

תוכנה 1 – אביב תשע"ה

תרגיל מספר 4

חיפוי מילים וקריאה מקבצים

הנחיות כלליות:

- קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.
- הגשת התרגיל תעשה במערכת ה-moodle בלבד (<http://moodle.tau.ac.il/>).
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש aviv יקרא הקובץ aviv_hw4.zip). קובץ ה-zip יכול:
 - א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
 - ב. קבצי ה-java של התוכניות אותם התבקשתם לממש, כולל מבנה התיקיות של החבילה.

הגשת מחלקה עם חבילות: יש לכווץ בתוך קובץ ה-zip שאתם מגישים את כל היררכיית התיקיות מתחת ל-src באקליפס. למשל, כדי להגיש את המחלקה sw1.pac.MyClass העתיקו את התיקיה sw1 שמתחת ל-src כולל כל מה שבתוכה לתוך קובץ ה-zip.

הקדמה: בדיקות קלט

- החל מתרגיל זה ואילך, יהא עליכם לבצע בדיקות של תקינות הקלט המתקבל לתכנית. למשל, האם מס' הארגומנטים תקין, האם ערכי הארגומנטים תקינים וכו'. עם זאת:
- אם מצוין בסעיף כלשהו כי ניתן להניח משהו על הקלט, אין צורך לבדוק זאת בקוד של אותו סעיף (הנחת קדם).
 - בפונקציות בהן מתקבל קלט מסלול לקובץ, אין צורך לבדוק את חוקיות המסלול או אם הוא מצביע לקובץ קיים.
 - אין צורך לבדוק את מה שנאסף כבר ע"י הקומפיילר, למשל את טיפוס הארגומנטים למתודה.
- כדי להודיע למשתמש על ארגומנט לא תקין לתכנית (משורת הפקודה), נשתמש בפקודת throw באופן הבא:

```
throw new Exception("[ERROR] " + message);
```

כאשר המחרוזת [Error] תשמש אותנו להבדיל בין שגיאות שנזרקו ע"י הקוד שלכם לבין שגיאות שנזרקו ע"י מחלקה מהספריה הסטנדרטית. המחרוזת message תכיל מסר קריא המסביר את הבעיה. השגיאה תיזרק ותגרום לסיים התכנית.

כדי להודיע למשתמש על קלט לא תקין מה-console, נדפיס הודעת אזהרה למשתמש באופן הבא:

```
System.out.println("[WARNING] " + message);
```

הודעה זו תסביר למשתמש כיצד עליו לתקן את הקלט בפעם הבאה.

חלק א' – בניית אוצר מילים

צרו את המחלקה `sw1.ex4.WordSearch` בשלב הראשון, נוסף לה שירותים לקריאת רשימת מילים מקובץ.

(1) [20 נק'] נכתוב מתודה לקריאת מילים באמצעות `Scanner`. בתוך המחלקה הגדירו את השירות:
`public static String[] scanVocabulary(Scanner scanner)`

השירות יקבל כקלט עצם מסוג `java.util.Scanner` ויניח שהוא כבר מאותחל לקריאה ממקור כלשהו (למשל, קובץ). עליכם לקרוא את המילים בעזרת ה- `Scanner` ולהחזיר מערך עם כל המילים שנקראו **ממינות לקסיקוגרפית וללא חזרות**.

- נגדיר "מילה" כרצף של תווים שמופרד ע"י רווחים לבנים (whitespaces) מהרצפים האחרים. ניתן להניח שמפרדי ברירת המחדל של `Scanner` הוא לפי רווחים לבנים.
- עליכם להמיר תחילה את כל המילים שתקראו ל- `lowercase`.
- אם מילה שקראתם היא באורך 0, או מכילה תו שאינו אות בא"ב האנגלי (a-z), היא אינה "חוקית" ועליכם להתעלם ממנה. למשל, המילים "mp3" ו- "however," אינן חוקיות כי הן מכילות תו שאינו בתחום a-z.
- יש להחזיר רק את 3000 המילים ה"חוקיות" השונות הראשונות. בפרט, גודל המערך המוחזר לא יעלה על 3000.
- אם בסיום הקריאה מצאתם פחות מ- 3000 מילים שונות, גודל המערך המוחזר צריך להיות בהתאם. למשל, אם קראתם 2463 מילים, החזירו מערך בגודל 2463 שמכיל אותן.
- אין לסגור את ה-`Scanner` בתוך המתודה.
- ניתן להיעזר בשירותים של המחלקה `java.util.Arrays`.

