

# היכון מונחה עצמים (10119):

## תרגיל בית להגשת מס' 1. (אכילון ברקת)

סמסטר א', תשע"ט, 2019.

### Files, Collections, ListIterator

תאריך חלוקת התרגיל בכיתה: יום ראשון 04/11/18.  
התאריך להגשת התרגיל: עד יום ראשון 25/11/18. (כולל) (עד השעה 23:45).

יש להגיש אך ורק דרך תפריט המטלות שבאתר הקורס, כפי שהוסבר בתרגול.  
שם הקובץ המוגש יכול את שם המגיש ויהיה אך ורק לפי הפורמט הבא: למשל:  
עבור תרגיל בית מס' 1: `HW1_AviLevi.java`. עבור תרגיל בית מס' 2: `HW2_AviLevi.java`, וכך הלאה.  
שם המחלקה יהיה אף הוא בהתאם. למשל: `public class HW1_AviLevi{`.  
בנוסף, הקובץ הראשי יכול בשורה הראשונה למעלה, הערה ובה השם המלא של המגיש/ה.  
אם העבודה מכילה רק קובץ אחד, יש להגיש את קובץ ה-`java` היחיד הזה.  
אחרת, יש לזפזף ולהגיש קובץ `zip`, **שיכיל רק קבצי Java**, ולא פרויקט של `Eclipse` וכו'.  
בכל מקרה אין להגיש באתר יותר מקובץ אחד.  
על הסטודנט חלה האחריות שהקובץ שהוא הגיש תקין, נשלח בצורה נכונה ומתאים לנדרש.  
על התרגילים להיפתח בהצלחה בתוכנת ה-`Eclipse`, ללא שגיאות הידור או אזהרות.  
יש לשלב בקוד הערות ותיעוד מתאים. (באנגלית בלבד!)  
יש לבצע הזחה של הקוד כנדרש.  
**יש להקפיד על כללי הנדסת התוכנה ככל הניתן**: קוד קצר, לא מסורבל ויעיל הן מבחינת כתיבתו והן מבחינת ריצת התוכנית. לא להשתמש במשתנים סטטיים שלא לצורך, תוכנית כללית שניתנת בקלות לשינויים והרחבות בעתיד, שימוש בקבועים, חלוקה מתאימה לפונקציות, פונקציות עצמאיות (כלומר שאינן תלויות בקוד /משתנה חיצוני), וכיוצ"ב. בנוסף, פתרון התרגיל צריך להיות גם כללי לכל שינוי של הנתונים, שינוי הגדלים של המערכים באם קיימים, וכו'.  
כאשר הבדיקה תסתיים, תהיה על כך הודעה באתר. **עד שאין את ההודעה, יש עוד עבודות שלא נבדקו.**  
**לא יבדקו תרגילים שמוגשים באיחור ו/או שאינם עומדים בדרישות הנ"ל**

## המטלה הנדרשת

1. נעבוד עם התוכנית של AddressBookJavaFx שנלמדה בשיעור.  
(נמצאת בתיקיית הקבצים של פרק 19.)
2. בחלונית, יש להוסיף שלושה כפתורים ולשנות ממדים של חלק מהתיבות :

3. בנוסף, בקובץ המקורי של AddressBookJavaFx, יש לשנות את ההגדרה  
`public final static int STATE_SIZE = 2;`

לגודל של 10 (במקום 2). אין לשנות קבועי final אחרים מהקובץ המקורי.

