

תרגיל 3

תאריך פרסום: 10.05.23

תאריך הגשה: 24.05.23

דגשים:

- קראו בקפידה את הנחיות ההגשה. אל תגישו קבצים או קוד מיותר ווודאו שאתם עומדים בכל ההנחיות לפני ההגשה.
- **אין לשנות חתימה של מתודות או קוד קיים בשום צורה** למעט ההגדרות של המחלקות עצמן. למשל להוסיף ממשקים נחוצים או לשנות את הגדרת הגנריות.
- מצורף לכם קובץ בדיקה שמאפשר לכם לבדוק את הקוד שלכם, אולם הוא לא מהווה את הבדיקה הכוללת שעליה ניתן לכם הציון.
- יש להקפיד לרשום הערות לאורך הקוד.
- יש להימנע משכפול קוד.
- שאלות לגבי העבודה יש לשאול בפורום המיועד.
- במקרה שבו הקוד עובר את מצב הריצה בהצלחה מלאה, אולם לא מתקפל או שמצב הבדיקה קורס, או שאתם חושבים שיש טעות במערכת הבדיקה, יש ליצור קשר עם ליאור במייל **בצירוף תיאור הבעיה והקבצים להגשה בלבד ארוזים ב-zip**.

הערות לביצוע העבודה:

- ניתן להשתמש רק בחומר שנלמד במסגרת ההרצאות.
- ניתן להשתמש ב-break ו-continue בלולאות.
- ניתן להעתיק אובייקטים באופן שטחי, אלא אם נאמר אחרת.
- ניתן להניח שהפרמטרים לפונקציות תקינים (כולל לא null) ושהקבצים תקינים, אלא אם נאמר אחרת.
- **אין להשתמש במבני הנתונים הבuiltin של Java.**
- יש מספר import מותרים, כל השאר יובילו לפסילת העבודה:
 - java.util.Iterator
 - java.util.NoSuchElementException
 - java.util.Scanner
 - java.io.*

חלק 1 – מבני נתונים ו-Serializable

בחלק זה יהיה עליכם לממש שני מבני נתונים, ArrayList ו-Set. על שני המבנים להיות גנריים, אולם הם דורשים אובייקטים שהם Serializable.

על המבנים עצמם לממש את הממשקים Serializable ו-Iterable (כאשר אם קוראים ל-next אבל hasNext()==false, נזרקת NoSuchElementException).

Null הוא איבר שיכול להופיע בסוף המערכים המתקבלים אולם זהו פרמטר לצורך החזקת מקום, ואף אחד מהמבנים הנ"ל לא "באמת" מכיל נ"ל. למשל Set אמור להחזיר שהוא לא מכיל null, גם אם המערך שמתחתיו מכיל כאלה לצורכי מקום.

ArrayList

עליכם לממש את המחלקה ArrayList כך שתהווה מערך שיכול לגדול.

אם צריך להגדיל את המערך, עליכם להכפיל את גודלו ב-2 ואין להקטין חזרה את גודלו. עליכם להשתמש במחלקה SimpleArray שניתנה לכם, קראו את הקוד שלה כדי להבין אילו מתודות ניתנו לכם.

עליכם לממש את המתודות הבאות (וכן את הממשקים הנ"ל):

חתימה	הסבר
public ArrayList()	על הבנאי לייצר מערך ריק בגודל 1.
public ArrayList(T[] arr)	בנאי שמקבל מערך, ניתן להעתיק באופן שטחי. ידוע שבמערך קודם מופיעים איברים רגילים, ורק לאחר מכן null. על גודל המבנה להיות בגודל המערך שניתן, ואורך המערך לפחות 1 (כולל nulls).
public T get(int index)	מחזירה את האיבר במיקום ה-index. אם אין אינדקס כזה, עליה לזרוק IndexOutOfBoundsException.
public int size()	מחזירה את כמות האיברים שיש במבנה.
void add(T element)	מוסיפה איבר למערך.
public void remove(T element)	מסירה איבר מהמערך. אם האיבר לא קיים, צריכה לזרוק NoSuchElementException.
public int index(T element)	מחזירה את המיקום הראשון של האיבר במערך. אם הוא לא קיים, מחזירה -1.
public int count(T element)	מחזירה את כמות הפעמים שהאיבר מופיע במערך.

Set

Set היא קבוצה בה כל איבר יכול להופיע לכל היותר פעם אחת וסדר האיברים לא חשוב. מומלץ להשתמש ב-ArrayList בתור מבנה הנתונים מאחורי הסט.