חלק ב' – חיפוש מילים

בחלק זה נוסף למחלקה `WordSearch` יכולות חיפוש מילים באוצר המילים.

החיפוש מקבל כקלט שלושה תווים מבין a-z (ב-`lowercase`), ומוצא מילה שמכילה את שלושת התווים בסדר בו הם ניתנו, ומתחילה בראשון מביניהם. אסור שתהיה חפיפה בין התווים במילה, כלומר, בהינתן התווים e e e, על השחקן למצוא מילה שמתחילה ב-e ומכילה שני מופעים נוספים של e. לכן, `expected` היא מילה חוקית, אך `elephant` (מכילה רק שני e) ו- `seemed` (לא מתחילה ב-e) אינן חוקיות עבור הקלט הנ"ל.

(2) [40 נק'] הוסיפו ל- `WordSearch` את המתודה `printWords` בעלת החתימה הבאה:

```
public static void printWords(String[] vocabulary,  
String firstLetter, String secondLetter, String thirdLetter)
```

המתודה מקבלת כקלט את פלט השירות scanVocabulary ושלוש מחרוזות שכל אחת מהן מכילה תו a-z יחיד (ניתן להניח שהקלט תקין, אנו נבצע בדיקה זו במתודות אחרות בהמשך). המתודה מחפשת מילים מתאימות לשלושת התווים במערך הנתון, ומדפיסה כל אחת מהן בשורה נפרדת. לבסוף, היא מדפיסה את מספר המילים שנמצאו בפורמט "found %d words". לדוגמא, עבור אוצר המילים של הקובץ tale.txt הנתון והאותיות "l", "t", "l" יודפס

```
lately  
little  
found 2 words
```

עבור אותו אוצר מילים והאותיות "t", "l", "l" יודפס

```
found 0 words
```

(3) **[40 נק']** כעת, נוסיף למחלקה WordSearch מתודת main עם חיפוש אינטראקטיבי של מילים דרך ה-console (כלומר, `System.in`). התכנית תקבל כארגומנט משורת הפקודה את המסלול לקובץ קלט, תריץ את `scanVocabulary` בכדי לקבל את אוצר המילים, ותדפיס את מס' המילים שנקראו בפורמט: "Read %d words from %s\n", כאשר %s הוא מחרוזת המסלול לקובץ.

לאחר מכן תתנהל אינטראקציה עם המשתמש (ראו מצגת תרגול 4 לגבי יצירת Scanner וחיבורו ל-`System.in`) באופן הבא:

- i. בכל פעם התכנית תבקש מהמשתמש להקיש 3 תווים מ-a-z ב-console, בשורה אחת, כאשר בין כל שני תווים מופיע רווח יחיד (ראו דוגמה למטה).
- ii. התכנית תוודא את תקינות הקלט שהקיש המשתמש.
- iii. אם הקלט אינו תקין, תודפס אזהרה למשתמש בפורמט שלעיל והוא יוכל לנסות שוב.
- iv. אם הקלט תקין, התכנית תריץ עליו את `printWords` ושוב תאפשר למשתמש להקיש אותיות.
- v. התכנית תסתיים כאשר המשתמש יקיש את השורה "exit". זכרו לסגור בסוף התכנית את ה-Scanner שפתחתם.

להלן אינטראקציה לדוגמא המדגימה את ההדפסות בכל שלב. אתם רשאים לבחור בעצמכם את נוסח הודעות השגיאה / אזהרה שהתכנית שלכם תדפיס. וודאו שהודעות אלה משמעותיות ומסבירות את הבעיה למשתמש. קובץ הקלט בדוגמא למטה הוא `tale.txt`. קלט מהמשתמש מסומן בכחול.

```
Read 2463 words from tale.txt  
Enter 3 letters or "exit"  
b a a  
balanced  
breakfasted  
found 2 words  
Enter 3 letters or "exit"  
b y  
[WARNING] Expecting 3 letters  
Enter 3 letters or "exit"  
b u zz  
[WARNING] Expected a letter but got: zz  
Enter 3 letters or "exit"  
exit
```

(כאשר אתם מריצים את התכנית שלכם לצרכי בדיקה, זכרו לתת לתכנית כארגומנט מסלול לקובץ השמור על המחשב שלכם).

בהצלחה!