

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: **מחרוזות**

מחרוזות

משימה 1 – חלק א'

```
import java.util.Scanner;
public class TestString
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        String st1, name;
        st1 = "hello ";
        System.out.println ("enter your name ");
        name= input.next();
        System.out.println (st1+name);
    }
}
```

מחלקה מוכנה נוספת היא המחלקה String. עצם מטיפוס String הוא מחרוזת (או רצף של תווים).

- הקלידו, שימרו והריצו את המחלקה הבאה. כאשר תתבקשו הקלידו את שמכם.
- השלימו:

♦ הפעולה מגדירה שני משתנים שיכולים להכיל

הפניה לעצם מטיפוס _____

♦ st1 מקבל את ערכו באמצעות הוראת השמה:

_____ = "hello ";

♦ name מקבל את ערכו באמצעות הוראת הקלט:

- עצם מטיפוס String הוא מחרוזת (או רצף של תווים).
- בכל המחלקות שהכרנו עד כה, המבנה של ההוראה ליצירת עצם חדש והצבת הפניה אליו בתוך משתנה הוא: (פראמטריט) `שם המחלקה new שם המשתנה שם המחלקה`
- המחלקה String יוצאת דופן מבחינה זאת. String היא המחלקה היחידה בה לא משתמשים במילה השמורה new על מנת ליצור עצם!
- הוראה ליצירת עצם מטיפוס String והצבת הפניה אליו בתוך משתנה:

באמצעות השמה: `String שם המשתנה = "רצף תווים" שם המשתנה`

או באמצעות קלט: `String שם המשתנה = input.next();`

כמובן שניתן לפצל לשתי הוראות:

1. הגדרת משתנה שיכול להכיל הפניה לעצם מטיפוס String: `String שם המשתנה;`

למשל, `String st1;`

2. הצבת הפניה לעצם מטיפוס String בתוך המשתנה

באמצעות הוראת השמה: `שם המשתנה = "רצף תווים" שם המשתנה`

או באמצעות הוראת קלט: `שם המשתנה = input.next ();`

- המחלקה String **מיובאת באופן אוטומטי** ואין צורך לייבא אותה כדי להשתמש בה.
- כותבים String עם S גדולה כי String זו מחלקה (ושם של מחלקה מתחיל באות גדולה) ולא טיפוס נתונים פשוט כמו int או char.

משימה 1 – חלק ב'

הריצו את הפעולה פעם נוספת. הפעם, כשתתבקשו להקליד את שמכם, הקלידו את שמכם ואת שם

משפחתכם עם רווח ביניהם ורשמו את פלט של הפעולה: _____

הפעולה `next()` מחכה לרצף של תווים שהמשתמש יקליד. רצף התווים מסתיים ב**תו enter** (מעבר שורה) או ב**תו רווח** או ב**תו Tab**. לכן כאשר הקלדנו רווח, נכנסו למשתנה רק התווים שהוקלדו לפני תו ה _____

משימה 1 – חלק ג'

מחקו כעת את ההוראה `input.next()` ורשמו במקומה את ההוראה `input.nextLine()` הריצו את הפעולה פעם נוספת. גם הפעם, כשתתבקשו להקליד את שמכם, הקלידו את שמכם ואת שם

משפחתכם עם רווח ביניהם. רשמו את פלט של הפעולה: _____

- עצם מטיפוס String הוא _____
- פעולה לקליטת ערך למחרוזת:
 - א. אפשרות ראשונה (כאשר המחרוזת מסתיימת ב `enter` או ברווח או ב `Tab`):
 $se =$ _____ *המחרוזת*
 - ב. אפשרות שניה (כאשר המחרוזת עשויה להכיל רווחים):
 $se =$ _____ *המחרוזת*

משימה 2 – חלק א'

