

פרק 9 – המחלקה מחרוזת (String)

הצגת הפרק

מטרת הפרק

הכרת המחלקה מחרוזת (String), ואופן עיבוד מחרוזות:

- הקצאת מקום בזיכרון ואתחולו באמצעות הפעולה `new` לצורך הגדרת משתנה מטיפוס מחרוזת. נושא זה מוצג בתחילת הפרק כחזרה לנלמד בפרקים קודמים.
- הצגת קלט, פלט ופעולת שרשור על עצם מטיפוס מחרוזת.
- הצגת תכונות ופעולות המחלקה `String`, אופן השימוש בסימון הנקודה, וכן, הצגת מיקום התווים במחרוזת.

מושגים

- מחלקה ועצם
- מחרוזת
- הקצאת שטח זיכרון
- אתחול עצם
- פעולות
- תכונות
- שרשור

דגשים ודידקטיקה

- פרק זה הינו הזדמנות לרענן את הבנת התלמידים בתפיסת המושגים מחלקה ועצם. הזדמנות זו חשובה לקראת פרק 11 בו יידרשו התלמידים להבין לעומק את המושגים מחלקה ועצם ואת ההבדלים ביניהם.
- בעת לימוד פעולת השרשור מומלץ להראות כי הוראת ההדפסה למסך `println` משתמשת בשרשור מחרוזות.
- יש להדגיש כי הפעולות של המחלקה מחרוזת אינן משנות את ערך המחרוזת השמור בעצם המפעיל, אלא מחזירות מחרוזת חדשה, השמורה בשטח זיכרון חדש המכיל מחרוזת הזוהה בערכה לערך המחרוזת המקורית לאחר השינוי המבוקש.
- יש להדגיש כי בעת השמת מחרוזות, הערך המושם הוא כתובת המחרוזת השמורה בזכרון, ולא ערך המחרוזת.

פתרונות

שאלה 9.1

פתחו וישמו אלגוריתם שמקבל כקלט רשימה של 100 מחרוזות. האלגוריתם מחשב בכמה מחרוזות מתוך ה-100 התו הראשון שווה לתו האחרון, ומציג את הערך שחישב.

תשובה 9.1

רעיון הפתרון:

לפתרון שאלה זו מתאים להשתמש בתבנית **מניה**. עלינו למנות את מספר המחרוזות בהן התו הראשון שווה לתו האחרון. את הבדיקה האם התו הראשון שווה לתו האחרון נבצע בעזרת הפעולה `charAt`. את ערך התו הראשון של המחרוזת `st` נמצא על ידי `st.charAt(0)` ואילו את ערך התו האחרון של המחרוזת `st` נמצא על ידי `st.charAt(st.length()-1)`.

תבניות:

מניה

פעולות של המחלקה `String`:

`charAt`

בחירת משתנים:

`st` – מחרוזת, המחרוזת התורנית ברשימת מחרוזות הקלט

`count` – שלם, מונה למניית מחרוזות המקיימות את התנאי

יישום האלגוריתם:

```
/* קלט: רשימה של 100 מחרוזות
פלט: בכמה מחרוזות מתוך ה-100 התו הראשון שווה לתו האחרון
*/
import java.util.Scanner;
public class FirstLastEquals
{
    public static void main(String[] args)
    {
        String st;
        int count = 0;
        Scanner in = new Scanner(System.in);
        for (int i=0; i<100; i++)
        {
            System.out.print("Enter a string: ");
            st = in.nextLine();
            if (st.charAt(0) == st.charAt(st.length()-1))
                count++;
        }
        System.out.println(count);
    }
}
// main
}
// FirstLastEquals
```

שאלה 9.3

פתחו אלגוריתם המקבל כקלט רשימת מחרוזות המסתיימת במחרוזת "****". פלט האלגוריתם יהיה אורך המחרוזת הארוכה ביותר ברשימה. ישמו את האלגוריתם בשפת Java.

