

תוכנה 1 – אביב תשע"ז

תרגיל מספר 4

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 יכיל:
 - א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
 - ב. קבצי ה-java של התוכניות אותם התבקשתם לממש.

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.

הערה: התרגיל מנוסח בלשון נקבה ומיועד לסטודנטיות ולסטודנטים משני המינים.

משחק מילים אינטרקטיבי

בתרגיל זה נממש משחק אינטרקטיבי המורכב משני שלבים:

- שלב ההגדרות שבו המשתמשת יוצרת חידה שהיא מילה שחלק מהאותיות שלה מוסתרות.
- שלב המשחק שבו על המשתמשת לנחש את האותיות המוסתרות עד לחשיפת המילה בשלמותה.

שימו לב, בתוכנית זאת עליכם להשתמש **במערכים בלבד**, ולא במבני נתונים גנריים כמו Lists/Maps/Sets וכו'. שימוש במבני נתונים גנריים יכול לגרום להורדת ניקוד משמעותית עד כדי ציון נכשל. בפרט, אין להשתמש ב Arrays.asList שכן שירות זה מייצר אוסף גנרי מטיפוס List. כל הקוד שלכם יכתב במחלקה il.ac.tau.cs.sw1.ex4.WordPuzzle, שלד המחלקה מצורף לתרגיל הבית.

הערה: בשלד המחלקה מופיעות הגדרות קבועים ומימושים של פונקציות המייצרות את הפלט של התוכנית. השתמשו בהם בקוד שלכם, זה יעזור לכם לשמור על פורמט פלט תקין וימנע טעויות בהגדרות קבועים.

הנחות על הקלט לפונקציות:

הניחו כי הקלט לכל הפונקציות חוקי, **אלא אם כן צויין אחרת בגוף הפונקציה ויש התייחסות מפורשת לטיפול בערכים שאינם חוקיים**. בפרט:

- אם פונקציה מקבלת מילה (מחרוזת), המילה אינה null ואינה ריקה, ומכילה רק את התווים a-z.
- אם פונקציה מקבלת חידה (מערך תווים), החידה אינה null ואינה מערך ריק.

(1) [10 נק'] בשלב הראשון, נוסף למחלקה שירות לטעינת אוצר המילים שלנו מקובץ ע"י מימוש 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 **ממוינות לקסיקוגרפית וללא חזרות**.

- נגדיר "מילה" כרצף של תווים שמופרד ע"י רווחים לבנים (whitespaces) מהרצפים האחרים. ניתן להניח שמפריד ברירת המחדל של Scanner הוא לפי רווחים לבנים.
- מילה חוקית היא מילה שאינה ריקה, וכן, שמכילה אותיות בלבד. R2-D2 היא דוגמה למילה שאינה חוקית.
- את כל המילים יש להמיר ל lowercase.
- יש להחזיר רק את 3000 המילים ה"חוקיות" השונות הראשונות, כלומר 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.

(2) [5 נק'] נוסף את השירות isInVocabulary אשר מקבל מערך מחרוזות המייצג את האוצר המילים וכן מחרוזת נוספת, ובודק אם מחרוזת זו קיימת באוצר המילים. השירות יחזיר ערך בוליאני. ממשו את המתודה isInVocabulary על פי החתימה הבאה:

```
public static boolean isInVocabulary(String[] vocabulary, String word)
```

(3) [5 נק'] חידה הינה מילה שחלק מאותיותיה מוסתרות. חידה תצוייג בתוכנית ע"י מערך של תווים (char). התא ה i במערך יכול להכיל אחד משני ערכים: אות בין a ל z או את התו "_" (השתמשו בקבוע HIDDEN_CHAR אשר הוגדר עבורכם במחלקה). אם במקום ה i מופיע התו "_", פירושו שאות זו היא מוסדרת.

חידה בעלת מבנה חוקי היא חידה שיש בה לפחות אות מוסתרת אחת ובנוסף, שכל התווים שלה הם אותיות בין a-z והתו "_".

ממשו את השירות isLegalPuzzleStructure אשר מקבל חידה ומחזיר true אם המבנה שלה חוקי, false אחרת. חתימת השירות:

```
public static boolean isLegalPuzzleStructure(char[] puzzle)
```

(4) [5 נק'] נוסף שירות אשר סופר את מספר האותיות המוסתרות בחידה puzzle כלשהי, כלומר, מספר הפעמים שהתו _ מופיע בחידה. הפונק' תחזיר 0 אם אף אות אינה מוסתרת בחידה.