לפניכם שתי פעולות מממשק המחלקה String

החתימה של הפעולה	תיאור הפעולה	דוגמאות name="gil"
boolean equals (String otherString)	מקבלת כפרמטר עצם מטיפוס String. מחזירה true אם הערך של המחרוזת הנוכחית שווה לערך של הפרמטר. ומחזירה false אחרת.	ההוראה name.equals ("tal") תחזיר false ההוראה name.equals ("gil") תחזיר true
int compareTo (String otherString)	מקבלת כפרמטר עצם מטיפוס String. מחזירה מספר שלילי אם המחרוזת הנוכחית מופיעה לפני הפרמטר על-פי הסדר המילוני, מספר חיובי אם המחרוזת הנוכחית מופיעה אחרי המחרוזת על-פי הסדר המילוני, 0 אם המחרוזות זהות.	ההוראה name.compareTo ("tal") תחזיר מספר שלילי ההוראה name.compareTo ("gil") תחזיר 0

```
import java.util.Scanner;
public class Tal
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int count = _____;
        System.out.print ("enter name");
        String name = input.next();
        while (! name.equals ("abc") )
        {
            if (name.equals ("_____") )
                count ++;
            System.out.print ("enter name");
            name = input.next();
        }
        System.out.println (_____);
    }
}
```

לפניכם שלד של פעולה ראשית שקולטת את השמות של כל תלמידי בית הספר, מונה ומציגה כפלט את מספר התלמידים והתלמידות ששםם tal. קליטת הנתונים מסתיימת עם קליטת השם "abc".

- השלימו את הפעולה.
- הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.
- השלימו:
- משמעות ההוראה:
 - while (! name.equals ("abc")) היא כל זמן ש name אינו שווה ל _____
 - הפעלה של פעולה המוגדרת במחלקה String היא כמו הפעלה של פעולות המוגדרות במחלקות האחרות. כלומר:
 - (צרכיכם) *פעולת הפעולה. השתנה למשל,* _____
 - הביטוי: name.equals ("abc") הוא הפעלת הפעולה equals על העצם שההפניה אליו נמצאת

במשתנה _____, והמחרוזת "abc" משמשת כפרמטר לפעולה.

משימה 2 – חלק ב'

- שנו את הפעולה האחרונה כך שלא יעשה בה כלל שימוש בפעולה equals .
- רמז: השתמשו בפעולה compareTo .
- שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 3

- כתבו פעולה שקולטת שמות של תלמידים וציונים. הפעולה תציג כפלט את השם של התלמיד בעל הציון הגבוה ביותר. קליטת הנתונים תסתיים עם קליטת השם "noName" (אין לקלוט ציון עבור שם זה).
- שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 4

לפניכם שתי פעולות נוספות מממשק המחלקה String

החתימה של הפעולה	תיאור הפעולה	דוגמאות name="ortal"tal"
int indexOf (String otherString) int indexOf (char ch)	מקבלת כפרמטר עצם מטיפוס String או משתנה מטיפוס char. מחזירה את המיקום הראשון בו מופיע הפרמטר בתוך המחרוזת הנוכחית. אם הפרמטר אינו מופיע, יוחזר -1. התו הראשון נחשב כתו מספר 0, התו השני כתו מספר 1 וכן הלאה.	ההוראה name.indexOf ("tal") תחזיר 2 ההוראה name.indexOf ("gal") תחזיר -1 ההוראה name.indexOf ('a') תחזיר 3
int lastIndexOf (String otherString) int lastIndexOf (char ch)	מקבלת כפרמטר עצם מטיפוס String או משתנה מטיפוס char. מחזירה את המיקום האחרון בו מופיע הפרמטר בתוך המחרוזת הנוכחית. אם הפרמטר אינו מופיע, יוחזר -1.	ההוראה name.lastIndexOf("tal") תחזיר 5 ההוראה name.lastIndexOf("gal") תחזיר -1 ההוראה name.lastIndexOf ('a') תחזיר 6

- כתבו פעולה ראשית שקולטת את השמות של כל תלמידי בית הספר, ומציגה כפלט את השמות של התלמידים והתלמידות שיש בשםם tal (... , talor , lital , revival , ortal). קליטת הנתונים תפסק עם קליטת השם "abc".
- הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 5