תשובה 9.3

רעיון הפתרון:

לפתרון שאלה זו מומלץ להשתמש בתבנית **מציאת מקסימום**, כאשר הערך הנבדק יהיה אורך המחרוזת. על פי דרישות השאלה, הביצוע החוזר יתבצע על ידי לולאת זקיף. שימו לב כי ההשוואה בין המחרוזת הנקלטת st לזקיף "****" תעשה על ידי הפעולה `st.equals("****")`.

תבניות:

מציאת מקסימום

פעולות של המחלקה String:

`equals`

`length`

בחירת משתנים:

st – מחרוזת, המחרוזת התורנית ברשימת מחרוזות הקלט
max – שלם, אורך המחרוזת הארוכה ביותר שנקלטה עד כה

יישום האלגוריתם:

```
/* קלט: רשימת מחרוזות המסתיימת במחרוזת "****"
   פלט: אורך המחרוזת הארוכה ביותר ברשימה
*/
import java.util.Scanner;
public class MaxLength
{
    public static void main(String[] args)
    {
        String st;
        int max = 0;
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a string: ");
        st = in.nextLine();
        while (!st.equals("****"))
        {
            if (st.length() > max)
                max = st.length();
            System.out.print("Enter a string: ");
            st = in.nextLine();
        }
        System.out.println("Max length is: " + max);
    }
}
// main
// MaxLength
```

שאלה 9.4

פתחו אלגוריתם שיקבל כקלט שתי מחרוזות ויציג אותן לפי סדר הא"ב. ישמו את האלגוריתם בשפת Java.

תשובה 9.4

רעיון הפתרון:

לפתרון שאלה זו עלינו להשוות בין שתי מחרוזות הקלט st1 ו-st2 באופן לקסיקוגרפי. השוואה זו תעשה על ידי הפעולה st1.compareTo(st2).

פעולות של המחלקה String:

compareTo

בחירת משתנים:

st1 – מחרוזת הקלט הראשונה

st2 – מחרוזת הקלט השנייה

יישום האלגוריתם:

```
/* קלט: שתי מחרוזות
   פלט: המחרוזות לפי סדר הא"ב
   */
import java.util.Scanner;
public class ABStrings
{
    public static void main(String[] args)
    {
        String st1, st2;
        Scanner in = new Scanner(System.in);
        System.out.print("Enter two strings: ");
        st1 = in.nextLine();
        st2 = in.nextLine();
        if (st1.compareTo(st2) < 0)
            System.out.println(st1 + " " + st2);
        else
            System.out.println(st2 + " " + st1);
    } // main
} // ABStrings
```

שאלה 9.5

פתחו וישמו אלגוריתם שיקבל כקלט מילה באנגלית ויציג אותה פעמיים, פעם באותיות גדולות ופעם באותיות קטנות. למשל עבור הקלט Memory הפלט יהיה: MEMORY ומיד אחר-כך memory. חפשו פעולות מתאימות בטבלת הפעולות המופיעה בסוף הפרק.

תשובה 9.5

רעיון הפתרון:

לפתרון שאלה זו יש להפעיל על מחרוזת הקלט st את הפעולות המתאימות: st.toUpperCase() ו-st.toLowerCase(). יש להדגיש לתלמידים כי כל אחת מפעולות אלו אינה משנה את ערך המחרוזת עליה היא מופעלת, אלא מחזירה מחרוזת חדשה המכילה את הערך החדש.

