

מטלת מנחה (ממ"ן) 17

הקורס: 20441 - מבוא למדעי המחשב ושפת Java

חומר הלימוד למטלה: יחידות 17 - 18 נושא המטלה: רשימה מקושרת

מספר השאלות: 1 משקל המטלה: 5 נקודות

סמסטר: 2009 מועד אחרון להגשה: 27.6.2009

(ת)

שאלה 1 - להרצה (100%)

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 (' ') המפריד בין מילה למילה.

המחלקה תייצג את קבוצת המילים בצורת רשימה ממוינת, לפי האות הראשונה במילה (אין חשיבות למיון נוסף, עבור מילים בעלות אות תחילית זהה). על מנת להשוות בין תווים, אפשר להשתמש ב-JAVA באופרטורים הפשוטים: '<', '>' ו-'=' (ממש כאילו היו אלה מספרים).

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 של פרופ' רוזנשיין). לפי המחלקה `TextList` חשבו אם אתם רוצים שהמחלקה `WordNode` תכיל תכונות נוספות.

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 – אין להגדיר אף תכונה נוספת!

המחלקה TextList תכיל שני בנאים ושיטות 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.
- `public void addToData (String word)`
המקבלת מילה שנקראה מהטקסט ומוסיפה אותה למבנה הנתונים. **שימו לב לכל מקרי הקצה האפשריים (חשבו מהם).**
- `public int howManyWords ()`
המחזירה את מספר המילים שבטקסט;
- `public int howManyDifferentWords ()`
המחזירה את מספר המילים השונות שבטקסט;
- `public String mostFrequentWord ()`
המחזירה את המילה השכיחה ביותר בטקסט; אם יש יותר ממילה אחת כזאת, תוחזר הראשונה ביניהן לפי מיון הרשימה.
- `public char mostFrequentStartingLetter ()`
המחזירה את האות שהכי הרבה מילים מתחילות בה בטקסט; **שימו לב, השיטה הזו צריכה להיות ממומשת ברקורסיה וללא לולאות בכלל (הרקורסיה עצמה עשויה להיות מוגדרת בשיטת עזר)! אם יש יותר מאות אחת כזאת, תוחזר הראשונה מביניהן, לפי סדר מיון הרשימה.**
- `public int howManyStarting (char letter)`
המקבלת אות, ומחזירה את מספר המילים בטקסט שמתחילות באות זו;
- `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 מופיעה יש לרדת שורה. לדוגמא: עבור הטקסט "hello world hello first" תוחזר המחרוזת **(בדיוק בפורמט זה)**

```
first    1
hello    2
world    1
```

שימו לב לכל מקרי השגיאה האפשריים!

כתבו 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.

שימו לב: אסור להשתמש במחלקות מוכנות של Java. מותר ורצוי להשתמש במחלקות שניתנו בהרצאה ונמצאות בחוברת השקפים. כמובן שניתן לכתוב שיטות עזר, אך יש להגדירן כשיטות פרטיות.

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. **גם לקראת סוף הקורס יש להקפיד על תיעוד API.**
4. את התשובות לשאלות יש להגיש בשני קבצי Java הבאים: WordNode.java, TextList.java, ארוזים יחד בתוך קובץ zip יחיד. אין לשלוח קבצים נוספים.