לפניכם שתי פעולות נוספות מממשק המחלקה String

החתימה של הפעולה	תיאור הפעולה	דוגמאות st="I love you"
int length()	לא מקבלת כל כפרמטר. מחזירה את מספר התווים במחרוזת הנוכחית.	ההוראה st.length() תחזיר 10
char charAt (int index)	מקבלת כפרמטר מספר טבעי. מחזירה את התו שנמצא במקום מספר index במחרוזת הנוכחית.	ההוראה st.charAt(4) תחזיר v

```
import java.util.Scanner;
public class InputChar {
    public static void main(String[] args)
    {
        char tav ;
        Scanner input = new Scanner(System.in);
        for (int i=1; i<=5;i++)
        {
            System.out.println ("enter char");
            tav = input.next().charAt(0);
            System.out.println ("char = " + tav);
        }
    }
}
```

- פיתחו מחלקה חדשה והקלידו בה את הפעולה הראשית הבאה.
- שימרו, והריצו. בכל פעם שתבקשו הקלידו תו יחיד.

.....

הפעולה **input.next().charAt(0);** מאפשרת לקלוט תו בודד.

הפעולה input.next() היא הוראה לקליטת מחרוזת והפעולה charAt(0) מחזירה את תו מספר _____ (התו הראשון) מהמחרוזת שנקלטה.

.....

- מה יקרה לדעתכם אם במקום תו בודד תקלידו מחרוזת ? _____

- הריצו פעם נוספת ובידקו את תשובתכם.

משימה 6

```
import java.util.Scanner;
public class HowMany {
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.println ("enter sentence");
        String sen = input.nextLine();
        scan (sen);
    }
    private static void scan (String st)
    {
        char tav;
        int len = sen.length();
        for (int i=0; i<len;i++)
        {
            tav = sen.charAt (i);
            System.out.println ("char number " + i + " is " + tav);
        }
    }
}
```

- פיתחו מחלקה חדשה והקלידו בה את המחלקה הבאה.
- שימרו והריצו עם קלטים שונים.
- השלימו: הפעולה scan מקבלת מחרוזת ומציגה כפלט (מדפיסה)

סריקה של מחרוזת נעשית באמצעות לולאת מונה. מספר הפעמים שהלולאה מתבצעת הוא כמספר התווים

ב. _____.

בכל "סיבוב" של הלולאה מתייחסים לתו מספר i. התו הראשון במחרוזת נחשב כתו מספר _____,

והתו האחרון נחשב כתו מספר _____.

מכיוון שעצם מטיפוס מחרוזת הוא רצף של תווים (char), בעת סריקת המחרוזת מכניסים כל תו למשתנה

מטיפוס _____.

משימה 7

א. כתבו פעולה שתקבל משפט (מחרוזת) ותו נוסף. הפעולה תחזיר את מספר הפעמים שהתו הנוסף

מופיע במשפט. למשל עבור המשפט we learn java והתו a, הפעולה תחזיר את המספר 3.

ב. כתבו פעולה ראשית שתקלוט מחרוזת ותו נוסף. הפעולה תזמן את הפעולה מסעיף א' ותציג כפלט,

בליווי הודעה מתאימה, את הערך המוחזר על-ידה.

- הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 8

- א. כתבו פעולה שתקבל מחרוזת המייצגת משפט שבו בין שתי מילים יש רווח יחיד. הפעולה תציג כפלט את ראשי התיבות של המשפט (כלומר, את האות הראשונה של כל מילה). למשל אם הפעולה תקבל את המשפט: we learn java , אז היא תציג כפלט: wlj
- ב. כתבו פעולה ראשית שתקלוט מחרוזת משפט שבו בין שתי מילים יש רווח יחיד. הפעולה תזמן את הפעולה מסעיף א'.
- הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 9

- א. כתבו פעולה שתקבל מחרוזת המייצגת משפט שבו בין שתי מילים יתכנו מספר רווחים. הפעולה תציג כפלט כל מילה בשורה נפרדת. לא יודפסו שורות ריקות! הפעולה גם תמנה ותחזיר את מספר המילים שבמשפט.
- ב. כתבו פעולה ראשית שתקלוט מחרוזת המייצגת משפט שבו בין שתי מילים ייתכנו מספר רווחים. הפעולה תזמן את הפעולה מסעיף א' ותציג כפלט, בליווי הודעה מתאימה, את הערך המוחזר על-ידיה.
- הקלידו, שימרו, הריצו עם מספר קלטים ובדקו שהתקבל הפלט המבוקש.

```
import java.util.Scanner;
public class TestString{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.println ("Enter sentence");
        String s1 = _____;
        String newSt = oneSpace (s1);
        System.out.println ("St with one space= "+ newst);
    }
    private static String oneSpace (String st)
    {
        String newSn="";
        int len = st.length();
        for ( _____ )
        {
            if (st.charAt (i) != ' ')
                newSn= newSn+st.charAt(i);
            else if (st.charAt(i+1) != ' ')
                newSn= newSn+st.charAt(i);
        }
        return (newSn);
    }
}
```

