

# תוכנה 1 – אביב 2024

## תרגיל מספר 6

### iterators ,collection framework

#### הנחיות כלליות:

קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.

את התרגיל הבא צריך להגיש באופן הבא:

- הגשה במערכת ה-Git תבצע על פי ההנחיות שראיתם בתרגול 1. צרו את ה repository שלכם מתוך הקישור הבא:

<https://classroom.github.com/a/AA12R7Rh>

יש לוודא שבתיקיית הגיט שלכם נמצאים הקבצים הבאים:

- קובץ פרטים אישיים בשם details.txt המכיל את שם המשתמש שלכם ב Moodle ואת מספר תעודת הזהות שלכם.
- 2 תיקיות partA, partB להגיש את ה repository - ואת מבנה התיקיות שבתוכו בדיוק באותה היררכיה שקיבלת אותם

- הגשה במערכת ה Moodle (<http://moodle.tau.ac.il/>): עליכם להגיש את קובץ הטקסט assignment.txt ובו קישור ל repository git האישי שלכם.

#### ייבוא הפרויקט:

מומלץ ליצור 2 פרויקטים שונים. אחד עבור החלק הראשון ואחד עבור החלק השני.

ברפוסטורי שקיבלתם יש 2 תיקיות מתאימות עם השמות partA, partB, ובכל אחת מהן תיקיית src.

עבור כל אחד מהפרויקטים קחו את תיקיית ה-src המתאימה, כאשר הנחיות יבוא הפרויקט הן אותן ההנחיות כמו בתרגילים הקודמים.

---

## חלק א' (80 נק')

בחלק זה של התרגיל זה תעבדו על תוספת קוד שכולל אוספים גנריים למשחק הפקמן שלנו. התרגיל כולל שימוש ב: ירושה, אוספים גנריים וכן מימוש של איטרטורים. תרגיל זה מתבסס על תרגיל 4 וכן על המשימה שעשיתם בכיתה (תוספת בונוס דובדבן). אנו מספקים כאן פתרון בסיסי למשימות אלו אך ממליצים לכם בחום להשתמש בפתרונות הקודמים שלכם (להדביק את הקבצים שלכם בתוך התיקייה, שימו לב להערה המודגשת המופיעה מטה).

0. ודאו שביצעתם את הוראות ההתקנה של **LWJGL** שמופיעות במדריך במודל.

כמו בתרגיל 4, בחבילה `il.ac.tau.cs.software1.inventory` (מעטה ואילך בהתייחסות לחבילות, נשמיט את הקידומת) מופיעים שני מנשקים: `Inventory`, `Collectible`. כאשר `Inventory` מייצג מנשק עבור אוסף של עצמים מטיפוס שממש את המנשק `Collectible`.

אנו נוסף שני מימושים נוספים ל `Inventory`.

**שימו לב:** בשונה מתרגיל 4, הגדרנו את המנשק `Inventory` לרשת מ `Iterable<Collectible>`. זה אומר שמחלקה שתממש את `Inventory` תצטרך לממש גם את המתודה `iterator`.

**`public interface Inventory extends Iterable<Collectible>`**

1. הקוד כפי שהוא, אינו מתקמפל משום שהמחלקה **`NaiveInventory`** שנמצאת תחת החבילה `inventory`, לא מכילה את המתודה `iterator`. הוסיפו מימוש (ריק כרגע) למתודה זו.
2. תחת החבילה `inventory`, צרו מחלקה חדשה בשם **`NaiveInventoryIterator`**, שתייצג את הטיפוס איטרטור עבור `NaiveInventory`. כלומר, איטרטור עבור הבונסים שנאספו ב `Inventory` עד עתה.
  - a. כאשר אתם מייצרים מחלקה זו, איזה מנשק עליה לממש?
  - b. אובייקט מטיפוס זה צריך לקבל גישה לאובייקט מטיפוס `NaiveInventory`. אפשר להעביר את האובייקט מטיפוס `NaiveInventory` כקלט לבנאי.
  - c. חישובו אילו שדות נוספים צריך האיטרטור להחזיק.
  - d. ממשו את המתודות `next`, `hasNext`.
3. עדכנו את מימוש המתודה `iterator` במחלקה **`NaiveInventory`** כך שתחזיר אובייקט חדש מטיפוס **`NaiveInventoryIterator`**.
4. תחת החבילה `inventory`, צרו מחלקה חדשה בשם **`MapInventory`** שתממש את המנשק `Inventory`. זה יהיה מימוש אחר ל `Inventory`, שבו נשמור את הבונסים שנאספים ב `HashMap` לפי `Price` של כל `Collectible` שאספנו. כלומר ה `HashMap` יכיל מפתחות מטיפוס `Float` וערכים מטיפוס `List<Collectible>`, וכשאר נעשה `collect` לבונס נשמור אותו ברשימה המתאימה לפי ה `Price` שלו. השתמשו במתודה `getPrice()` של `Collectible`.  
היזכרו בתרגיל 4. יהיה עליכם לממש את המתודות `getCurrentCount`, `getTotalWeight`, `collect`. וכן את המתודה `iterator`, כרגע עם מימוש ריק.

