תוכנה 1 – סתיו תשע"ד

תרגיל מספר 8

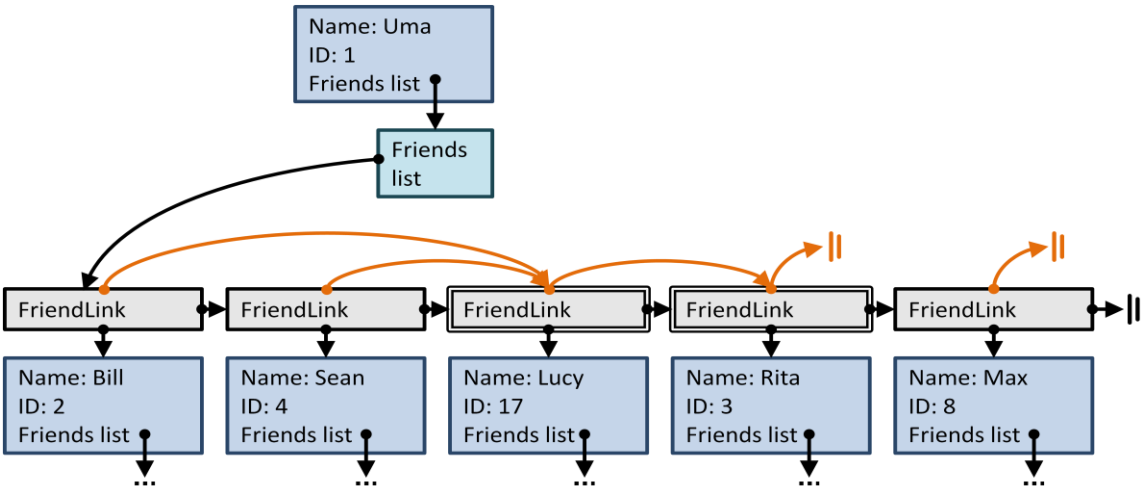
אוספים Collections

<u>הנחיות כלליות:</u>
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_hw8.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 תיקיות החבילה.

<u>חלק א' (50%): רשימות מקושרות</u>

בחלק זה של התרגיל נכתוב מחלקות לניהול רשימת חברים ברשת חברתית. ברשת שלנו, לכל משתמש יש שם, מזהה ייחודי, וחברים המופיעים ברשימה מקושרת לפי סדר הוספתם כחברים. ניתן להגדיר חלק מן החברים ברשימה כ"חברים קרובים". מופעי המחלקה `SocialNetworkUser` ייצגו משתמשים ברשת החברתית. לכל משתמש תהיה רשימת חברים מטיפוס `FriendsList`. הרשימה המקושרת הזו תורכב מאיברים מטיפוס `FriendLink`, וכל איבר כזה יכיל מצביע ל- `SocialNetworkUser` הרלוונטי.
להלן תרשים סכמטי המתאר <u>חלק</u> מן הקשרים בין המחלקות. קשרים אלה יוסברו בהמשך.
<u>שימו לב:</u> לא כל המצביעים הדרושים מתוארים בתרשים, וכנראה שתרצו להגדיר מצביעים נוספים.



בצעו את המשימות הבאות. אתם רשאים להוסיף חוזים, שדות, מתודות עזר **<u>פרטיות בלבד</u>**, ומחלקות עזר לפי הצורך, אך שמרו על שמות מחלקות וחתימות של מתודות נתונות, **<u>ואל תשתמשו באוספים קיימים מהספרייה הסטנדרטית</u>** - עליכם לממש את המבנה המקושר בעצמכם.

1. השלימו את מימוש המחלקה `FriendLink`. כל עצם מטיפוס זה יכיל מצביע לעצם מטיפוס `SocialNetworkUser`, <u>ושני מצביעים נוספים</u>: מצביע לחוליה (לינק) של החבר העוקב ברשימה, ומצביע לחוליה של החבר הקרוב הבא בהמשך רשימה. אם אין חבר\חבר קרוב בהמשך הרשימה, המצביע המתאים יפנה ל- `null`. כאשר החבר העוקב ברשימה הוא גם חבר קרוב, שני המצביעים יפנו לאותו החבר.