פעולות של המחלקה String:
toUpperCase
toLowerCase

בחירת משתנים:

st – מחרוזת הקלט

יישום האלגוריתם:

```
/* קלט: מחרוזת
פלט: המחרוזת פעם באותיות גדולות ופעם באותיות קטנות
*/
import java.util.Scanner;
public class StringUpDown
{
    public static void main(String[] args)
    {
        String st;
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a string: ");
        st = in.nextLine();
        System.out.println(st.toUpperCase());
        System.out.println(st.toLowerCase());
    } // main
} // StringUpDown
```

שאלה 9.6

פתחו אלגוריתם שיקבל כקלט 100 מחרוזות. עבור כל זוג מחרוזות תוצג כפלט מחרוזת שהיא שרשור של שתי המחרוזות עם הסימן '@' ביניהן (כלומר בסך הכול תוצגנה 50 מחרוזות). ישמו את האלגוריתם בשפת Java.

תשובה 9.6

רעיון הפתרון:

לפתרון שאלה זו נשתמש בפעולות השרשור (+) בין מחרוזות. נשרשר מחרוזת קלט ראשונה למחרוזת "@", ולמחרוזת המתקבלת נשרשר מחרוזת קלט שנייה. הביצוע יחזור על עצמו 50 פעמים.

פעולות של המחלקה String:
שרשור

בחירת משתנים:

st1 – מחרוזת הקלט הראשונה
st2 – מחרוזת הקלט השנייה

יישום האלגוריתם:

```
/* קלט: 100 מחרוזות
פלט: עבור כל זוג מחרוזות יוצג שרשור של שתי המחרוזות עם הסימן '@'
ביניהן
*/
import java.util.Scanner;
public class ConcatenatedStrings
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        for (int i=0; i<50; i++)
        {
            System.out.print("Enter two strings: ");
            String st1 = in.nextLine();
            String st2 = in.nextLine();
            System.out.println(st1 + "@" + st2);
        }
    }
} // main
} // ConcatenatedStrings
```

שאלה 9.8 (מתוך בגרות 1995)

פתחו וישמו אלגוריתם הקולט מחרוזת. האלגוריתם מציג כפלט כל תו במחרוזת פעמיים, פרט לתו '*' (כוכבית). גם אם התו כלול במחרוזת הוא לא מוצג כלל. הפלט מוצג בשורה אחת. למשל עבור הקלט: AB*3B*? הפלט יהיה A.ABB33BB??.

תשובה 9.8

רעיון הפתרון:

לפתרון שאלה זו עלינו לסרוק את כל תווי מחרוזת הקלט. לשם כך נשתמש בלולאה אשר מספר הפעמים שהיא תבצע הוא כאורך מחרוזת הקלט. את אורך מחרוזת הקלט נדע לפי הערך המוחזר מהפעולה `st.length()`. כל תו מתווי המחרוזת עלינו להדפיס פעמיים, בתנאי שאיננו התו כוכבית. את התו התורן נשלוף מהמחרוזת `st` על ידי הפעולה `st.charAt(i)`, כאשר `i` הוא משתנה הבקרה של הלולאה. את התו התורן נשמור במשתנה `ch` על מנת להימנע משימוש בפעולה `st.charAt(i)` מספר רב של פעמים.

פעולות של המחלקה `String`:

`length`

`charAt`

בחירת משתנים:

`st` – מחרוזת הקלט

`ch` – התו התורן הנבדק

יישום האלגוריתם:

```
/* קלט: מחרוזת
פלט: כל תו במחרוזת יוצג פעמיים, פרט לתו '*'
*/
import java.util.Scanner;
public class DoubleString
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String st = in.nextLine();
        for (int i=0; i<st.length(); i++)
        {
            char ch = st.charAt(i);
            if (ch != '*')
            {
                System.out.print(ch);
                System.out.print(ch);
            }
        }
    }
}
// main
}
// DoubleString
```

שאלה 9.11

פתחו אלגוריתם המקבל כקלט רשימת מחרוזות המסתיימת במחרוזת "stop". פלט האלגוריתם יהיה המחרוזת הארוכה ביותר ברשימה. ישמו את האלגוריתם בשפת Java.

