

תוכנה 1 – אביב תשע"ד

תרגיל מספר 4

עיבוד מחרוזות 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 יכיל:
 - 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.

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 כקלט 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 קלט לא תקין או על כך שהתקבלה שגיאה, נשתמש במחלקה `ErrorHelper.util.cs.sw1.tau.ac.il`. מחלקה פשוטה זו מאפשרת להדפיס שגיאות משני סוגים:

א. שגיאות שלאחריהן התכנית ממשיכה בריצתה כרגיל. אלו שגיאות שניתן להתאושש מהן, למשל, בסיטואציה בה קיבלנו קלט לא תקין בקלט הסטנדרטי מן המשתמש, ויש באפשרותנו לבקש ממנו לתקן את הקלט. במקרה כזה נשתמש באחד מן השירותים של `ErrorHelper` באופן הבא:

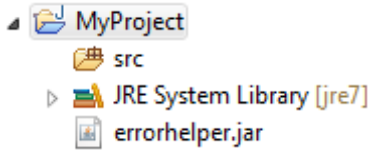
```
// will print the message to the user
ErrorHandler.notifyRecoverable("Invalid input, please try again");
:א

// will print the message only if the value of the first argument is
// false, i.e., input <=0
ErrorHandler.assertRecoverable(input > 0,
    " Invalid input, please try again ");
```

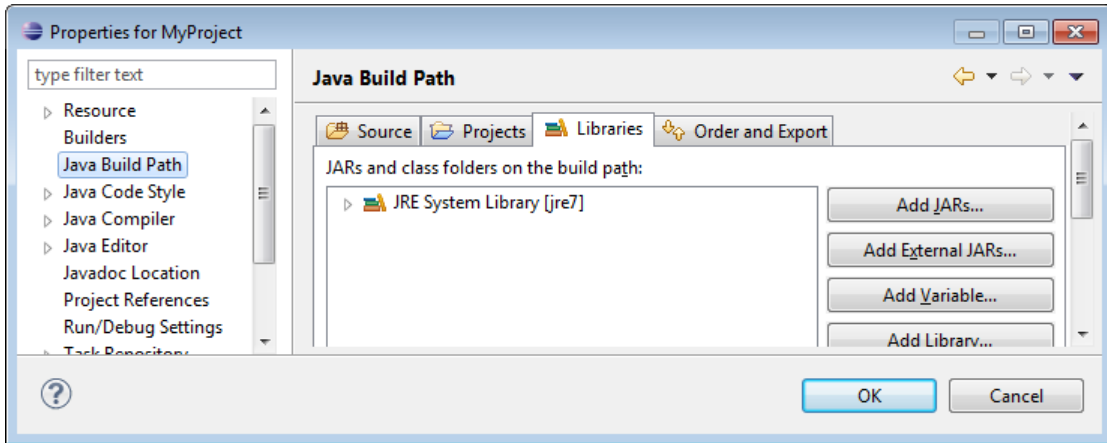
ב. שגיאות שלאחריהן נרצה שהתכנית תסתיים באופן מיידי, למשל, קלט לא תקין משורת הפקודה. במקרה כזה נשתמש ב-`ErrorHelper.notifyUnrecoverable` או ב-`ErrorHelper.assertUnrecoverable`, המקבילים לשתי המתודות הקודמות, אך במקום רק להדפיס שגיאה, גם גורמים לתעופה של התכנית.

הוספת `ErrorHelper` לפרוייקט שלכם ב-Eclipse תבצע באופן הבא:

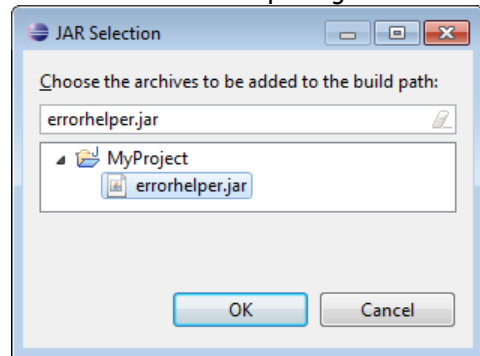
הורידו את errorhelper.jar מאתר הקורס ושימרו אותו בתוך הפרוייקט שלכם באקליפס.