ניתן לקרוא עוד על סטים כאן: [https://en.wikipedia.org/wiki/Set_\(mathematics\)](https://en.wikipedia.org/wiki/Set_(mathematics))
עליכם לממש את המתודות הבאות (וכן את הממשקים הנ"ל):

חתימה	הסבר
public Set()	בנאי שמייצר סט ריק.
public Set(Iterable<? extends T> collection)	בנאי שמקבל iterable ומוסיף את כל איבריו לסט.
public Set(T[] arr)	בנאי שמקבל מערך ומוסיף את כל איבריו לסט. ידוע שבמערך קודם מופיעים איברים רגילים, ורק לאחר מכן null. אורך המערך לפחות 1 (כולל nulls).
public void add(T element)	מתודה שמוסיפה איבר לסט.
public boolean contains(T element)	מחזירה האם איבר נמצא בסט.
public int size()	מחזירה את כמות האיברים בסט.
public void remove(T element)	מסירה איבר מהסט אם הוא נמצא.
public Set<T> union(Set<T> other)	מייצרת סט חדש שהוא האיחוד של הסט הנוכחי עם סט נוסף.
public Set<T> intersection(Set<T> other)	מייצרת סט חדש שהוא החיתוך של הסט הנוכחי עם סט נוסף.
public Set<T> difference(Set<T> other)	מייצרת סט חדש שהוא החיסור של הסט הנוכחי בסט הנוסף.
public boolean equals(MySet<T> other)	בודקת האם הסטים שווים אחד לשני (מכילים את אותם איברים בדיוק).

חלק 2 – קריאה וכתיבה

כעת עליכם לכתוב את המחלקה PersonRegistry שמהווה מרשם של אוכלוסין (מבנה נתונים שמכיל אובייקטים מסוג Person (מחלקה שניתנת לכם, נא לקרוא אותה)) בו סדר האנשים חשוב.

למחלקה שדה boolean corrupted שנועד לדווח על מקרים של שגיאות, בחלק מהמתודות יהיה עליכם להפכוהו ל-true כתוצאה מ-exceptions. שימו לב שמחלקה זו צריכה לטפל השגיאות שעלולות לעלות ולא לזרוקן הלאה.

הערה: הפעולה get לא נבדקת באופן מפורש, אולם היא משמשת לבדיקות רבות אחרות. על כן, אם אתם רואים שגיאות לא ברורות (וככל הנראה גם רבות) על אף שהכול ממומש, כדאי לנסות להסתכל האם get עובדת כראוי.

חתימה	הסבר
public PersonRegistry(String filePath)	על המתודה לקרוא מקובץ אוסף של אנשים. אם הקובץ נגמר ב-".bin", הקובץ הוא קובץ שמכיל ArrayList של Person שמור דרך Serializable. אחרת, הקובץ הוא קובץ csv שמכיל איש בכל שורה באופן הבא: שם, גיל, מספר מזל. אחרי הפסיקים אין רווח. שימו לב להבדלים בין ירידת שורה של Windows לבין זו של Linux. אם הקריאה לא הצליחה מכל סיבה, על corrupted להידלק.
public PersonRegistry()	בנאי שמייצר מרשם אנשים ריק.
public PersonRegistry(Iterable<? extends Person> persons)	בנאי שמקבל Iterable ומוסיף את כל האנשים למרשם.
public PersonRegistry(Person[] persons)	בנאי שמקבל מערך ומוסיף את כל האנשים למרשם.
public void add(Person p)	מוסיפה אדם למרשם.
public Person get(int index)	מחזירה את האיבר במיקום ה-index. אם אין אינדקס כזה, עליה לזרוק IndexOutOfBoundsException.
public boolean writeCSV(String filePath)	רושמת את האנשים במרשם אל קובץ CSV בפורמט שמתואר בבנאי. מחזירה true אם המבנה נרשם בהצלחה, אחרת false.
public int maxAge()	מחזירה את הגיל הגדול ביותר מבין האנשים במרשם.

public double meanLuckyNumber()	מחזירה ממוצע של מספרי המזל של כל האנשים.
public MyArrayList<String> uniqueNames()	מחזירה ArrayList של כל השמות הייחודיים המופיעים במרשם.
public int count(Person person)	סופרת את כמות הפעמים שהאדם מופיע במרשם.