

פת"מ 1 סמסטר א' מועד א2 תש"ף

מרצה: ד"ר אליהו חלסצ'י

שאלה 1 – generics, containers, lambdas (נק' 34):

בקובץ Dictionary.java עליכם לממש את המחלקה Dictionary.

- הבנאי מקבל מחרוזת המכילה טקסט.
 - הטקסט עשוי להיות ארוך ומילים עשויות לחזור על עצמן מספר פעמים בטקסט.
 - המילים בטקסט מופרדות ע"י white spaces כגון תו בודד או רצף של רווחים, טאבים, ירידות שורה וכדומה.
 - מותר לכם להוסיף פעולות לבנאי.
- עליכם לממש את המתודה getList כך שבהינתן תו (char) המתודה תחזיר ב $O(1)$ זמן רשימה של כל המילים מתוך הטקסט שמתחילות בתו זה.
 - על הרשימה להיות מסודרת לפי סדר לקסיקוגרפי
 - וכל מילה צריכה להופיע ברשימה בדיוק פעם אחת.
 - אם אין מילים שמתחילות בתו זה אז המתודה תחזיר רשימה ריקה

דוגמה להפעלה:

```
Dictionary dic=new Dictionary("hello world! world world wonderful bye");
```

```
List<String> ws=dic.get('w'); // O(1) time
```

```
ws.forEach(System.out::println); // print the list
```

```
// output:
```

```
// wonderful
```

```
// world
```

```
// world!
```

נשים לב שהמילה world! (עם סימן קריאה למשל) שונה מהמילה world. וכן שהמילה world מופיעה פעמיים בקלט אך רק פעם אחת בפלט. לכן הפלט יצא כמתואר לעיל.

בדיקה:

- קוד מתקמפל ורץ ללא שגיאות ריצה (10 נק')
- כל התווים הרלוונטיים נמצאים במילון (6 נק')
- כל המילים מהטקסט מקובצות לפי האות הראשונה שלהן (8 נק')
- לכל אות – רשימת המילים ממוינת וללא חזרות (10 נק')

ב MainTrain1 נמצא מוד האימון לשאלה זו שכולל בדיקה שטחית.

שאלה זו אף תיבדק ידנית.

שאלה 2 – קבצים (33 נק'):

בקובץ Q2.java עליכם לממש את המתודה writeWords כך שבהינתן שני שמות של קובצי טקסט – קלט (source) ופלט (destination), המתודה תקרא את תוכן קובץ הקלט, ותיצור קובץ פלט לפי החוקיות הבאה:

- כל מילה בקובץ הפלט תופיע רק פעם אחת גם אם היא הופיעה מס' פעמים בקלט
- סדר הופעת המילים בקובץ הפלט יהיה לפי מספר ההופעות של כל מילה בקובץ הקלט – מהגדול לקטן.
- במקרה של מספר הופעות שווה, הסדר יהיה לקסיקוגרפי
- כל מילה בקובץ הפלט תופיע בשורה נפרדת

שימו לב שאין להוסיף לחתימה של המתודה זריקה של Exception כלשהו.

לדוגמה הביטוי ב MainTrain2. יצרנו קובץ שמכיל את הטקסט הבא:

```
hello    world world
world    good by
```

המילים מופרדות ע"י white spaces כגון רווחים טאבים וירידות שורה.

לאחר ההפעלה של writeWords() קובץ הפלט יראה כך:

```
world // כי יש לה 3 הופעות בקלט
by     // הופעה אחת, אך ראשונה לקסיקוגרפית
good   // הופע האחת, שנייה לקסיקוגרפית
hello  // הופעה אחת, שלישית לקסיקוגרפית
```

בדיקה:

ב MainTrain2 ישנה בדיקה שטחית על דוגמה פשוטה. במוד ההגשה הקבצים יהיו מורכבים יותר. הבדיקה לשאלה זו **אוטומטית לחלוטין** ולפיכך הקוד חייב להתקמפל ולרוץ ללא שגיאות על מנת להיבדק.

שאלה 3 – תבניות עיצוב (33 נק'):

בקובץ MyStringBuilder.java עליכם ליצור חיקוי של StringBuilder. לכן, אין להשתמש במחלקות הקיימות של StringBuilder (או StringBuffer).

המחלקה MyStringBuilder מכילה <ArrayList<Character> כדי להחזיק את התווים של המחרוזת.

עליכם להשלים את המתודות הבאות:

- append שבהינתן מחרוזת המתודה תוסיף אותה לסוף רשימת התווים (5 נק')
- reverse שתהפוך את הסדר של התווים (5 נק')
- apply שבהינתן פונקציה מתאימה היא תחיל אותה על כל התווים (10 נק')
- toString() תחזיר String המורכבת מהתווים ברשימה (5 נק')

בנוסף מתודות אלו צריכות לתמוך ב fluent programming.

דוגמה להפעלה מתוך MainTrain3:

```
MyStringBuilder sb=new MyStringBuilder();
String s = sb.append("abc").append("def").reverse()
              .apply(c->(char)(c-('a'-'A'))).toString();
System.out.println(s); // FEDCBA
```

כפי שניתן לראות, נוכל לשרשר איזו פעולה שנרצה ובאיזה סדר שנרצה מתוך 3 המתודות לעיל. לבסוף נקרא ל toString() כדי לקבל את התוצאה הסופית.

בדוגמה שרשרנו את המחזורות "abc" ואת "def" ולכן כרגע יש "abcdef". לאחר הפעלת reverse קבלנו את התווים בסדר הפוך. ההפעלה של apply קבלה כפרמטר פונקציה שבהינתן תו היא החסירה את ההפרש בין 'a' ל 'A' ולכן הפכה את כל התווים מ lower case ל upper case.

הבדיקה במוד ההגשה מורכבת יותר וייתכנו למשל פונקציות שונות שיוזנו ל apply.

בנוסף, שאלה זו תיבדק ידנית. משקל קוד שמתקמפל ורץ ללא שגיאות ריצה הוא 8 נק'.

בהצלחה!

הגשה:

יש להגיש את Dictionary.java, Q2.java, MyStringBilder.java ואותם בלבד למערכת ההגשה בכתובת: <http://cktest.cs.colman.ac.il/> לתיבה PTM1_test ומועד א'. הקפידו שכל הקבצים עוברים קומפילציה גם אם לא עניתם על שאלה או סעיף כלשהו. בטרם היציאה מהכיתה יש להגיש במוד הגשה סופית ולהראות מס' אסמכתא בן 4 ספרות לבוחנים.