חתימת השירות

```
public static int countHiddenInPuzzle(char[] puzzle)
```

(5) [10 נק'] נוסף שירות אשר בודק אם המחרוזת word מהווה פתרון חוקי לחידה puzzle. המילה word מהווה פתרון חוקי ל puzzle אם מתקיימים התנאים הבאים:

- ל word ול puzzle אורכים זהים.
 - האם האות ה i ב puzzle חשופה, היא צריכה להיות זהה לאות ה i ב word (אם המילה word מהווה פתרון חוקי לחידה, כשנחשוף את כל האותיות המוסתרות בחידה, נקבל את המילה word).
 - אם אות כלשהי מופיעה יותר מפעם אחת במילה word, כל המופעים שלה יהיו או מוסתרים או חשופים.
- לדוגמא, המילה wheeps אינה מהווה פתרון חוקי עבור החידה [w, h, e, _, p, s], שכן המופע הראשון של ה e הוא גלוי, והשני מוסתר. מילה זו כן מהווה פתרון חוקי לחידה [w, h, _, _, p, s].

ממשו את השירות checkSolution המקבל puzzle (מערך של תוים) ומילה word. השירות מחזיר true אם word מהווה פתרון חוקי ל puzzle, ו false אחרת. חתימת השירות:

```
public static boolean checkSolution(char[] puzzle, String word)
```

(6) [5 נק'] נרצה לבדוק אם לחידה יש פתרון יחיד מבין כל המילים השמורות באוצר המילים. ממשו את השירות checkSingleSolution אשר מקבל puzzle (מערך של תוים המייצג חידה בעלת מבנה חוקי) ו vocabulary (מערך של מחרוזות). אם קיים פתרון יחיד לחידה, השירות יחזיר פתרון זה. אחרת, ופרט אם אין שום פתרון לחידה, הפונקציה תחזיר null.

חתימת השירות:

```
public static String checkSingleSolution(char[] puzzle, String[] vocabulary)
```

(7) [10 נק'] נוסף שירות אשר מקבל תו (guess), מילה (solution) וחידה (puzzle). במידה וישנם מופעים מוסתרים של התו guess בחידה puzzle, נחשוף את המופעים הללו ע"י עדכון המקומות המתאימים puzzle מהתו ' _ ' לתו guess. הפונקציה תחזיר את מספר המקומות שעודכנו. למשל, עבור החידה [w, _, _, _, p], הפתרון wheep, והתו e, הפונקציה תעדכן את החידה ל [w, _, e, e, p] ותחזיר את הערך 2. אם עבור אותן חידה ומילה היינו מעבירים את התו f, האובייקט puzzle לא היה משתנה והפונקציה הייתה מחזירה את הערך 0.

ממשו את המתודה המתודה applyGuess על פי החתימה הבאה:

```
public static int applyGuess(char guess, String solution, char[] puzzle)
```

הנחות:

- guess יהיה תמיד אות חוקית בא"ב האנגלי ב lowercase.
- המילה solution מהווה פתרון חוקי לחידה puzzle
- Puzzle היא חידה בעלת מבנה חוקי (ראו הגדרה בסעיף 3)

(8) [40 נק'] כעת, נוסף למחלקה WordPuzzle מתודת main שתפעיל את המשחק האינטרקטיבי באמצעות קבלת קלט מהconsole (כלומר, `System.in`). התכנית תקבל כארגומנט משורת הפקודה את המסלול לקובץ אוצר המילים, תפעיל את השירות `scanVocabulary` בכדי לקבל את אוצר המילים, ותדפיס את מס' המילים שנקראו בפורמט: "Read DDD words from SSS", כאשר DDD הוא מספר המילים באוצר המילים שנוצר ו SSS הוא שם הקובץ כפי שהועבר ברשימת הארגומנטים. במידה וחסר ארגומנט יש להדפיס הודעת שגיאה לבחירתכם ולצאת מהתוכנית. הניחו כי אם מספר הארגומנטים תקין, שם הקובץ שהועבר הוא קובץ שאכן קיים בנתיב שהועבר, ואינו ריק.

