מטלת מנחה (ממ"ן) 15

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידה 11 | **נושא המטלה:** רשימות מקושרות

מספר השאלות: 1 | **משקל המטלה:** 5 נקודות

סמסטר: 2021ב | **מועד אחרון להגשה:** 12.6.2021

ברצוננו לחשב כל מיני נתונים הקשורים לטקסטים. לשם כך, בהינתן טקסט אלפביתי כלשהו (נניח, ספר או מאמר), עלינו ליצור מבנה נתונים המאחסן את המילים המופיעות בו. לכל מילה נציין גם כמה פעמים היא מופיעה בטקסט.
נרצה לבצע שאילתות על הטקסט ומילותיו. לשם כך נגדיר מחלקה המייצגת את קבוצת המילים שבטקסט. אין צורך לשמור על סדר המילים כפי שהופיעו בטקסט. **לצורך הפשטות, נניח שהטקסט מכיל אך ורק אותיות קטנות מהאלף-בית האנגלי ('a'-'z'), כלומר 26 אותיות בלבד, ותו רווח בודד (' ') המפריד בין מילה למילה.**

המחלקה תייצג את קבוצת המילים בצורת רשימה ממוינת לפי סדר מילוני (לקסיקוגרפי).

עליכם ליצור מחלקה שנקראת WordNode שמייצגת איבר אחד ברשימה, ומחלקה שנקראת TextList שמייצגת את הרשימה כולה.

המחלקה WordNode תכיל **לפחות** את שתי התכונות הבאות:

1. String _word
2. WordNode _next

עליכם לכתוב את הבנאים והשיטות הדרושות לשימוש במחלקה WordNode לפי המקובל בקורס. (תוכלו להיעזר במחלקה IntNode שהוגדרה בהרצאות). לפי המחלקה TextList חשבו אם אתם רוצים שהמחלקה WordNode תכיל תכונות נוספות.

אין להפוך את הרשימה לדו כיוונית.

שימו לב שכאן aliasing הוא לא טעות. יש לעדכן ולהחזיר את המידע next עצמו ולא עותק.

עליכם לכתוב את הגדרת המחלקה **TextList**, כאשר התכונה (instance variable) **היחידה** המותרת היא הצבעה לתחילת הרשימה המקושרת. **ברור שתכונות נוספות יכולות להפוך את השיטות ליעילות יותר, ובכל זאת – במחלקה TextList אין להגדיר אף תכונה נוספת!**

המחלקה TextList תכיל שני בנאים שיוגדרו להלן:

- `public TextList ()`

בנאי ריק המאתחל את בסיס הנתונים.

- `public TextList (String text)`

בנאי המקבל כפרמטר מחרוזת שמייצגת טקסט (אוסף של מילים) **ובונה** מהמחרוזת את הרשימה. אתם יכולים להניח שהמחרוזת מכילה רק מילים מהאלף-בית האמור, עם אותיות קטנות בלבד, בלי סימני פיסוק או סימנים אחרים, וכן יש רווח יחיד בין כל מילה. **אם המחרוזת ריקה יש לבנות רשימה המייצגת מחרוזת ריקה. אפשר להניח שלא התקבל null כפרמטר.**

זכרו שהרשימה צריכה להיות ממוינת לפי סדר לקסיקוגרפי (מילוני).

הקפידו במיוחד על יעילות! ציון מלא יינתן רק עבור סיבוכיות מיטבית.

השיטות שיוגדרו במחלקה TextList הן:

- `public void addToData (String word)`

המקבלת מילה ומוסיפה אותה למבנה הנתונים. **שימו לב לכל מקרי הקצה האפשריים (חשבו מהם).**

אם המחרוזת שמתקבלת כפרמטר ריקה לא יתבצע דבר. אפשר להניח שלא התקבל null כפרמטר.

- `public int howManyWords ()`

המחזירה את מספר המילים **הכולל** שבטקסט;

לדוגמא: עבור הטקסט "anything you can do i can do better" יוחזר הערך 8.

- `public int howManyDifferentWords ()`

המחזירה את מספר המילים **השונות** שבטקסט;

לדוגמא: עבור הטקסט "anything you can do i can do better" יוחזר הערך 6.

- `public String mostFrequentWord ()`

המחזירה את המילה השכיחה ביותר בטקסט; אם יש יותר ממילה אחת כזאת, תוחזר הראשונה ביניהן לפי מיון הרשימה.

לדוגמא: עבור הטקסט "anything you can do i can do better" תוחזר המילה "can".

אם הרשימה ריקה יש להחזיר מחרוזת ריקה.

- `public int howManyStarting (char letter)`

המקבלת אות, ומחזירה את מספר המילים בטקסט שמתחילות באות זו;

לדוגמא: עבור הטקסט "anything you can do i can do better" והאות 'c' יוחזר הערך 2.

- `public char mostFrequentStartingLetter ()`

המחזירה את האות שהכי הרבה מילים מתחילות בה בטקסט; **אם יש יותר מאות אחת כזאת, תוחזר הראשונה מביניהן, לפי סדר מיון הרשימה.**

לדוגמא: עבור הטקסט "anything you can do i can do better" תוחזר האות 'c'.

