

# תוכנה 1 – סתיו תשע"ג

## תרגיל מספר 8

### הנחיות כלליות:

קראו בעיון את קובץ נהלי הגשת התרגילים אשר נמצא באתר הקורס.

- הגשת התרגיל תעשה במערכת ה moodle בלבד (<http://moodle.tau.ac.il/>).
- יש להגיש קובץ zip יחיד הנושא את שם המשתמש ומספר התרגיל (לדוגמא, עבור המשתמש zvainer יקרא הקובץ zvainer\_hw8.zip). קובץ ה-zip יכיל:
  - א. קובץ פרטים אישיים בשם details.txt המכיל את שמכם ומספר ת.ז.
  - ב. קבצי ה-java של התוכניות אותם התבקשתם לממש.
  - ג. קובץ טקסט אחד עם העתק של כל קבצי ה-java.
  - ד. קובץ answers עם תשובות לשאלות

### חלק א' (30%): שיפור סגנון קידוד (Java Coding Style)

בשאלה זו נתון לכם קובץ קוד בשם *TextAnalyzer.java* המממש מחלקה לניתוח טקסט. עליכם לשכתב את המחלקה תוך תיקון סגנון הקידוד כך שיעמוד במוסכמות הסגנון, ואף יהיה קריא, מובן, ויעיל ככל האפשר.

המחלקה הנתונה מקבלת שם קובץ כארגומנט בשורה הפקודה, ומבקשת מהמשתמש 2 שמות קבצים שישמשו לפלט. לקובץ הראשון תדפיס את כל המילים שהופיעו בקובץ הקלט ואת מספר המופעים שלהן. לקובץ השני תדפיס התוכנית את כל המילים שהופיעו בקובץ הקלט, כשהן ממוינות בסדר לקסיקוגרפי.

ניתן להניח שהמחלקה פועלת באופן תקין, למרות סגנון הקידוד הלקוי.

עליכם לשפץ את הקוד בהסתמך על המוסכמות לקידוד ג'אווה כפי שהן מופיעות בקישור הבא:  
<http://www.oracle.com/technetwork/java/codeconv-138413.html>

שימו לב במיוחד לנקודות הבאות (שעלו מניתוח מדגמי של התרגילים הקודמים שהגשתם):

- שמות משמעותיים למחלקות, משתנים ומתודות
  - שם המשתנה אמור לציין את תפקידו ואת התוכן שלו. לא ראשי תיבות, לא מילים בעברית, לא אותיות בודדות (אלא אם כן זה משתנה אינדקס למשל בלולאת For ואז מקובל להשתמש ב-i,j,k)
  - שמות של משתנים בולאניים יהיו מהצורה isVisible , isValid
  - שמות מחלקות יתחילו באות גדולה, שמות משתנים ומתודות באות קטנה, ולאחר מכן תופיע אות גדולה בתחילת כל מילה (למשל studentAverage).
  - שמות קבועים יהיו מהצורה COURSE\_ID.
  - שם המתודה יהיה לרוב פועל, ויצוין את הפונקציה אותה היא מבצעת.
- יש להשתמש בקבועים לאחסון קבועים או ערכים בהם יש שימוש חוזר.
- מניעת שכפול קוד – ארגון הקוד במתודות כאשר כל מתודה מבצעת פונקציה מוגדרת. אין לכתוב את כל הקוד במתודה ה-main.

- טווח ההכרה של המשתנים אמור להיות מינימלי – משתנים אמורים להיות כמה שיותר מקומיים.
- אינדנטציה (הזחה)
- תיעוד – יש לכתוב מעל כל מתודה, משתנה, בלוק קוד מה הם עושים, פירוט לפי רמת המורכבות, עם זאת אין לציין את המובן מאליו.
- הרשאות נראות (private, public) של שדות צריכות להיות מינימליות, וגישה פומבית לשדות (במידה והיא נדרשת) פרטיים צריכה להיעשות ע"י Getters ו-Setters.
- בכל שורה תופיע פקודה אחת. בכל שורה יופיע רק סוגר מסולסל יחיד.
- עדיף להפריד חישובים ולוגיקה מפעולות קלט\פלט.