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

```
i שלב ההגדרות:
a. התוכנית תדפיס את השורה --- Settings stage ---
b. לאחר מכן, המשתמשת תתבקש להזין את החידה. התוכנית תדפיס את ההודעה
Enter your puzzle:
i. המחרוזת הראשונה שהמשתמשת מזינה הינה החידה (החידה מודפסת כרצף
של תוים, ראו דוגמת הרצה בסוף התרגיל). עליכם להתייחס לכל התוים
שהוזנו ע"י המשתמשת כחלק מהחידה.
ii. במידה ופורמט החידה אינו חוקי (על פי ההגדרה בסעיף 3), התוכנית תדפיס
את ההודעה Illegal puzzle, try again! ותחזור לתחילת שלב b.
iii. אחרת, אם לחידה לא קיים פתרון יחיד מתוך אוצר המילים, התוכנית תדפיס
את ההודעה Puzzle doesn't have a single solution, try
again! ותחזור לתחילת שלב b.
iv. אחרת (במידה והחידה חוקית וקיים לה פתרון יחיד) התוכנית תעבור לשלב
המשחק.
ii שלב המשחק:
a. התוכנית תדפיס את השורה: --- Game stage ---
b. כל משחק מתחיל עם מספר נסיונות השווה למספר התוים החסרים בחידה + 3. כלומר,
אם בחידה מוסתרים 4 מקומות, למשתמשת יהיו סה"כ 7 נסיונות.
c. התוכנית מדפיסה את החידה, ולאחר מכן, בשורה חדשה, את ההודעה:
Enter your guess: וממתינה לקלט מהמשתמשת. ניתן להניח שהמשתמשת תזין
תו אחד ואחריו תקיש על Enter.
d. התוכנית בודקת אם האות שהוזנה מופיעה כאות מוסתרת במילה.
i. אם אות זו מופיעה כאות מוסתרת במילה:
1. התוכנית בודקת אם כל החידה פוענחה (כל האותיות גלויות). אם כן,
תודפס ההודעה
Congratulations! You solved the puzzle
והתוכנית תסתיים.
2. אחרת, התוכנית מדפיסה את ההודעה הבאה (XXX - מס' הניחושים
שנותרו) ועוברת לשלב e
Correct Guess, xxx guesses left
ii. אחרת, (האות לא מופיעה או שהיא מופיעה וגלויה), מודפסת ההודעה:
Wrong Guess, xxx guesses left
והתוכנית עוברת לשלב e.
e. במידה ונשארו עוד ניחושים (כל ניחוש מוריד 1 ממספר הנסיונות), התוכנית חוזרת
לשלב c. אחרת, מודפסת ההודעה Game over! והתוכנית מסתיימת.
```

לנוחותכם, מצורף שלד המחלקה בו תוכלו להשלים את המימוש שלכם. השלד כולל מתודות אשר מבצעות את כל ההדפסות הנדרשות בתרגיל. מומלץ להשתמש בהן על מנת לוודא שאתם שומרים על הפורמט הנדרש, אך אין זה חובה.

הנחות נוספות:

1. בשלב המשחק, ניתן להניח שהקלט מהמשתמש הוא חוקי, כלומר, בכל תור המשתמש תזין אות אנגלית אחת ב lowercase.

טסטר:

לתרגיל זה מצורפת מחלקת טסטר בשם WordPuzzleTester. מחלקה זו מיועדת לרוץ מאותו ה package של המחלקה WordPuzzle אותה אתם מגישים. הריצו את הטסטר לאחר סיום המימוש, במידה וכל הבדיקות עברו, פלט הריצה של הטסטר יהיה "done!" בלבד, אחרת יודפס מספר השגיאה.

הטסטר משתמש גם לבדיקת נכונות החתימות של המתודות שתממשו (אם המחלקה לא מתקמפלת, חסרה מתודה או שחתימת אחת המתודות לא נכונה.) וגם לבדיקה שטחית של נכונות המימוש. **אל תסתפקו בבדיקות שמופיעות בטסטר. הוסיפו בדיקות משלכם על מנת לוודא שהקוד עובד באופן תקין לכל קלט.** הטסטרים לא יבדקו כך שאין צורך (אבל זה גם לא יפריע) להגיש אותם.

להלן אינטראקציה לדוגמא המדגימה את ההדפסות בכל שלב. שימו לב לנוסחים של ההודעות שהתוכנית מדפיסה.

קובץ הקלט בדוגמא למטה הוא vocabulary.txt וניתן להניח שבהרצה זו הקובץ נמצא בתיקיה resources/hw4. קלט מהמשתמש מסומן בכחול. בתיקיית קבצי התרגיל תוכלו למצוא פלט ריצה נוסף על אוצר המילים שנבנה מתוך blackbird, בקובץ output1.out.

```
Read 15 words from resources/hw4/vocabulary.txt
--- Settings stage ---
Enter your puzzle:
  9
Illegal puzzle, try again!
Enter your puzzle:
_____
Puzzle doesn't have a single solution, try again!
Enter your puzzle:
w_i_e
--- Game stage ---
w_i_e
Enter your guess:
e
Wrong Guess, 4 guesses left
w_i_e
Enter your guess:
h
Correct Guess, 3 guesses left
whi_e
Enter your guess:
t
Wrong Guess, 2 guesses left
whi_e
Enter your guess:
l
Congratulations! You solved the puzzle
```

(כאשר אתם מריצים את התכנית שלכם לצרכי בדיקה, עליכם להעביר כארגומנטים כתובות של קבצים השמורים על המחשב שלכם).

בהצלחה!