תשובה 9.11

רעיון הפתרון:

פתרון שאלה זו דומה לפתרון שאלה 9.3, בהבדל אחד: בפתרון זה מתאים להשתמש בתבנית **מציאת ערך נלווה למקסימום בסדרה**. ערך המקסימום הוא אורך המחרוזת הארוכה ביותר, והערך הנלווה לה הוא המחרוזת הארוכה ביותר.

תבניות:

מציאת ערך נלווה למקסימום

פעולות של המחלקה String:

equals

length

בחירת משתנים:

st – מחרוזת, המחרוזת התורנית ברשימת מחרוזות הקלט
max – שלם, אורך המחרוזת הארוכה ביותר שנקלטה עד כה
maxSt – מחרוזת, המחרוזת הארוכה ביותר שנקלטה עד כה

יישום האלגוריתם:

```
/* קלט: רשימת מחרוזות */
/* פלט: המחרוזת הארוכה ביותר ברשימה */
import java.util.Scanner;
public class MaxLength2
{
    public static void main(String[] args)
    {
        String st, maxSt="";
        int max = 0;
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a string: ");
        st = in.nextLine();
        while (!st.equals("stop"))
        {
            if (st.length() > max)
            {
                max = st.length();
                maxSt = st;
            }
            System.out.print("Enter a string: ");
            st = in.nextLine();
        }
        System.out.println("Max string is: " + maxSt);
    }
} // main
} // MaxLength2
```


שאלה 9.14

רן ודן המציאו שפה מוצפנת. המשפטים הניתנים להצפנה כוללים מילים ורווחים בלבד, בהינתן שכל מילה נכתבת רק באותיות אנגליות קטנות. ההצפנה מתבצעת באופן הבא: כל רווח מוחלף בסימן '@', ולאחר כל אות מופיעה האות האנגלית הגדולה המתאימה לה. למשל, המשפט good morning מוצפן כך gGoOoOdD@mMoOrRnNiInNgG. פתחו אלגוריתם המקבל משפט כקלט ומציג את המשפט המוצפן המתאים לו כפלט. ישמו את האלגוריתם בשפת Java.

תשובה 9.14

לשאלה זו נציג שני פתרונות שונים. כל שפתרון מציג פן אחר בלימוד נושא המחרוזות. מומלץ להציג לתלמידים את שני הפתרונות.

רעיון פתרון א':

לפתרון שאלה זו אנו נדרשים להמיר אות לועזית קטנה לאות לועזית גדולה. המרה זו נלמדה בפרק 4, בבעיה פתורה 4. בהנחה שבתו ch שמורה אות לועזית קטנה, הנוסחה להמרת אות לועזית קטנה זו לאות הלועזית הגדולה המתאימה לה היא: $(ch + 'a' - 'A')$. להסבר מפורט על נוסחה זו פנו לעמודים 84-86 בפרק 4.

פעולות של המחלקה String לפתרון א':

length
charAt

בחירת משתנים לפתרון א':

st – מחרוזת הקלט
encodedSt – המחרוזת מוצפנת
ch – התו התורן הנבדק

יישום האלגוריתם לפתרון א':

```
/* קלט: מחרוזת
פלט: המחרוזת מוצפנת
*/
import java.util.Scanner;
public class EncodeString1
{
    public static void main(String[] args)
    {
        String st, encodedSt="";
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a string: ");
        st = in.nextLine();
        for (int i=0; i<st.length(); i++)
        {
            char ch = st.charAt(i);
            if (ch == ' ')
                encodedSt += '@';
            else
            {
```

```

        encodedSt += ch;
        encodedSt += (char) (ch + 'A'-'a');
    }
}
System.out.println("Encoded string is: " + encodedSt);
} // main
} // EncodeString1

```

רעיון פתרון ב':

לפתרון שאלה זו ניצור מחרוזת נוספת, הזהה למחרוזת הקלט בעלת האותיות הלועזיות הקטנות, כאשר כל האותיות בה הן לועזיות גדולות. לשם כך נפעיל על מחרוזת הקלט st את הפעולה st.toUpperCase(). לבניית המחרוזת המוצפנת, נשרשר לסירוגין תו אחד ממחרוזת הקלט, ותו אחד ממחרוזת הקלט בעלת האותיות הגדולות.