משימה 10

- לפניכם שלד של מחלקה עם שתי פעולות:
- א. פעולה oneSpace שמקבלת מחרוזת המייצגת משפט שבו בין שתי מילים יתכנו מספר רווחים. הפעולה **בונה ומחזירה מחרוזת חדשה** זהה למחרוזת שהתקבלה פרט לכך שבין כל שתי מילים יש רווח יחיד.
- ב. פעולה ראשית שקולטת מחרוזת, מזמנת את הפעולה oneSpace ומדפיסה את הערך המוחזר על-ידיה.
- השלימו את המחלקה.
 - הקלידו, שימרו, הריצו עם מספר קלטים ובדקו שהתקבל הפלט המבוקש.

• פעולת השרשור + מאפשרת לבנות מחרוזת.

למעשה, השתמשנו כבר בשתי הזדמנויות שונות ב + כדי לבנות מחרוזת.

מקרה ראשון: _____

מקרה שני: כשהגדרנו את המחלקות Date ו- Bucket השתמשנו ב + כדי ליצור מחרוזת שתוחזר על-ידי

הפעולה toString.

• ההוראה newSn = ""; מציבה במחרוזת _____ ערך של מחרוזת ריקה.

משימה 11

א. כתבו פעולה שתקבל מחרוזת, **תבנה ותחזיר** מחרוזת חדשה שבה לאחר כל תו (חוץ מהתו רווח) מופיע התו * . למשל אם הפעולה תקבל את המחרוזת good day , היא תבנה ותחזיר את המחרוזת:

`g*o*o*d* d*a*y*`

ב. כתבו פעולה ראשית שתקלוט מחרוזת, תזמן את הפעולה מסעיף א' ותדפיס, בליווי הודעה מתאימה, את הערך המוחזר על-ידה.

• הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 12

א. כתבו פעולה שתקבל שתי מחרוזות, **תבנה ותחזיר** מחרוזת חדשה המכילה את החיתוך שלהן (התווים שמופיעים בשתי המחרוזות). למשל, אם הפעולה תקבל את המחרוזות "abcdef" ו-

"xbad4", היא תבנה ותחזיר את המחרוזת: "abd".

הנחה: אין חזרה בכל אחת מהמחרוזות. כלומר, אף תו לא יופיע יותר מפעם אחת באותה המחרוזת.

ב. כתבו פעולה ראשית שתקלוט שתי מחרוזות, תזמן את הפעולה מסעיף א' ותדפיס, בליווי הודעה מתאימה, את הערך המוחזר על-ידה.

• הקלידו, שימרו, הריצו עם מספר קלטים ובדקו שהתקבל הפלט המבוקש.

משימה 13

א. כתבו פעולה שתקבל שתי מחרוזות. הפעולה **תבנה ותחזיר** מחרוזת חדשה זהה למחרוזת הראשונה שהתקבלה עם התוספות הבאות: כל תו יופיע פעמיים, ולאחר מכן תופיע המחרוזת השניה. למשל אם

הפעולה תקבל את המחרוזות "abc" ו-"12", היא תבנה ותציג כפלט את המחרוזת: aa12bb12cc12

ב. כתבו פעולה ראשית שתקלוט שתי מחרוזות, תזמן את הפעולה מסעיף א' ותדפיס, בליווי הודעה מתאימה, את הערך המוחזר על-ידה.

• הקלידו, שימרו, הריצו עם מספר קלטים ובדקו שהתקבל הפלט המבוקש.

