

## Chapter 19 - Generics

### תרגיל 18:

כתבו מחלקה גנרית המשמשת להגדרת קבוצות (set) של איברים מטיפוס כלשהו. ממשו את הקבוצה באמצעות מבנה נתונים מסוג List. הגדירו במחלקה את הפעולות הבאות:

- בנאי חסר פרמטרים המאתחל את הקבוצה כקבוצה ריקה.
  - בנאי המקבל מערך של איברים מהטיפוס ויוצר קבוצה המכילה איברים אלה.
  - union - פעולת איחוד המקבלת קבוצה ומאחדת אותה עם הקבוצה. כלומר הקבוצה עליה הופעלה הפעולה תייצג את האיחוד.
  - intersect - פעולת חיתוך המקבלת קבוצה ומבצעת חיתוך קבוצות. הקבוצה עליה הופעלה הפעולה תייצג את החיתוך.
  - subset – פעולה המקבלת קבוצה ובודקת אם היא מייצגת תת-קבוצה של הקבוצה.
  - isMember – פעולה המקבלת איבר ובודקת אם הוא שייך לקבוצה.
  - insert – פעולה המקבלת איבר ומוסיפה אותו לקבוצה.
  - delete – פעולת המקבלת איבר ומוציאה אותו מהקבוצה.
  - הגדירו מחדש (override) את המתודות toString ו-equals הנורשות מ-Object באופן הבא:
  - toString – תחזיר ייצוג מחרוזתי של הקבוצה.
  - equals – תבדוק אם הקבוצה שווה לקבוצה אחרת, כלומר אם היא מכילה את אותם האיברים.
- כתבו תכנית המשתמשת במחלקת הקבוצות באופן הבא:
- צרו 3 קבוצות מסוג Integer, מלאו כל קבוצה ב- 10 ערכים רנדומלים בתחום 0 .. 100 והציגו את תוכן הקבוצות.
  - בצעו איחוד של הקבוצה הראשונה עם הקבוצה השנייה והציגו את הקבוצה המתקבלת.
  - בצעו חיתוך של הקבוצה הראשונה (קבוצת האיחוד שהתקבלה מהפעולה הקודמת) עם הקבוצה השלישית והציגו את הקבוצה המתקבלת.
  - קלטו מהמשתמש 2 מספרים וצרו קבוצה רביעית. בדקו אם קבוצה זו היא תת-קבוצה של אחת הקבוצות והציגו את התוצאה.
  - קלטו מהמשתמש מספר ובדקו אם הוא שייך לקבוצה הראשונה והציגו את התוצאה.
  - קלטו מהמשתמש מספר נוסף, הוסיפו אותו לקבוצה הראשונה והציגו את התוצאה לאחר ההוספה.
  - קלטו מהמשתמש מספר נוסף, הוציאו אותו מהקבוצה הראשונה והציגו את התוצאה לאחר ההוצאה.
  - בדקו באמצעות equals אם הקבוצה הראשונה שווה לשנייה והציגו את התוצאה.
  - צרו קבוצות של מחרוזות והדגו את הפעולות union, intersect, subset.
  - צרו קבוצה של סטודנטים והדגו את הפעולות isMember, insert ו-delete. עליכם להגדיר מחלקת סטודנט המכילה פרטים כגון: שם, ת.ז, וציון. הגדירו במחלקת הסטודנט את המתודה equals כך שתבדוק שוויון בין שני סטודנטים על פי מספר ת.ז.

## תרגיל 19:

- א. כתבו מחלקה גנרית בשם `SortedPair` המשמשת ליצירת זוגות ממוינים של איברים (שני איברי הזוג הם מאותו הטיפוס). החלק הגנרי מגדיר את טיפוס האיברים כטיפוס הממש את הממשק `Comparable<T>`. כלומר כל שידוע על האיברים הוא שהם ניתנים להשוואה באמצעות המתודה `compareTo`. המחלקה תכלול בנאי המאפשר ליצור אובייקטים המייצגים זוג איברים מהטיפוס. הבנאי מקבל את האיברים בסדר כלשהו ויוצר מהם זוג ממוין. במקרה שהאיברים שווים יש לעורר מצב חריג בשם `IllegalPair` (עליכם להגדיר מחלקה זו כתת-מחלקה של `Exception`). הוסיפו מתודות `getFirst` ו-`getSecond` להחזרת האיבר הקטן והגדול של הזוג ומתודה `toString` להחזרת מחרוזת המייצגת את הזוג הממוין (קודם הקטן ולאחריו הגדול).
- ב. בדקו את המחלקה הגנרית `SortedPair` עבור זוגות של מחרוזות. כתבו תכנית ראשית הקולטת זוגות של מחרוזות, יוצרת זוגות ממוינים ושומרת אותם ב-`ArrayList`. לבסוף, יש לבצע איטרציה על ה-`ArrayList` ולהציג את הזוגות הממוינים.
- ג. בדקו את המחלקה הגנרית `SortedPair` עבור זוגות של נקודות במישור (כל נקודה מורכבת מ-x ו-y). בהינתן שתי נקודות: נקודה תחשב כקטנה מנקודה אחרת אם רכיב ה-x שלה קטן מרכיב ה-x של הנקודה האחרת, ורכיב ה-y שלה קטן מרכיב ה-y של הנקודה האחרת. הנקודה תחשב כגדולה מהנקודה האחרת אם רכיב ה-x שלה גדול מרכיב ה-x של הנקודה האחרת, ורכיב ה-y שלה גדול מרכיב ה-y של הנקודה האחרת. בכל מקרה אחר, הנקודות תחשבנה שוות. לדוגמה, הנקודה (5,7) גדולה מהנקודה (3,2) ואילו הנקודות (2,4) ו-(1,5) ייחשבו כשוות.
- כתבו תכנית ראשית המייצרת 10 זוגות של נקודות עם ערכי x ו-y רנדומלים בין 0.0 ל-100. עבור כל זוג, התכנית יוצרת זוג ממוין ומציגה אותו בפלט הסטנדרטי. במקרה של ניסיון ליצור זוג לא חוקי יש לתפוס את המצב החריג ולהציג הודעה מתאימה.