4. לצורך הדוגמא יש להזין את הרשומות הבאות לספר הכתובות, ואת הקובץ שנוצר **במצב ראשוני זה** יש לצרף לקובץ ה- zip שאתה מגיש :

Name	Street	City	State	Zip
Kinder	K	Tel-Aviv	Israel	5555
<del>Bamba</del>	B	Tel-Aviv	Israel	2222
Kinder	K	Tel-Aviv		5555
<del>Bamba</del>	B	Tel-Aviv	Israel	2223
Kinder	K	Tel-Aviv		5556
<del>Bisli</del>	B	Tel-Aviv	Israel	2222
<del>Bisli</del>	B	Tel-Aviv	Israel	2223
<del>Bisli</del>	B			1111
<del>Tapuchips</del>	AA	Tel-Aviv	Israel	99

(בסעיף 2 מופיעה הרשומה האחרונה בספר הכתובות.)

5. כפתור Sort1: יבצע מיון של השמות בסדר עולה. (אם יש שמות זהים אין חשיבות למיון ביניהם.)

(כלומר, כעת אחת מ-ה Bamba צריכה כעת להיות ראשונה בספר הכתובות.)

המיון יתבצע על גבי רשומות הקובץ ממש. כלומר אין לקרוא את תוכן הקובץ לתוך מערך או Collection כלשהו או משהו דומה, למיין אותו ואז להחזיר לקובץ.

הכוונה היא שהמיון יתבצע על הקובץ עצמו - יש להחליף בין רשומות בתוך הקובץ וכו'.

לצורך המיון, יש להתאים את אחד המיונים הבאים כרצונך: בועות, הוספה, בחירה, מהיר.

(מותר להשתמש בשני משתנים זמניים לצורך שמירה / עבודה כל פעם עם 2 רשומות כלשהן שאותן קוראים מהקובץ לצורך השוואה בהתאם למיון.)

## יש להשתמש ב- Comparator בהתאם.

6. כפתור Sort2: יבצע מיון לפי ה- Zip בסדר עולה. (אם יש Zip שהם זהים אין חשיבות למיון ביניהם).

ניתן להניח ש- ה- Zip שרשום הוא תמיד מספר שלם וחיובי, ואת זה אין צורך לבדוק או לוודא.

בכל מקרה, שים לב שכאשר אתה קורא את ה- Zip מתיבת ה- טקסט הוא עדיין String, וגם בקובץ הוא נשמר כ- String כמו בתוכנית המקורית.

אבל המיון עצמו צריך למיין כאמור לפי **הערכים המספריים** של ה- Zip. (ולא מיון של מחרוזות).

**גם כאן, המיון יתבצע על רשומות הקובץ כמתואר קודם.**

**יש להשתמש ב- Comparator בהתאם, ואין לכתוב שוב את פונקציית המיון עצמה.**

(כלומר, כעת הרשומה של Tapuchips צריכה להיות ראשונה בספר הכתובות, כי 99 זה המספר הכי קטן של ה- Zip).

7. כעת יש לממש ListIterator שיפעל על **הקובץ**. יש לממש את כל הפעולות לפי **המקובל וכמוגדר בממשק**.

בדומה למה שהיה קודם עם המיון, גם הפעם עם ה- ListIterator, מימוש הפעולות של ה- ListIterator יתבצעו על הרשומות **רק בקובץ עצמו**, כלומר ללא שימוש במערך עזר, Collection כלשהו או משהו דומה.

הערה: רק בשתי המתודות add ו- remove כן מותר מערך עזר או ArrayList וכו'.

הגישה לאיטרטור תהיה בעזרת מתודות listIterator כמקובל.

8. כפתור Iter: הבעיה היא, שפה ושם היו ברשומות עדכונים של ה- Zip, אבל מי שעדכן שכח למחוק את הרשומות

הקודמות עם ה- Zip הישן. למשל: בדוגמא למעלה, אפשר לראות שהרשומה עם Bamba בהתחלה היה לה Zip

של 2222. לאחר מכן, אותה רשומה בדיוק מופיעה עם Zip חדש של 2223.