משימה 14

- א. כתבו פעולה שתקבל מחרוזת באורך לא ידוע. הפעולה **תבנה ותחזיר** מחרוזת שבה כל תו מהמחרוזת שנקלטה מופיע פעם אחת בדיוק. למשל, עבור הקלט "abeer saad" הפעולה תחזיר את המחרוזת "aber sd" או את המחרוזת "ber sad".
- ב. כתבו פעולה ראשית שתקלוט מחרוזת, תזמן את הפעולה מסעיף א' ותדפיס, בליווי הודעה מתאימה, את הערך המוחזר על-ידה.
- הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 15

- א. כתבו פעולה שתקבל שתי מחרוזות. שתיהן באורך לא ידוע. הפעולה תחזיר את הערך true אם כל אחד מהתווים של המחרוזת הקצרה מופיע במחרוזת הארוכה לפחות פעמיים. בכל מקרה אחר, הפעולה תחזיר false.
- למשל, עבור המחרוזות "abcaaxb" ו-"ab", הפעולה תחזיר את הערך true, ועבור המחרוזות "tp" ו-"tappa", הפעולה תחזיר את הערך false.
- ב. כתבו פעולה ראשית שתקלוט שתי מחרוזות, תזמן את הפעולה מסעיף א' ותודיע האם כל אחד מהתווים של המחרוזת הקצרה מופיע במחרוזת הארוכה לפחות פעמיים.
- הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 16

- א. כתבו פעולה שתקבל שתי מחרוזות. הפעולה תחזיר את מספר המופעים של המחרוזת הקצרה יותר במחרוזת הארוכה יותר. למשל אם הפעולה תקבל את המחרוזות: "abcdabcd", "ab", היא תחזיר את המספר 2.
- ב. כתבו פעולה ראשית שתקלוט שתי מחרוזות, תזמן את הפעולה מסעיף א' ותדפיס, בליווי הודעה מתאימה, את הערך המוחזר על-ידה.
- הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 17

במחלקות שהכרנו עד עכשיו הגדרנו פעולות שמשנות את הערכים של התכונות של העצם הנוכחי. String היא מחלקה יוצאת דופן גם בכך שלא ניתן לשנות ערך של עצם מטיפוס String. לכן, אף אחת מהפעולות במחלקה String אינה משנה את העצם עליו היא פועלת! אבל מוגדרות במחלקה String מספר פעולות שמחזירות עצם מטיפוס String.

החתימה של הפעולה	תיאור הפעולה	דוגמא st=" We Learn Java "
String replace (char oldChar, char newChar)	מחזירה מחרוזת חדשה זהה למחרוזת הנוכחית עם השינוי הבא: בכל מקום שבמחרוזת הנוכחית הופיע oldChar, יופיע במחרוזת החדשה newChar .	ההוראה st.replace ('a','e') תבנה ותחזיר את המחרוזת: " We Leern Jeve "
String replace (String oldSt, String newSt)	מחזירה מחרוזת חדשה זהה למחרוזת הנוכחית עם השינוי הבא: בכל מקום שבמחרוזת הנוכחית הופיע oldSt, יופיע במחרוזת החדשה newSt	ההוראה st.replace ("ea","123") תבנה ותחזיר את המחרוזת: " We L123rn Java "
String toLowerCase()	מחזירה מחרוזת חדשה זהה למחרוזת הנוכחית עם השינוי הבא: כל התווים נכתבים באותיות קטנות.	ההוראה st.toLowerCase() תבנה ותחזיר את המחרוזת: " we learn java "
String toUpperCase()	מחזירה מחרוזת חדשה זהה למחרוזת הנוכחית עם השינוי הבא: כל התווים נכתבים באותיות גדולות.	ההוראה st.toUpperCase () תבנה ותחזיר את המחרוזת: " WE LEARN JAVA "
String trim()	מחזירה מחרוזת חדשה זהה למחרוזת הנוכחית ללא תווי רווח בהתחלת המחרוזת ובסופה.	ההוראה st.trim() תבנה ותחזיר את המחרוזת: "We Learn Java"

הפעולות toUpperCase , toLowerCase שימושיות בקלט של מחרוזת.
נשתמש בהן כדי לאפשר למשתמש להקליד אותיות קטנות / גדולות כאוות נפשו. למשל,
String name = (input.next()).toUpperCase();