התרשים למעלה מתאר את רשימת חמשת החברים של אומה (לכל משתמש יש רשימת חברים משלו). לוסי וריטה הן החברות הקרובות היחידות של אומה (חוליות ה- `FriendLink` המתאימות להן בתרשים מסומנות בקו מתאר כפול). כל `FriendLink` מכיל מצביע לזה שאחריו ברשימה (חץ שחור), אך גם לחוליה של חבר הקרוב הבא (חץ כתום). למשל, במקרה של ביל ושון, החברה הקרובה הבאה ברשימה היא לוסי, ואחרי ריטה ומקס אין חבר\ה קרוב\ה ועל כן מצביע החבר הקרוב של החוליה שלהם שווה ל- `null` (מסומן בתרשים בשני קווים).

2. השלימו את מימוש המחלקה `FriendsList`. מחלקה זו תייצג רשימה מקושרת של עצמים מסוג `FriendLink`, ותאפשר גישה להתחלת ולסוף הרשימה. חישבו מה יהיה הייצוג הפנימי של מחלקה זו (בתרשים הסכמטי מתואר מצביע לתחילת הרשימה, אך אתם יכולים להשתמש בשדות נוספים). בהוספת חבר הרשימה תדאג **<u>לא ליצור כפילויות</u>**, וכן ש**<u>חברות היא הדדית</u>** - כאשר ביל מתווסף לרשימת החברים של אומה, אומה צריכה להתווסף לרשימת החברים של ביל; אם אומה חברה קרובה של ביל אז ביל חבר קרוב של אומה.

```
/**
 * Returns the user that this friends list belongs to
*/
public SocialNetworkUser getUser()

/**
* Returns the first link in the list
*/
public FriendLink getFirstLink()

/**
 * Returns the friend that was added first to the friends list
 */
public SocialNetworkUser getOldestFriend()

/**
 * Returns the oldest friend that is also a close one
 * (this friend didn't necessarily become close first)
 */
public SocialNetworkUser getOldestCloseFriend()

/**
 * Returns the number of friends in the list
 */
public int getNumFriends()

/**
 * Returns the number of close friends in the list
 */
public int getNumCloseFriends()

/**
 * @pre user != null
 * @pre user != getUser()
 *
 * If user is already a friend, nothing happens.
 * Otherwise, the user is added as a friend to the end of the list.
 * Add this.getUser() to the friends list of user
 * Make sure you don't create an endless loop!
 */
public void friend(SocialNetworkUser user)
```

```
/**
 * If user is not in the friends list, nothing happens.
 * Otherwise, find user in the friends list, make him a close friend and
 * update all the relevant fields and pointers of other friends in the
 * list accordingly.
 * Make this.getUser() a close friend in the list of user.
 */
public void upgradeToCloseFriend(SocialNetworkUser user)
```

3. השלימו את המחלקה `SocialNetworkUser`.

```
/**
 * @pre name!=null
 * @pre name.length() > 0
 * @post getName() == name
 * @post getId() returns an ID unique for this user
 * @post getFriendsList() != null
 * @post getFriendsList().getNumFriends() == 0
 */
public SocialNetworkUser(String name)

public String getName()

public int getId()

public FriendsList getFriendsList()

/**
 * Returns the user details in the following format:
 * "Name: <name>, ID: <ID>, #friends: <num_friends> (<num_close_friends>
        close)"
 * For example,
 * Name: Uma, ID: 1, #friends: 2 (1 close)
 */
public String toString()
```

4. הוסיפו ל- `FriendsList` מתודה המאפשרת הדפסה של החברים ברשימה עם פרטיהם.

```
/**
 * Prints the friends list to System.out.
 * - denotes a "regular" friend and + denotes a close friend
 * Example for the format:
 * #friends: 2 (1 close)
 * - Name: Bill, ID: 2, #friends: 1 (0 close)
 * + Name: Sean, ID: 4, #friends: 1 (1 close)
 */
public void printFriends()
```

