

תרגיל 10 – מחלקות אובייקטים ואוספים

שאלה 1

כתוב מחלקה בשם **MyString** שדומה למחלקת **String**.

משתני עצם המחלקה:

◀ מערך של תווים (**char[] data**) המחזיק "מחרוזת". אורך התחלתי של המערך נקבע ע"י משתנה
final int INIT_SIZE = 256
◀ **length** - מספר תווים שבפועל נמצאים בתוך המערך

בנאים:

◀ בנאי רגיל שמקבל מערך של תווים: **public MyString(char[] c){...}**
◀ בנאי מעתיק שמקבל אובייקט מטיפוס **MyString**: **public MyString(MyString ms){...}**

מתודות:

◀ **public int length(){ ... }** - המתודה מחזירה מספר תווים שבפועל נמצאים בתוך המערך

◀ **public void concat(MyString s){ ... }** - המתודה מחברת שתי "מחרוזות": מחרוזת **s** ו"מחרוזת" מקורית. המקור צריך להשתנות.

◀ **public char charAt(int index) { ... }** - המתודה מחזירה תו הנמצא במיקום **index**.

◀ **public String toString(){ ... }** - המתודה מחזירה מחרוזת המורכבת מתווים הנמצאים במערך **data**.

◀ **public MyString substr(int beg, int end)** - המתודה מחזירה אובייקט מטיפוס **MyString**, והמערך של האובייקט מכיל את תת-מערך של המערך המקורי ממיקום **beg** (כולל) עד מיקום **end** (לא כולל).

◀ **public boolean equals(MyString s) {...}** - המתודה משווה בין שני אובייקטים מטיפוס **MyString**. המתודה מחזירה אמת אם האובייקטים מחילים מערכי **data** שווים. שני מערכים שווים אם הם מכילים אותם איברים ובאותו סדר.

◀ **public int compare(MyString s) {...}** - המתודה מאפשרת השוואה של שני אובייקטים מטיפוס **MyString**. הסדר הוא סדר מילוני. כאשר המקור גדול מ-**s** המתודה מחזירה 1, כאשר המקור קטן מ-**s** המתודה מחזירה -1, כאשר הם שווים המתודה מחזירה 0.

שאלה 2

כתוב מחלקה בשם `StringsContainer` שתייצג אוסף של מחרוזות. המחלקה תאפשר להוסיף לאובייקט קיים מחרוזת חדשה. המחרוזת החדשה תתווסף בסוף הרשימה הקיימת.

בנאי:

```
public StringsContainer ()
```

בנאי מעתיק:

```
public StringsContainer (StringsContainer sc)
```

יצירה של אובייקט חדש שאינו מכיל מחרוזות.
עליך לאחסן את אוסף המחרוזות באמצעות מערך.

למחלקה קבוע פרטי ומוגדרת בה פעולת גידול (שיטה פרטית):

```
public static final int INIT_SIZE = 3;
```

קבוע `INIT_SIZE` הוא גודל המערך הראשוני שייבצר בעת יצירת אובייקט חדש, וגם הוא מידת הגדילה של המערך במידה ואין בו יותר מקום לקלוט מחרוזת חדשה.
פעולת הגדילה תתבצע רק כאשר יש צורך להוסיף מחרוזת ואין מקום.

כתוב את השיטות הבאות:

1. הפונקציה מגדילה את המערך – מוסיפה לו `INIT_SIZE` תאים
`private void resize ()`

2. הפונקציה מוסיפה מחרוזת חדשה לאוסף
`public void add (String str)`

3. הפונקציה מחזירה את כל המחרוזות
`public void toString()`

4. הפונקציה מחזירה מספר מחרוזות הנמצאות בתוך האוסף
`public int size()`

5. הפונקציה מחזירה מחרוזת הנמצאת במקום `index`
`public String at(int index)`

6. הפונקציה מחליפה מחרוזת `oldStr` במחרוזת `newString`
`public void replace (String oldStr, String newString)`

7. הפונקציה מחזירה את המחרוזת הארוכה ביותר
`public String longestString()`

8. הפונקציה מחזירה `true` אם מחרוזת `str` נמצאת בתוך האוסף, אחרת היא מחזירה `false`
`Public boolean contains(String str)`

9. הפונקציה מוחקת מחרוזות `str` מהאוסף

```
public void remove(String str)
```

10. הפונקציה מחזירה אוסף חדש שמכיל את המחרוזות של אוסף המקורי שנמצאות גם באוסף `other`

```
public StringContainer intersection2(StringContainer other)
```

11. הפונקציה מחזירה `true` אם המחרוזות שוות, כלומר מורכבות בדיוק מאותם איברים, אחרת מחזירה `false`.
הסדר אינו חשוב. ניתן להניח כי קונטיינר מכיל מחרוזות שונות.

```
public boolean contEqual(StringContainer other)
```

```
// create new Container that contains only those strings of the first  
// Container that the other Container does not contain them  
public StringContainer Diff2(StringContainer other)
```

כתוב מחלקה חדשה TestStringContainer לבדיקת השיטות של StringContainer