א. כתבו פעולה שתקבל מחרוזת המייצגת משפט שבו בין שתי מילים יתכנו מספר רווחים, ויתכנו גם רווחים בתחילת המשפט ובסופו. הפעולה תבנה ותחזיר מחרוזת חדשה שבה אין רווחים בתחילת המשפט ובסופו,

ובין שתי מילים עוקבות יופיע התו 8. למשל אם ייקלט המשפט: we love computer science
הפעולה תבנה ותחזיר את המחרוזת: we8love8computer8science

במז: לאחר קליטת המחרוזת, יש ליצור מחרוזת חדשה הזזה למחרוזת הנקלטת אך ללא רווחים בתחילתה ובסופה. אפשר לעשות זאת באמצעות הפעולה trim. כלומר: sen=sen.trim();

ב. כתבו פעולה ראשית שתקלוט מחרוזת, תזמן את הפעולה מסעיף א' ותדפיס, בליווי הודעה מתאימה, את הערך המוחזר על-ידה.

- הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

ההוראה sen=sen.trim() מפעילה את הפעולה trim על העצם שהפניה אליו נמצאת במשתנה sen ומציבה הפניה לעצם המוחזר מהפעולה במשתנה sen.
שימו לב ש"אבד הקשר" עם העצם המקורי.

משימה 18

- א. כתבו פעולה שתקלוט עבור כל אחד מתלמידי הכיתה מחרוזת אחת הכוללת את שמו ואת שם משפחתו (הרווח היחיד במחרוזת הנקלטת יהיה בין השם הפרטי לבין שם המשפחה). קליטת הנתונים תפסק עם קליטת השם "abc". עבור כל תלמיד הפעולה תבנה ותציג כפלט שתי מחרוזות. הראשונה תכלול את השם הפרטי של התלמיד, והשנייה תכלול את שם המשפחה שלו. אם השם הפרטי מתחיל באותיות a-s, השם הפרטי יודפס באותיות גדולות ושם המשפחה יודפס באותיות קטנות. ואם השם הפרטי מתחיל באותיות p-z, השם הפרטי יודפס באותיות קטנות ושם המשפחה יודפס באותיות גדולות.
- למשל, עבור הקלט "Ortal Avraham" הפעולה תציג כפלט ORTAL avraham ועבור הקלט Tomer Avraham הפעולה תציג כפלט tomER AVRAHAM
- ב. כתבו פעולה ראשית שתזמן את הפעולה מסעיף א.
- הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 19

- ראינו שג'אווה מאפשרת להגדיר מספר פעולות בעלות אותו שם בתנאי שהן שונות זו מזו ברשימת הפרמטרים (במספר הפרמטרים / או בטיפוסים שלהם). גם במחלקה String מוגדרות שתי פעולות בעלות אותו שם: substring ששונות זו מזו במספר הפרמטרים.

החתימה של הפעולה	תיאור הפעולה	דוגמה st="We Learn Java"
String substring (int first, int last)	מחזירה מחרוזת חדשה הזוה לקטע המחרוזת הנוכחית המתחיל בתו שנמצא במיקום first ומסתיים בתו שנמצא לפני המיקום last	ההוראה st.substring (3, 7) תבנה ותחזיר את המחרוזת: "Lear"
String substring (int first)	מחזירה מחרוזת חדשה הזוה לקטע המחרוזת הנוכחית החל מהתו שנמצא במיקום first ועד לסוף המחרוזת.	ההוראה st.substring (3) תבנה ותחזיר את המחרוזת: "Learn Java"

s קטנה

- א. כתבו פעולה שתקבל שתי מחרוזות. הפעולה תחזיר את מספר המופעים של המחרוזת הקצרה יותר במחרוזת הארוכה יותר. למשל אם הפעולה תקבל את המחרוזות: "abcdabcd", "ab" היא תחזיר את המספר 2, ואם הפעולה תקבל את המחרוזות: "aaaa", "aa" היא תחזיר 3.