מצורפת תכנית בדיקה בשם `SocialNetworkTest`. פונקצית ה-`main` של התכנית יוצרת תחילה את המשתמשים אומה, ביל ושון. לאחר מכן היא מוסיפה את ביל ואז את שון כחברים של אומה. אז, שון יהפוך להיות חבר קרוב של אומה. כך פלט התכנית אמור להיראות (עד כדי שינוי `IDs` של המשתמשים):

```
>>>>>>> Start
Uma:
#friends: 0 (0 close)
```

```
Bill:
#friends: 0 (0 close)

Sean:
#friends: 0 (0 close)


>>>>>>> After adding friends
Uma:
#friends: 2 (0 close)
- Name: Bill, ID: 2, #friends: 1 (0 close)
- Name: Sean, ID: 3, #friends: 1 (0 close)

Bill:
#friends: 1 (0 close)
- Name: Uma, ID: 1, #friends: 2 (0 close)

Sean:
#friends: 1 (0 close)
- Name: Uma, ID: 1, #friends: 2 (0 close)


>>>>>>> After upgrade
Uma:
#friends: 2 (1 close)
- Name: Bill, ID: 2, #friends: 1 (0 close)
+ Name: Sean, ID: 3, #friends: 1 (1 close)

Bill:
#friends: 1 (0 close)
- Name: Uma, ID: 1, #friends: 2 (1 close)

Sean:
#friends: 1 (1 close)
+ Name: Uma, ID: 1, #friends: 2 (1 close)
```

הערה: בד"כ לא מקובל לחשוף החוצה את המבנה הפנימי של רשימה מקושרת, כפי שעשינו כאן עם `FriendLink`. לפי עקרונות ההכמסה (`encapsulation`), היינו צריכים לממש אותה, למשל, כמחלקה מקוננת פרטית. כאן נתבקשתם לאפשר גישה לחוליות הרשימה לצורכי בדיקת התרגיל.

חלק ב' (50%): Java collection framework

בחלק זה נתרגל עבודה עם אוספים (Collections) ע"י מימוש מנוע חיפוש פשוט. בין קבצי העזר של התרגיל תוכלו למצוא את המחלקות HTMLTokenizer, SimpleSearchEngine ואת המנשק WordIndex שכבר נכתבו עבורכם.

מנוע החיפוש שלנו יטפל במספר מצומצם של דפי HTML אותם הוא יקרא מהרשת. דפים אלו מוגדרים מראש ורשימתם נמצאת במחלקה SimpleSearchEngine. אם תרצו תוכלו להוסיף או לשנות את הדפים בהם אתם משתמשים.

לאחר שהורדנו דף HTML מהרשת נתייחס רק לחלק הטקסט שבדף ונפרק חלק זה למילים בודדות. קוד זה כבר מומש עבורכם במחלקה HTMLTokenizer.

בנוסף, נתונה המחלקה SimpleSearchEngine שבה הקוד המפעיל את המערכת ומתקשר עם המשתמש. הקוד במחלקה זו קורא בתחילה למתודה index אשר תאכלס את אינדקס המילים. לאחר מכן, כתלות בקלט המשתמש יתבצע חיפוש של דפים לפי מילת שאילתה, או חיפוש של דפים לפי דמיון לכתובת דף נתון (במקרה זה יש להכניס את כתובת דף השאילתה לאחר הסימן ~, ראו דוגמא בהמשך).