כדי לתקן את הקובץ, נעבוד באופן הבא:

**בעזרת ה- ListIterator על הקובץ שבנית**, יש לצור אובייקט של ה- ListIterator, בשם lit.

בעזרת lit, יש לרוץ על הקובץ ולהעתיק את **כל** רשומות הקובץ לתוך **Map** כלשהי מתאימה, באופן כזה ש- ה-

Map עצמה תשמור בסופו של דבר רק את הרשומות שבהן יש את העדכון **האחרון** של ה- Zip. הרשומות

המתאימות יישמרו לפי הסדר.

כלומר, ה- Map לא תשמור רשומות קודמות שבהן **כל השדות** הם בדיוק אותו דבר מלבד ה- Zip, אלא רק

רשומה אחרונה כל פעם עם ה- Zip הכי אחרון שיש.

בשלב הזה, **בעזרת אותו האובייקט lit הנ"ל (אין לבנות אובייקט ListIterator כזה חדש)**, יש להעתיק את כל

הרשומות מ- ה- Map אל הקובץ, כך שהקובץ יכיל כעת רק את הרשומות החדשות.

(כלומר עדכון הקובץ יתבצע **רק** ע"י עבודה עם המתודות של lit, כגון add ו- remove וכך הלאה).

אם נלחץ על הכפתור Iter **מהמצב התחלתי של רשומות המתואר למעלה**, להלן מצב הרשומות שצריך לקבל

בספר הכתובות:

Name	Street	City	State	Zip
Kinder	K	Tel-Aviv	Israel	5555
Bamba	B	Tel-Aviv	Israel	2223
Kinder	K	Tel-Aviv		5556
Bisli	B	Tel-Aviv	Israel	2223
Bisli	B			1111
Tapuchips	AA	Tel-Aviv	Israel	99

לדוגמא: ה- Kinder הראשון עם Israel 5555 נשאר כמו שהוא כי שני ה- Kinder שאחריו (בטבלה המקורית) אין להם מדינה Israel כמוהו, ולכן הוא רשומה 'אחרונה' מסוגה.

ה- Kinder בטבלה המקורית עם 5555 ובלי מדינה, חוזר שם על עצמו בדיוק, אבל רק עם Zip של 5556, ולכן הרשומה האחרונה הזו עם ה- 5556 היא זו שמופיעה לבסוף.

כפתור Iter - לחיצה **שנייה ואילך**: כאשר אני לוחץ עליו שוב, כלומר החל מהלחיצה השנייה ואילך, יש להעתיק את הרשומות מה- Map לתוך TreeSet באופן כזה שנקבל כעת ב- TreeSet מיון בסדר עולה לפי שם הרחוב. ה- Tree לא ימחק אף רשומה שהייתה ב- Map, אלא רק ישמור הכול מחדש עם המיון. (אם יש רחובות זהים אין חשיבות למיון ביניהם.)

כעת, **בעזרת אותו האובייקט lit הנ"ל (אין לבנות אובייקט ListIterator כזה חדש)**, יש להעתיק את כל

הרשומות מ- ה- Tree אל הקובץ, כך שהקובץ יכיל כעת רק את הרשומות המעודכנות.

(כלומר, עבור הדוגמא הנ"ל, כעת Tapuchips יהיה הרשומה הראשונה של ספר הכתובות.)

### הערות לתוכנית

9. שים לב שכל פעולה כגון מיון וכו' מתחילה מהמצב האחרון שאליו הגיע הקובץ בהתאם לפעולה האחרונה שעשינו.
10. שאר הכפתורים בספר הכתובות ימשיכו לעבוד כרגיל.
11. בנושא של לכידת חריגות בעבודה עם קבצים ניתן לעבוד בדומה למה שמופיע בתוכנית המקורית ובאופן חריג אין צורך לסגור את הקובץ. (הוא נסגר כאשר סוגרים את החלונית.)
12. בתוכנית המקורית יש שימוש בתבנית העיצוב Command. לא חובה להשתמש בה וניתן לשנות את התוכנית כך שתעבוד בלעדיה.

**בהצלחה!**