יש להשתמש בפעולה substring

- ב. כתבו פעולה ראשית שתקלוט שתי מחרוזות, תזמן את הפעולה מסעיף א' ותדפיס, בליווי הודעה מתאימה, את הערך המוחזר על-ידיה.
- הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 20

- א. כתבו פעולה שתקבל מחרוזת באורך 2 לפחות. הפעולה תחזיר את הערך true אם קיים במחרוזת רצף שאינו כולל את התו הראשון ואת התו האחרון, בו מופיע התו הראשון של המחרוזת ולאחריו התו האחרון. במידה ולא קיים רצף כזה, הפעולה תחזיר false.
- למשל, עבור המחרוזת abagbeg הפעולה תחזיר true, ועבור המחרוזת gomana הפעולה תחזיר false.
- א. כתבו פעולה ראשית שתקלוט מחרוזת, תזמן את הפעולה מסעיף א', ותדפיס הודעה אם קיים או שלא קיים במחרוזת רצף (שאינו כולל את התו הראשון ואת התו האחרון) בו מופיע התו הראשון של המחרוזת ולאחריו התו האחרון.
- הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 21

- א. כתבו פעולה שתקבל שתי מחרוזות. הפעולה תחזיר את הערך true אם קיים במחרוזת הראשונה רצף הזהה למחרוזת השניה. במידה ולא קיים רצף כזה, הפעולה תחזיר false.
- למשל, עבור המחרוזות "libi" ו-"bi" הפעולה תחזיר true, ועבור המחרוזות "tom" ו-"tm" הפעולה תחזיר false.
- ב. כתבו פעולה ראשית שתקלוט שתי מחרוזות, תזמן את הפעולה מסעיף א', ותדפיס הודעה אם קיים או שלא קיים במחרוזת הראשונה רצף הזהה למחרוזת השניה.
- הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 22 – חלק א'

- פלינדרום הוא מילה שהאות הראשונה והאות האחרונה שלה זהות, וביניהן נמצא פלינדרום קצר יותר. במקרה של מספר אותיות זוגי, הפלינדרום הבסיסי, מכיל שתי אותיות זהות. במקרה של מספר אותיות אי זוגי, הפלינדרום הבסיסי, מכיל אות כלשהי.
- דוגמאות: אבא , אמא , סוס , ילד כותב בתוך דלי , אבבא
- לפניכם שלד של פעולה רקורסיבית שמקבלת מחרוזת st. הפעולה מחזירה true אם המחרוזת מהווה פלינדרום, ומחזירה false אחרת.

• השלימו את הפעולה.

```
private static boolean pali (String st)
{
    int len = st.length();
    if (len==1 || len== 0)
        return _____;
    else
        if (st.charAt (0) != _____)
            return false;
        else
            return pali (st.substring(1,len-1));
}
```

משימה 22 – חלק ב'

כתבו פעולה ראשית שתקלוט מהמשתמש מחרוזת. הפעולה תודיע האם המחרוזת מהווה או שאינה מהווה פלינדרום.

- הקלידו, שימרו, הריצו ובדקו שהתקבל הפלט המבוקש.

משימה 23

פלינדרום-כוכב הוא פלינדרום שמכיל מספר אי-זוגי של תווים, והתו האמצעי הוא * . דוגמאות: abc^*cba , $yu^*val^*lav^*uy$,

כתבו פעולה ראשית שתקלוט מחרוזת ותדפיס את האורך של הפלינדרום-כוכב האורך ביותר שבה.

למשל, אם תקלט המחרוזת $as^*sagab^*bac^*cab^*ba$ הפעולה תדפיס 13, ואם תקלט המחרוזת g^*a הפעולה תדפיס 1.

הנחה: הקלט תקין.

לפתרון המשימה כתבו תחילה שתי פעולות:

1. פעולה `paliLength` שתקבל מחרוזת ומיקום של * במחרוזת. הפעולה תחזיר את האורך של הפלינדרום-כוכב האורך ביותר שה- * הוא האמצע שלו. למשל עבור המחרוזת abc^*ab^*bac והמיקום 6, הפעולה תחזיר 5.

2. פעולה `maxStar` שתקבל מחרוזת, ותחזיר את האורך של הפלינדרום-כוכב האורך ביותר שבה.