**כדי להבין כיצד לעבוד עם `HashMap` היעזרו בשקפים 11-26 במצגת הלמידה העצמית - מבני נתונים גנריים.**

5. תחת החבילה inventory, צרו מחלקה חדשה בשם `MapInventoryIterator`, שתייצג את הטיפוס איטרטור עבור `MapInventory`. כלומר, איטרטור עבור הבונוסים שנאספו ב `Inventory` כזה עד עתה.

- a. כאשר אתם מייצרים מחלקה זו, איזה מנשק עליה לממש?
- b. אובייקט מטיפוס זה צריך לקבל גישה ל `Map` של האובייקט מטיפוס `MapInventory`. אפשר להעביר את האובייקט מטיפוס `Map` כקלט לבנאי.
- c. חישבו אילו שדות נוספים צריך האיטרטור להחזיק.
- d. ממשו את המתודות `next`, `hasNext`.
- e. יהיה עליכם לעבור על תוכן של `map` היעזרו בשקפים 22-25 במצגת הלמידה העצמית – מבני נתונים גנריים.

f. יהיה עליכם לעבור על תוכן של רשי מות. זיכרו כי רשימה היא `iterable` וניתן לקבל איטרטור לאיברי רשימה. היעזרו בשקפים 16-17 במצגת הלמידה העצמית - מבני נתונים גנריים.

6. עדכנו את מימוש המתודה `iterator` במחלקה `MapInventory` כך שתחזיר אובייקט חדש מטיפוס `MapInventoryIterator`.

7. הריצו את הטסטר בקובץ `HW6_tester.java` שנמצא תחת החבילה `pacman` ובדקו שהפלט תואם את מה שציפיתם לראות.

8. עדכנו את שורה 7 בקובץ `InventoryComponent.java` בחבילה `inventory` להיות השורה הבא:

```
private Inventory inv = new MapInventory();
```

9. הריצו את הטסטר בקובץ `HW6_tester.java` ובדקו שהפלט תואם את מה שציפיתם לראות.

## חלק ב' (20 נק')

בחלק זה נתרגל עבודה עם אוספים (Collections). הקוד ימומש בחבילה `il.ac.tau.cs.sw1.ex6.collections` במחלקה `CollectionsExercise` שם כבר יש לכם את חתימות הפונקציות הרלוונטיות.

### סעיף 1 (10 נק')

ממשו את השירות `processDicts`. השירות מקבל רשימה של Map-ים שבהם המפתח הוא `Character` והערך הוא `Set<String>`.

בכל מילון המפתחות הם תווים, וכל תו ממופה ל`set` של מחרוזות. נאמר שתו `c` ממופה למחרוזת `s` במילון מסויים אם הזוג `c:'c'` מופיע במילון וגם `s` היא איבר ב `c_set`.

השירות יחזיר Map שיענה על הדרישות הבאות:

1. המפתחות בפלט יהיו איחוד מפתחות המילונים ברשימת הקלט.
2. הערך עבור מפתח `c` כלשהו יהיה המספר המקסימלי `k` של מילונים בהם `c` מופה למחרוזת מסויימת `s`.

לדוגמה, עבור הרשימה `mapsList` שבה שלושה מילונים.

```
mapsList = [ { 'a': {"tt"}, 'c': {"gg"} } , { 'a': {"kk"}, "mm" }, { 'b': {"hh"} } , { 'a': {"kk"}, 'b': {"yy"} } ]
```

הפונקציה תחזיר את המילון { 'a': 2, 'b': 1, 'c': 1 }. שלושת המילונים ב `mapsList` מכילים שלושה מפתחות שונים ('a', 'b', 'c'). עבור 'a' – המחרוזת "kk" מופיעה בקבוצה הממופה ל 'a' בשני מילונים שונים. 'b' אמנם מופיע בשני מילונים שונים, אך המחרוזות המופיעות בקבוצות הממופות ל 'b' הן שונות בין שני המילונים.

**היעזרו בשקפים 11-26 במצגת הלמידה העצמית - מבני נתונים גנריים.**

### סעיף 2 (10 נק')

ממשו את הפונקציה `weirdSorting` אשר מקבלת רשימה (List) של אובייקטים מטיפוס `Integer` ומחזירה רשימה חדשה שהיא הרשימה המקורית, ממוינת לפי `cos` שארית החלוקה במספר ראשוני נתון `p` (שנתון כקלט לפונקציה). כלומר, מספר `a` גדול ממספר `b` אם ורק אם `cos` שארית החלוקה של `a` ב-`p` גדול מ `cos` שארית החלוקה של `b` ב `p`. אין לשנות את המערך המקורי.

בסעיף זה ניתן להשתמש ב `Collections.sort` בלבד. **היעזרו בשקפים 26 ואילך במצגת הלמידה העצמית - מבני נתונים גנריים.**

**בהצלחה!**