תרגיל 4

תכנות מונחה עצמים

# שאלה 3 (package cities)

בשאלה זו נשתמש באוספים של ג'אווה כדי לשמור מידע על מדינות ועל הערים בהן, וכדי לאפשר גישה נוחה למידע זה.

נכתוב שלוש מחלקות: City, Country, World, ושימוש בהן יראה בסוף כך:

World w = new World();

w.addCountry("Spain");

w.addCity("Granada", "Spain", 233764);

w.addCountry("Brazil");

w.addCity("Salvador", "Brazil", 2677000);

w.addCity("Barcelona", "Spain", 1615000);

w.addCity("Rio de Janeiro", "Brazil", 6320000);

System.out.println(w.report());

int bound = 2000000;

System.out.println("Cities with population under " + bound + ":");

System.out.println(w.smallCities(bound));

וזה ידפיס (ושימו לב, שהמדינות מסודרות בסדר מילוני, וגם הערים בתוך כל מדינה):

Brazil(8997000) : Rio de Janeiro(6320000), Salvador(2677000)

Spain(1848764) : Barcelona(1615000), Granada(233764)

Total population is 10845764

Cities with population under 2000000:

[Barcelona (of Spain), Granada (of Spain)]

## City

יש לה השיטות הבאות:

|  |  |
| --- | --- |
| public City(String name, Country country, int population) | בנאי המקבל את שם העיר, את המדינה בה היא נמצאת, ואת גודל האוכלוסיה. |
| public String getName() | מחזירה את שם העיר. |
| public Country getCountry() | מחזירה את המדינה. |
| public int getPopulation() | מחזירה את גודל האוכלוסיה. |
| public String toString() | מחזירה ייצוג כמחרוזת. לדוגמא:  “Paris (of France)” |

בנוסף, כדי שתוכלו למיין בקלות (בעזרת TreeSet של ג'אווה), תצטרכו לממש גם את equals וגם את compareTo (בפרט, מחלקה זו תצטרך לממש את הממשק <Comparable<City). ההשוואה לעיר אחרת תהיה קודם לפי שם המדינה, ואז אם המדינות שוות אז לפי שם העיר.

## Country

כל מדינה שומרת את קבוצת הערים שנמצאות בה:

private Set<City> cities;

|  |  |
| --- | --- |
| public Country(String name) | מאתחל מדינה עם השם הנתון. כרגע ללא ערים כלל. |
| public void addCity(City city) | מוסיף את העיר לרשימת הערים של המדינה. אם המדינה המופיעה בתוך אובייקט city היא לא המדינה הזאת, תזרק חריגהIllegalArgumentException. |
| public int population() | מחזירה את סכום כל האוכלוסיות בכל הערים במדינה. |
| public String toString() | מחזירה את שם המדינה. |
| public List<City> smallCities(int under) | מחזירה את רשימת כל הערים במדינה שיש בהן פחות מ-under תושבים, ממוינת לפי שם העיר. |

בנוסף יש לה שיטה:

public String report()

המחזירה מחרוזת מהצורה הבאה:

“Spain(1848764) : Barcelona(1615000), Granada(233764)”

כלומר את שם המדינה, ומספר התושבים בסוגריים, ואחריו רשימת הערים במדינה ממוינות לפי שם העיר, שלכל אחת רשומים מספר התושבים בסוגריים.

גם פה, בשביל המחלקה הבאה, תצטרכו לממש את equals ואת compareTo (ואת הממשק <Comparable<Country). כאן ההשוואה תהיה לפי שם המדינה.

## World

מחזיקה קבוצת מדינות, אבל כדי שנוכל למצוא את האובייקט של כל מדינה, מידע זה מוחזק ב-Map שלכל שם של מדינה, מחזיק את אובייקט המדינה המתאים לה.

|  |  |
| --- | --- |
| private Map<String, Country> countries | קבוצת המדינות, שמורות כך שניתן להגיע אליהן בעזרת השם שלהן. |
| public void addCountry(String name) | יוצרת מדינה חדשה לפי השם הנתון, ומוסיפה אותה ל-countries. |
| public void addCity(String name, String countryName, int population) | יוצרת עיר חדשה לפי הנתונים ומוסיפה אותה לאובייקט המדינה המתאים - אותו היא מוצאת בעזרת countries. אם אין מדינה בשם הזה, אז נזרק פה IllegalArgumentException. |
| public int population() | מחזירה את סך כל האוכלוסייה בכל המדינות הרשומות. |
| public List<City> smallCities(int under) | מחזירה אוסף של כל הערים שהוספנו שהאוכלוסיה שלהן קטנה מ-under. הרשימה ממוינת לפי שם המדינה, ובתוך אותה מדינה, לפי שם העיר. |
| public String report() | מחזירה מחרוזת המייצגת את כל הנתונים. ראו דוגמת הרצת בתחילת התרגיל. |