שימו לב ,זה כולל כפילויות של מילים . כלומר ,מילה שמופיעה פעמיים , גם האות שבה היא מתחילה נספרת פעמיים.

אם הרשימה ריקה יש להחזיר תו רווח.

שימו לב, השיטה הזו צריכה להיות ממומשת ברקורסיה וללא לולאות בכלל (הרקורסיה עצמה עשויה להיות מוגדרת בשיטת עזר)!

- `public String toString()`

המעבירה את הרשימה למחרוזת, כאשר כל המילים מופיעות בה, ולצד כל מילה מופיע מספר המופעים שלה בטקסט (עם טאב – '\t' - ביניהם); לאחר כל מילה ומספר מופעיה יש לרדת שורה.

אם הרשימה ריקה יש להחזיר מחרוזת ריקה.

לדוגמא: עבור הטקסט "anything you can do i can do better" תוחזר המחרוזת **(בדיוק בפורמט זה)**

```
anything   1
better     1
can   2
do    2
i     1
you   1
```

שימו לב:

- **אנחנו כתבנו את הממשק הפומבי ועליו לא ניתן להוסיף שיטות, אולם ניתן להוסיף שיטות פרטיות לנוחות וקריאות המחלקה.**
- **אתם צריכים כמובן לכתוב API** ותיעוד פנימי **לשתי המחלקות.**
- **בכל השיטות, שימו לב לכל מקרי השגיאה האפשריים!**
- **כתבו כהערה ב- API מה סיבוכיות הזמן וסיבוכיות המקום של כל שיטה שכתבתם (למעט השיטה הרקורסיבית** `mostFrequentStartingLetter`**).**
- **הקפידו על יעילות השיטות שכתבתם! ציון מלא יינתן רק עבור סיבוכיות מיטבית.**
- **אסור להשתמש במחלקות מוכנות כבר של Java.**

אתם יכולים להשתמש בשיטות הבאות במחלקה String (**ולא באחרות ובפרט לא בשיטה split**):

- public char charAt(int i) - המחזירה את התו במקום ה- i במחרוזת (עליה היא מופעלת)
- public String concat (String str) - המחזירה מחרוזת המורכבת מהמחרוזת עליה היא מופעלת ובסופה משורשרת המחרוזת str.
- public int indexOf (int ch) - המחזירה את האינדקס במחרוזת עליה היא מופעלת של המופע הראשון של התו ch. אם התו ch לא מופיע במחרוזת, יוחזר 1-.
- public int indexOf (int ch, int fromIndex) - המחזירה את האינדקס במחרוזת עליה היא מופעלת של המופע הראשון של התו ch, כאשר החיפוש מתחיל באינדקס fromIndex. אם התו ch לא מופיע במחרוזת, יוחזר 1-.
- public int indexOf (String str) - המחזירה את האינדקס במחרוזת עליה היא מופעלת של המופע הראשון של המחרוזת str. אם המחרוזת str לא מופיעה במחרוזת שעליה היא מופעלת, יוחזר 1-.
- public int indexOf (String str, int fromIndex) - המחזירה את האינדקס במחרוזת עליה היא מופעלת של המופע הראשון של המחרוזת str, כאשר החיפוש מתחיל באינדקס fromIndex. אם המחרוזת str לא מופיעה במחרוזת שעליה היא מופעלת, יוחזר 1-.
- public boolean equals (String str) - המחזירה true אם המחרוזת עליה היא מופעלת זהה למחרוזת str. אחרת יוחזר false.
- public int compareTo (String str) - המשווה בין המחרוזת עליה מופעלת השיטה למחרוזת str שבפרמטר. השיטה מחזירה את הערך 0 אם המחרוזות שוות. אם המחרוזת שבאובייקט קטנה לקסיקוגרפית מהמחרוזת str שבפרמטר יוחזר מספר שלילי, ואם המחרוזת שבאובייקט גדולה לקסיקוגרפית מהמחרוזת str יוחזר ערך חיובי.
- public int length() - המחזירה את אורך המחרוזת עליה היא מופעלת.
- public String substring(int i) - המחזירה את התת-מחרוזת המתחילה במקום ה- i במחרוזת עליה היא מופעלת עד לסוף המחרוזת.
- public String substring(int i, int j) - המחזירה את התת-מחרוזת המתחילה במקום ה- i במחרוזת עליה היא מופעלת עד לתו במקום ה- j (לא כולל j עצמו).

כללי הגשה

1. הגשת הממ״ן נעשית בצורה אלקטרונית בלבד, דרך מערכת שליחת המטלות.
2. **הקפידו ששמות השיטות והמחלקות יהיו בדיוק לפי הוראות הממ״ן!**
3. **באתר הקורס תמצאו טסטר בסיסי לבדיקת המחלקה TextList.**
   חובה שהטסטר ירוץ מול המחלקה שלכם ללא שגיאות קומפילציה .
4. **גם לקראת סוף הקורס יש להקפיד על תיעוד API ותיעוד פנימי.**
5. את התשובות לשאלות יש להגיש בשני קבצי Java הבאים : WordNode.java, TextList.java,
   ארוזים יחד בתוך קובץ zip יחיד. אין לשלוח קבצים נוספים.