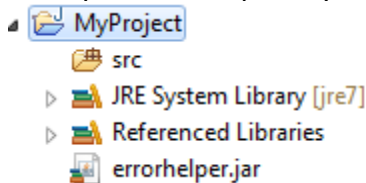
לחצו על Project > Properties > Java Build Path, ובטאב Libraries לחצו על Add JARs...



בחרו את errorhelper.jar ולחצו על OK.



לחצו שוב על OK לאישור סופי. כעת, errorhelper.jar יסומן בצלמית מיוחדת המעידה על כך שניתן להשתמש בו כספרייה חיצונית. כדי להשתמש ב- ErrorHandler יש להוסיף import מתאים בקוד.



תיעוד: באתר הקורס מופיע קישור המצביע לדפי התיעוד של ErrorHandler.

שימו לב: לצורך הבדיקות האוטומטיות עליכם להשתמש ב- ErrorHandler עבור כל שגיאה שנתקלתם בה. הקפידו להעביר ל- ErrorHandler הערות משמעותיות המסבירות מה הייתה הבעיה.

חלק א' – עיבוד מחרוזות המכילות מספרים (41%)

בחלק זה נכתוב מתודות לעיבוד מחרוזות המכילות מספרים. כל המתודות יישמרו במחלקה `.il.ac.tau.cs.sw1.ex4.StringMath`.

1. ממשו את המתודה

```
public static int getIntByRadix(String input)
```

המקבלת כקלט מחרוזת המייצגת מספר שלם אי שלילי. המתודה תדע לקרוא מספר בבסיס בינארי, דצימלי, או הקסדצימלי. מספר בינארי יסומן בעזרת התחילית 'b', למשל, "b0010" מייצג את המס' 2. מספר דצימלי יופיע ללא סימון מיוחד, למשל "345". ומספר בבסיס הקסדצימלי יסומן בעזרת התחילית 'x', למשל "x6". המתודה תחזיר את הערך המתאים כ-`int`. לדוגמא:

```
StringMath.getIntByRadix("b0010") // returns 2  
StringMath.getIntByRadix("345") // returns 345  
StringMath.getIntByRadix("x2f") // returns 47
```

מומלץ להיעזר במתודה `Integer.parseInt(String s, int radix)`. ניתן להניח כי המספרים עצמם בפורמט תקין, אך יש לבדוק את תקינות התחילית.

2. ממשו את המתודה

```
public static int calcExperssion(String exp)
```

המקבלת כקלט ביטוי חשבוני עם מספרים בייצוג המיוחד שהוגדר בסעיף הקודם, וסימני `+` ו-`*`, ומחשבת את ערך הביטוי. שימו לב לסדר פעולות חשבון. למשל,

```
StringMath.calcExperssion("4*x10+b11*x*7")
```

הקלט שקול לביטוי $4 \cdot 16 + 3 \cdot 14 \cdot 7$, ולכן התוצאה המוחזרת צריכה להיות 358. היעזרו ב-`getIntByRadix` מהסעיף הקודם. בסעיף זה ניתן להניח כי פורמט הקלט תקין, אך יש להתמודד עם מקרי קצה כגון מחרוזת ריקה.

רמז: ניתן לחלק את המחרוזת תחילה לפי `+`, אח"כ לחלק כל תת-מחרוזת לפי `*`, להמיר כל מס' ל-`int` ואז לחשב את התוצאה.