פעולות של המחלקה String לפתרון ב':

length
toUpperCase

בחירת משתנים לפתרון ב':

st – מחרוזת הקלט

stUp – מחרוזת הקלט בעלת אותיות לועזיות גדולות

encodedSt – המחרוזת מוצפנת

יישום האלגוריתם לפתרון ב':

```

/* קלט: מחרוזת
   פלט: המחרוזת מוצפנת
*/
import java.util.Scanner;
public class EncodeString2
{
    public static void main(String[] args)
    {
        String st, encodedSt="";
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a string: ");
        st = in.nextLine();
        String stUp = st.toUpperCase();
        for (int i=0; i<st.length(); i++)
        {
            if (st.charAt(i) == ' ')
                encodedSt += '@';
            else
            {
                encodedSt += st.charAt(i);
                encodedSt += stUp.charAt(i);
            }
        }
        System.out.println("Encoded string is: " + encodedSt);
    } // main
} // EncodeString2

```

שאלה 9.17

כתובת אתר אינטרנט של חברה מסחרית בינלאומית בנויה בדרך כלל מ-3 חלקים המופרדים בנקודות:

www.שם החברה.com

פתחו אלגוריתם המקבל כקלט כתובת של אתר של חברה מסחרית בינלאומית, במבנה שתואר לעיל, ומציגה כפלט את שם החברה בלבד. ישמו את האלגוריתם בשפת Java. **הדרכה:** השתמשו בפעולה `substring` שבטבלת הפעולות הנמצאת בסוף הפרק.

תשובה 9.17

פעולות של המחלקה `String`:

`substring`

`indexOf`

`lastIndexOf`

יישום האלגוריתם:

```
/* קלט: כתובת של אתר של חברה מסחרית בינלאומית
פלט: שם החברה
*/
import java.util.Scanner;
public class CompanyName
{
    public static void main(String[] args)
    {
        String url, compName;
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a company URL: ");
        url = in.nextLine();
        compName =
            url.substring(url.indexOf('.')+1,url.lastIndexOf('.'));
        System.out.println("The company name is: " + compName);
    } // main
} // CompanyName
```

שאלה 9.18

כתובת אתר אינטרנט של חברה מסחרית שאינה בינלאומית בנויה בדרך כלל מ-3 חלקים המופרדים בנקודות:

www.שם המדינה.שם החברה.com

פתחו אלגוריתם המקבל כקלט רשימת כתובות של אתרי חברות מסחריות כאלה (כל אחת מהן במבנה שתואר לעיל). הרשימה תסתיים במחרוזת "end". האלגוריתם יציג כפלט עבור כל חברה את שמה ואת סיומת המדינה שלה. ישמו את האלגוריתם בשפת Java.

תשובה 9.18

פעולות של המחלקה `String`:

`equals`

`substring`

`indexOf`

`lastIndexOf`

יישום האלגוריתם:

```
/* קלט: רשימת כתובות של אתרי חברות מסחריות שאינן בינלאומיות
פלט: עבור כל חברה שמה וסיומת המדינה שלה
*/
import java.util.Scanner;
public class CompanyName2
{
    public static void main(String[] args)
    {
        String url;
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a company URL: ");
        url = in.nextLine();
        while (!url.equals("end"))
        {
            int dotIndex = url.indexOf('.');
            int lastDotIndex = url.lastIndexOf('.');
            String compName =
                url.substring(dotIndex+1, lastDotIndex);
            String country = url.substring(lastDotIndex+1);
            System.out.print("The company name is: " +
                compName);
            System.out.println("\tThe country is: " + country);
            System.out.print("Enter a company URL: ");
            url = in.nextLine();
        }
    }
} // main
} // CompanyName2
```