2; break;
          case 2 : result = n1-n2; break;
          default : result = n1* n2; break;
        }
        return result;
    }
    private static void one_exe (int result, int operation)
    {
        int answer;
        char op;
        switch (operation)
        { case 1 : op = '+'; break;
          case 2 : op = '-'; break;
          default : op = '*'; break;
        }
        System.out.print (" "+n1+op+n2+"=?");
        answer= input.nextInt();

```

כאשר רוצים לקלוט נתונים מהמשתמש בשתי פעולות או יותר, יש שתי אפשרויות להשתמש במחלקה Scanner

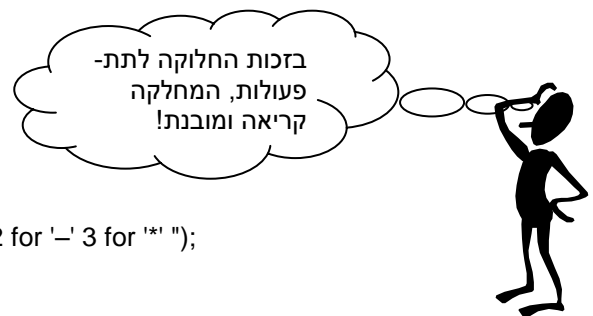
אפשרות ראשונה: לכתוב בתחילת כל אחת מהפעולות הרלוונטיות, את השורה:

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

אפשרות שנייה: לכתוב פעם אחת בתחילת המחלקה לפני כל הפעולות (אותה שורה בתוספת המילה `:(static`

```
static Scanner input=new Scanner (System.in);
```

במחלקה זו, בחרנו באפשרות השנייה.



```

        if (answer == result)
            System.out.println ("excellent");
        else
            System.out.println ("wrong");
    }
}

```

משימה 20

- א. כתבו פעולה שמקבלת מספר שלם וחיובי, ומחזירה true אם הספרה 5 מופיעה במספר.
- ב. כתבו פעולה ראשית שתקלוט מספר טבעי, תזמן עבורו את הפעולה מסעיף א ותדפיס את הערך שהוחזר מהפעולה.

משימה 21

- א. כתבו פעולה שתגריל 50 מספרים בתחום 1-100. לאחר הגרלת כל מספר, הפעולה תבקש מהמשתמש לנחש 10 פעמים מהו המספר שהוגרל. לכל מספר שהוגרל, הפעולה תבדוק ותודיע כמה מעשרת הניחושים היו בתחום של המספר פלוס מינוס 5. למשל אם הוגרל המספר 90. הפעולה תבקש מהמשתמש 10 ניחושים, תמנה ותודיע כמה מהניחושים היו בתום 85-95. לסיום, הפעולה תודיע עבור כמה מהמספרים המוגרלים, המשתמש ניחש נכון פלוס מינוס 5, לפחות 4 ניחושים.
- ב. כתבו פעולה ראשית שתזמן את הפעולה מסעיף א.

משימה 22

- "מספר קסם" הוא מספר שלם וחיובי המורכב ממספר זוגי של ספרות שבו הספרה הראשונה זהה לספרה השניה, הספרה השלישית זהה לספרה הרביעית, הספרה החמישית זהה לספרה השישית, וכן הלאה. למשל, המספרים 22446688 ו-3399 הם מספרי קסם, והמספרים 2244887 ו-2446 אינם מספרי קסם.
- א. כתבו פעולה magic שתקבל מספר שלם וחיובי. הפעולה תחזיר את הערך true אם המספר שהתקבל הוא "מספר קסם", ותחזיר את הערך false אם המספר איננו "מספר קסם".
- ב. כתבו פעולה ראשית שתקלוט מספר טבעי, תזמן עבורו את הפעולה מסעיף א ותדפיס את הערך שהוחזר מהפעולה.

משימה 23

- בתחרות לטיפוס על קירות משתתפים 30 אנשים. המנצח בתחרות הוא המשתתף שטיפס לגובה הגבוה ביותר. במידה ושני משתתפים טיפסו לאותו הגובה, המנצח הוא זה שנעזר במספר הקטן ביותר של ווים.
- א. כתבו פעולה שלא תקבל אף ערך. הפעולה תקלוט את הגובה (במטרים) אליו הגיע כל אחד מהמשתתפים, ואת מספר הווים בהם נעזר. הפעולה תציג כפלט את הגובה אליו הגיע המשתתף שניצח בתחרות ואת מספר הווים בהם הוא נעזר.
- קליטת הנתונים תלווה במסננת קלט מתאימה.
- ב. כתבו פעולה ראשית שתזמן את הפעולה מסעיף א.