חלק ב' – ספירת מילים בטקסט (59%)

1. כתבו מחלקה בשם `il.ac.tau.cs.sw1.ex4.TextStats` שבתוכה תוגדר המתודה

```
public static void printStats(Scanner scanner, int x);
```

המקבלת כארגומנט `Scanner` (שכבר מחובר למקור כלשהו, למשל קובץ) ומספר שלם `x`, כאשר $0 < x \leq 3000$. המתודה תקרא מן ה-`Scanner`, תנתח את המילים המופיעות בו על פי המוגדר בהמשך, ותדפיס למסך את תוצאות הניתוח.

- המתודה תדפיס את `x` המילים בעלות מספר המופעים הגדול ביותר בקובץ הטקסט, ואת מספר המופעים שלהן (מופרדים ב- `'\t'`)
- המתודה תדפיס את `x` המילים הארוכות ביותר בקובץ הטקסט, ואת אורכן (מופרדים ב- `'\t'`)

הערות:

- מילים מוגדרות כרצף תווים (שאינם `Whitespaces`) המופרדות ביניהם ע"י `Whitespaces`.
- (אוסף התווים שנחשבים כ-`Whitespace` נקבע לפי המתודה `Character.IsWhitespace`).
- יש להתעלם ממילים בלתי חוקיות המכילות תו שאינו אלפאביתי (אינו בין `a` ל-`z` או בין `A` ל-`Z`).
- יש להמיר תחילה את כל המילים ל-`lowercase`.
- ניתן לכלול בחישובים רק את 3000 המילים השונות הראשונות אותן נקרא מקובץ הקלט (רמז: כדי שנוכל להגדיר מראש מערך מילים ומערך מונים בגודל זה).
- המילים בפלט (או האותיות בסעיף השני) צריכות להיות ממוינות קודם לפי מספר המופעים \ האורך, ואחר כך לפי סדר הופעתן לראשונה בטקסט. בכל מקרה יש לכלול לכל היותר `x` תוצאות בכל סעיף, גם אם נמצאו מספר מילים עם מספר זה של מופעים \ אורך זהה.
- כאשר הקלט מכיל רק `y` מילים, $y < x$, נדפיס רק `y` מילים.
- הקפידו לבדוק את תקינות המס' `x` (ניתן להניח שה-`Scanner` תקין)
- לצורך קריאת הטקסט מקובץ הקלט וחלוקתו למילים ניתן להשתמש במחלקה `Scanner`. אין לסגור את ה-`Scanner` בסוף ריצת המתודה; יש להניח כי הוא ייסגר ע"י הלקוח שקרא למתודה.

דוגמא:

הרצת המתודה:

```
TextStats.printStats(scanner, 5);
```

כאשר ה-`Scanner` מחובר לקובץ `input.txt` המכיל את הטקסט הבא:

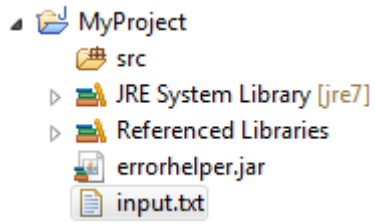
Everything that can be counted does not necessarily count; everything that counts cannot necessarily be counted. (Albert Einstein)

תביא להדפסת הפלט הבא למסך:

Word	Frequency
everything	2
that	2
be	2
necessarily	2
can	1

Word	Length
necessarily	11
everything	10
counted	7
counts	6
cannot	6

2. הוסיפו פונקציית main אשר תקבל בשורת הפקודה מסלול לקובץ ומספר, ותקרא ל- printStats.
לדוגמא, אם נקרא לתכנית עם הקלט
input.txt 5
והקובץ input.txt הנזכר בדוגמא למעלה שמור ישירות תחת תיקיית הפרוייקט שלנו ב-Eclipse,
כך:



התכנית תדפיס את אותו הפלט כמו בדוגמא למעלה.
○ הקפידו לבדוק את תקינות הקלט - מספר הארגומנטים, ואת תקינות x. לפי ההנחיות בתחילת התרגיל, אין צורך לבדוק שהמסלול תקין או מצביע לקובץ קיים; ואין צורך לבצע שוב בדיקות תקינות שמבוצעות ממילא ע"י printStats.