מה עליכם לעשות:

נרצה ליצור אינדקס של כל המילים שהופיעו בכל הדפים שהורדנו מהרשת. קיומו של אינדקס זה יאפשר לנו מאוחר יותר לבצע חיפושים עבור מילה מסוימת. עליכם לממש מחלקה בשם MyWordIndex המממשת את המנשק WordIndex. מחלקה זו שומרת את אינדקס המילים, מאפשרת הוספת מילים לאינדקס ולאחר מכן מאפשרת גם חיפוש באינדקס.

```
/**
 * A word index allows for a quick search over a large collection of
 * web pages.
 */
public interface WordIndex {

   /**
    * Add the words originating in the specifies URL.
    *
    * @param words
    *            - collection of words to add to the index
    * @param strURL
    *            - the location of the page containing the words
    */
   void index(Collection<String> words, String strURL);

   /**
    * Search for pages containing a given word in the index
    *
    * @param word
    *            - the word to search
    * @return A list of pages containing the word. The pages are ordered
    *         according to the relative importance of the word within them.
    */
   List<String> search(String word);
```

```
    /**
     * Search for pages that are similar to a given URL in the index
     *
     * @param strURL
     *            - a URL, possibly not in the index
     * @return The page in the index that is the most similar to the input
     *         strURL.
     */
    String findSimilarPage(String strURL);
}
```

הערה: בקוד למעלה התיעוד כתוב בסגנון javadocs: התגית @param משמשת להסבר על פרמטר של המתודה, ו-@return משמשת להסבר על ערך ההחזרה. לא מדובר בחוזה!

המתודות השונות:

- **המתודה `index`**
מתודה זו אחראית על אכלוס מבנה הנתונים שלכם. המתודה מקבלת אוסף של מילים (ייתכנו חזרות) ואת כתובת האינטרנט (URL) של הדף מהן הגיעו. עליכם לבחור את מבני הנתונים שבהם תשתמשו ולדאוג לשמור על הקשרים הבאים:
  - לכל מילה באילו דפים היא מופיעה וכמה פעמים בכל דף
  - לכל דף כמה מילים בסה"כ מופיעות בו (עם חזרות)
  - לכל דף אילו מילים ייחודיות (שונות) מופיעות בו.

  הערה: רשימת המילים בקלט היא כפי שהופיעה בדף המקורי, אולם יש לשמור גרסת `lowercase` של המילים.

- **המתודה `search`**
מקבלת מילה לחיפוש ומחזירה רשימה ממוינת של כתובות אינטרנט בהן המילה מופיעה. נמיין את הרשימה כך שככל שהמשקל היחסי של מילה בדף גבוה יותר כך הכתובת תופיע במקום גבוה יותר ברשימה.
המשקל היחסי של מילה בדף יהיה מספר המופעים של המילה בדף חלקי מספר המילים הכולל באותו דף (עם חזרות). לא נחזיר דפים בהם המילה אינה מופיעה כלל.

  שימו לב: כדי להשוות טיפוסים פרימיטיביים ניתן להשתמש בשירות `compare` של הטיפוס העוטף. לדוגמא: `Double.compare(double d1, double d2)`

- המתודה **`findSimilarPage`**
מקבלת כתובת של דף אינטרנט (URL) ומחזירה את כתובת האתר באינדקס הדומה ביותר לאתר הנתון. לצורך חישוב רמת הדמיון בין שני דפי אינטרנט, נשתמש ב-Jaccard Index (http://en.wikipedia.org/wiki/Jaccard_index) המחושב כך: נחשב את קבוצת המילים הייחודיות בכל אחד מהדפים אותם נרצה להשוות, ואז נחלק את גודל חיתוך הקבוצות בגודל איחוד הקבוצות.

  בנוסחא הבאה, A עשויה להיות קבוצת המילים הייחודיות של דף השאילתה, ו-B קבוצת המילים הייחודיות של דף אחר אליו נרצה לחשב רמת דמיון:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

  שימו לב: בתוצאות החיפוש שיוחזרו ע"י המתודה יכללו רק דפים שחיתוך המילים המשותפות להם ולדף השאילתה מכיל לפחות 100 מילים (כדי לא להציג דפים שבהם יש חיתוך נמוך מאוד של מילים לדף השאילתה).

