תרגיל 4

תכנות מונחה עצמים

# שאלה 2 (package equiv)

בשאלה זו נכתוב מחלקה גנרית התומכת בשמירת יחס שקילות. יש לכם החופש המלא להשתמש באיזה אלגוריתם שאתם רוצים, ואין צורך שהוא יהיה יעיל.

המחלקה שתכתבו נקראת <Equiv<E, ויש לה שתי שיטות:

|  |  |
| --- | --- |
| public void add(E e1, E e2) | שיטה לקבלת המידע ש-e1 ו-e2 הם שקולים. |
| public boolean are(E e1, E e2) | שיטה המחזירה true אם e1 ו-e2 באותה מחלקת שקילות. |

למשל:

Equiv<String> equiv = new Equiv<>();

equiv.add("ball", "balloon");

equiv.add("child", "person");

equiv.add("girl", "child");

equiv.add("ball", "sphere");

equiv.add("sphere", "circle");

equiv.add("dog", "cat");

System.out.println(equiv.are("balloon", "circle"));

System.out.println(equiv.are("child", "girl"));

System.out.println(equiv.are("sun", "sun"));

System.out.println(equiv.are("dog", "ball"));

System.out.println(equiv.are("table", "dog"));

ידפיס שלושה true ואז שני false.

אל תיבהלו מהניסוח המתמטי! אפשר לחשוב על הבעיה בצורה הבאה:

1. בתחילה, כל האוביקטים מסוג E (קיימים או לא קיימים) נמצאים כל אחד במחלקת שקילות משל עצמו.
2. ברגע שמישהו קורא ל-add, הוא בעצם מודיע לנו ששני אלמנטים הם שקולים, ולכן אפשר לאחד את מחלקות השקילות שלהם - ולקבל מחלקה אחת גדולה יותר. זה כי בעצם כל מי שהיה במחלקה של e1 שקול לכל מי שבמחלקה של e2.
3. כששואלים אותנו are, רק צריך לבדוק אם שני האוביקטים נמצאים באותה מחלקת שקילות.

ניתן להניח של-E יש hashCode ו-equals מוצלחים (למשל ל-String יש, ולכן הוא עובד בדוגמא).

### למי שרוצה הנחייה:

אפשר לשמור מערך (או עדיף, איזשהו List) של כל מחלקות השקילות (קבוצות של אברים - Set) שיש לנו כרגע. כשמישהו קורא ל-add:

1. עוברים על כל הקבוצות עד שמוצאים את זו שיש בה את e1 ואת זו שיש בה את e2.
2. אם הן אותה קבוצה, אז לא צריך לעשות כלום.
3. אם הם בקבוצות שונות, אז מוסיפים את אחת הקבוצות לשנייה ומוחקים את הקבוצה השנייה מרשימת הקבוצות.

המימוש של are אמור להיות ברור בשלב זה.