

תרגול #7 – רשימת ערים

בתרגול הזה נעסוק בסריקה ומיון של עצמים באוסף.

האוסף שנקים מכיל ערים גדולות בעולם.

בדף הקורס תמצאו קובץ בשם cities.txt המכיל רשימה של 80 הערים הגדולות בעולם.

לכל עיר נתון שמה, המדינה בה היא נמצאת, מספר התושבים בה (ב-2018) ושטחה (בק"מ רבוע). לדוגמה, להלן מספר שורות מהקובץ:

Kolkata	India	17560000	1351
Manila	Philippines	23088000	1873
Lagos	Nigeria	15279000	1965
Rio de Janeiro	Brazil	12272000	1912
Tianjin	China	10800000	2813
Kinshasa	Congo	13528000	473

נגדיר שתי מחלקות:

1. City – מחלקה המתארת עיר.

2. Cities – קבוצה של ערים.

המחלקה City כוללת את המידע שיש בקובץ הקלט לגבי העיר:

א. שם העיר

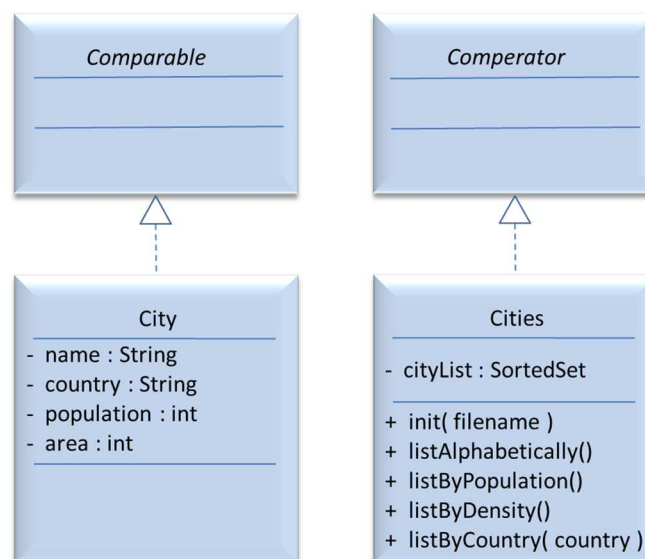
ב. המדינה בה העיר נמצאת

ג. מספר התושבים בעיר

ד. שטחה של העיר

המחלקה Cities כוללת רק קבוצה ממויינת של ערים – מופעים של המחלקה City.

ה-UML של מחלקות אלה מופיע בהמשך. כרגיל, אפשר שיהיו במחלקות אלה מתודות שאינן מצויינות ב-UML, אבל להסיק את קיומן מההקשר, כמו בנאים, getters, ומימושים של מתודות אבסטרקטיות מורשות.



מטלות:

1. ממשו את המחלקות לפי ה-UML. הותירו את המתודות ששמן מתחיל ב-list ללא תוכן לבינתיים.

2. ממשו את המתודה `init()` שמקבלת שם של קובץ קלט, קוראת אותו וטוענת את רשימת הערים בתוכן.
קובץ הקלט מתואר לעיל: בין השדות מפריד תו TAB (המוכר ב-Java כ-'`\t`') ובסוף השורה יש LF (המוכר ב-Java כ-'`\n`'). כדי לקרוא את הקובץ לתוך התכנית, אפשר להשתמש ב-`Scanner` כאשר מגדירים את שני התווים האלה כ-"תווים מפרידים" (`delimiters`) באמצעות קריאה למתודה `useDelimiters()` של `Scanner`.
שימו לב ששני שדות הם מחרוזות ושניים הם מספרים שלמים: אפשר להשתמש במתודות `nextInt()` ו-`hasNextInt()` של `Scanner` על מנת לקרוא את שני השדות האחרונים.

3. ממשו את המתודה `listAlphabetically()`, שאמורה להדפיס את הערים לפי הסדר האלפבתי של שמותיהן. זהו הסדר הטבעי של עצמים מסוג `City` – ודאו שכך הדבר באמצעות הגדרה מתאימה של המחלקה `City`.

ההדפסה צריכה להיות בפורמט הבא:

[שורת רווח]

City list

[שורת רווח]

Ahmedabad(India) population: 7410000 area: 359

[נוכח הלאה]

4. ממשו את המתודה `listByCountry()`, שמקבלת שם של מדינה כפרמטר ואמורה להדפיס רק את הערים המצויות במדינה זו. אפשר לסרוק את רשימת הערים ולהדפיס את אלה ששם המדינה שלהן מתאים לשם הנתון.
ההדפסה צריכה להיות בפורמט הבא:

[שורת רווח]

Cities in China:

[שורת רווח]

Beijing(China) population: 19433000 area: 4172

[נוכח הלאה]

5. ממשו את המתודה `listByPopulation()`, האמורה להדפיס את פרטי הערים כולן, כאשר הן ממויינות לפי מספר התושבים שבהן בסדר יורד, מהמאוכלסת ביותר (טוקיו) לדלילה ביותר (פוקואוקה). כיון שזה אינו הסדר הטבעי של הערים, יתכן שתמצאו לנכון להגדיר `Comparator` לצורך המיון.
ההדפסה צריכה להיות בפורמט הבא:

[שורת רווח]

City list by population

[שורת רווח]

Tokyo(Japan) population: 37977000 area: 8230

[וכך הלאה]

6. ממשו את המתודה `listByDensity()`, המדפיסה את הערים ופרטיהן לפי רמת הצפיפות שלהן: מספר התושבים מחולק בשטח. לצורך זה אפשר שתזדקקו ל-`Comparator` נוסף, שאותו יהיה עליכם, אולי, לממש במחלקה נפרדת מאלו שכבר הוגדרו.
ההדפסה צריכה להיות בפורמט הבא:

[שורת רווח]

City list by density

[שורת רווח]

Atlanta(United States) population: 5361000 area: 7400 density: 724

[וכך הלאה]

7. כתבו מחלקה ראשית: `Lab07Test` הבודקת את תקינות המחלקות והשיטות שכתבתם.

בהצלחה!