<u>הדרכה:</u>

השתמשו במחלקות מ-Java Collections שנלמדו בכתה. **שימו לב למתודות retainAll ו-addAll המאפשרות חישוב חיתוך ואיחוד של אוספים (בהתאמה).**

את מיון הרשימה של כתובות האינטרנט בצעו בעזרת הפונקציה Collections.sort(...). שימו לב שה"מיון הטבעי" של הכתובות (String) הוא מיון לקסיקוגרפי ולא כפי שמתבקש בשאלה. כדי לשנות את שיטת המיון עליכם לכתוב מחלקת עזר המממשת את המנשק Comparator (מומלץ לחפש באינטרנט דוגמאות למימוש מנשק זה). מחלקה זו תיעזר בנתונים שנשמרו באינדקס לצורך מיון הכתובות. שימו לב שתוצאות ההשוואה תלויות במילת החיפוש הנוכחית.

הערה: המילים המתקבלות מדף ה-HTML אינן "נקיות" (כוללות סימני פיסוק וכדומה). אין צורך לבצע כל פעולה על המילים מלבד הפיכתן ל-Lowercase.

כדי לבדוק את התכנית שלכם השתמשו במחלקה Main המייצרת אובייקט חדש מטיפוס SimpleSearchEngine ומעבירה לו בבנאי מופע של המחלקה MyWordIndex שמימשתם. לאחר יצירת האובייקט נקרא השירות run.

הנחיות:

- נתון לכם הקוד של הקבצים הבאים: Main.java, SimpleSearchEngine.java, WordIndex.java, HTMLTokenizer.
- עליכם לכתוב ולהגיש את המחלקה MyWordIndex שממשת את המנשק WordIndex. ניתן להוסיף מחלקות עזר באותו קובץ או בקבצים נוספים.
- אין לשנות את הקבצים הנתונים.
- לקבלת מלוא הנקודות על התרגיל יש להשתמש במבני הנתונים המתאימים לבעיה ובאופן ראוי (לא מספיק שהפלט יהיה נכון).

<u>דוגמא (בהתבסס על רשימת הכתובות המקודדת בקוד שניתן לכם)</u>:

```
Indexing "http://en.wikipedia.org/wiki/Java_(programming_language)" ...
Indexing "http://en.wikipedia.org/wiki/Java" ...
Indexing
"http://docs.oracle.com/javase/7/docs/technotes/guides/collections/reference.html" ...
Indexing "http://www.gutenberg.org/files/1342/1342-h/1342-h.htm" ...
Indexing "http://www.gutenberg.org/files/1400/1400-h/1400-h.htm" ...
Indexing "http://www.gutenberg.org/files/76/76-h/76-h.htm" ...
Indexing "http://www.gutenberg.org/files/863/863-h/863-h.htm" ...
> hello
Searching for pages containing "hello"...
1.      http://en.wikipedia.org/wiki/Java_(programming_language)
> world
Searching for pages containing "world"...
1.      http://en.wikipedia.org/wiki/Java_(programming_language)
2.      http://en.wikipedia.org/wiki/Java
3.      http://www.gutenberg.org/files/76/76-h/76-h.htm
4.      http://www.gutenberg.org/files/1342/1342-h/1342-h.htm
5.      http://www.gutenberg.org/files/1400/1400-h/1400-h.htm
6.      http://www.gutenberg.org/files/863/863-h/863-h.htm
> bye
Searching for pages containing "bye"...
       Unable to find relevant pages.
```

```
> ~http://www.gutenberg.org/files/161/161-h/161-h.htm
Searching for a page similar to URL "http://www.gutenberg.org/files/161/161-h/161-
h.htm"...
       http://www.gutenberg.org/files/1342/1342-h/1342-h.htm
> ~http://fr.wikipedia.org/wiki/Java_%28langage%29
Searching for a page similar to URL
"http://fr.wikipedia.org/wiki/Java_%28langage%29"...
       http://en.wikipedia.org/wiki/Java_(programming_language)
> (exit)
Thank you for using SimpleSearch
```

שימו לב שהתוצאות עשויות להשתנות כתלות בהתעדכנות האתרים שנסרקו. אתם מוזמנים להשוות את הפלט עם חברים.

בהצלחה !