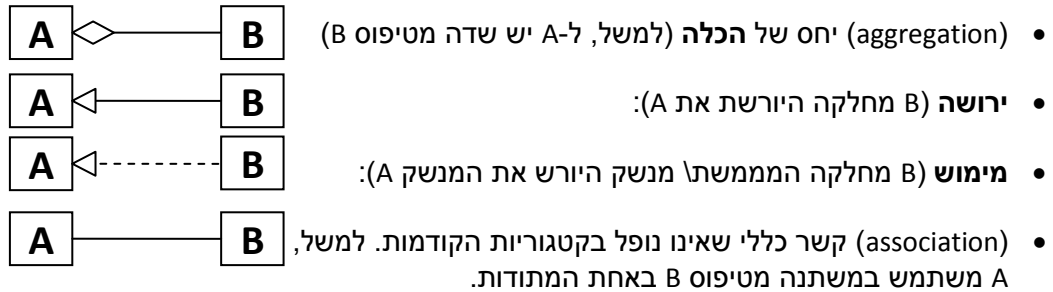
### בשאלה זו עליכם להגיש:

1. קובץ קוד מתוקן עבור המחלקה הכתוב על פי הכללים המפורטים בקישור הנ"ל. על הקובץ להיות בעל סגנון קידוד מושלם.
2. הסבירו בקצרה מהם 5 התיקונים השונים המשמעותיים ביותר שביצעתם ומדוע הם חשובים (להגיש בקובץ התשובות).

❖ שימו לב: מתרגיל זה והלאה, סגנון קידוד ראוי יהיה חלק מדרישות התרגילים ועל כן עליכם להקפיד על כתיבת קוד על פי כללי הסגנון.

## חלק ב' (70%): התעשיין העשיר (הורשה)

**הערה כללית:** בתרגיל זה אתם מתבקשים, בין היתר, לשרטט דיאגרמות של מחלקות. השתמשו בסימונים הבאים בלבד:



**יש לציין:** בתוך כל מלבן <<interface>>, <<abstract>> או <<class>>, ואת שם המנשק או המחלקה.

**אין צורך לציין:** מספרים ושמות שדות על יחסי אגרגציה ואסוציאציה; שמות מתודות ושדות בתוך מלבני המחלקות; יחסים "עקיפים" בין מחלקות (כלומר, אם C יורש מ-B שירש מ-A, אין צורך לציין קשר בין C ל-A אלא אם יש ביניהם קשר ישיר בנוסף, למשל של הכלה)

**יצירת הדיאגרמות:** ניתן לעשות זאת דרך Word, PowerPoint, תוכנת הציור המועדפת עליכם או לסרוק שרטוט (בכתב ברור!). כדי למנוע בעיות formatting שונות פורמט המועדף עבור answers.pdf הוא .pdf.

בחלק זה נתרגל הורשה באמצעות מחלקות לחישוב ירושה של תעשיינים עשירים. ברשות כל תעשיין עשיר סכום כסף מסוים, והוא מחליט כיצד לחלק את הכסף בין יורשיו הפוטנציאליים במותו.

עליכם להגדיר כל אחד משבעת הטיפוסים הבאים כמנשק, מחלקה מופשטת (אבסטרקטית) או מחלקה קונקרטית. בנוסף הציגו תרשים מחלקות המתאר את היחס בין הטיפוסים שהגדרתם (מכונה גם עץ הורשה או היררכית מחלקות) בקובץ answers, לפי המפתח המתואר למעלה.

- **Mortal** - טיפוס זה מתאר את כל מי או מה שיכול למות באופן כללי.
- **RichIndustrialist** - טיפוס המתאר תעשיין עשיר. בפרט, לתעשיין כזה יש שם וסכום כסף (holdings, אחזקות) במיליוני דולרים, ולמרבית הצער הוא יכול למות (רק פעם אחת).
- **PredecessorRichIndustrialist** - תעשיין עשיר אשר יש לו רשימת יורשים פוטנציאליים (successors), כולם תעשיינים עשירים כשלעצמם. יורש שמת יימחק מן הרשימה.
- **FairRichIndustrialist** - תעשיין עשיר הוגן אשר מוריש סכום שווה לכל אחד מיורשיו הפוטנציאליים.
- **UnfairRichIndustrialist** - תעשיין עשיר בלתי-הוגן, אשר מעדיף את היורש הפוטנציאלי הראשון (זה שהתווסף הראשון לרשימה מבין כל שאר היורשים), ומוריש לו את כל הכסף.
- **CrazyRichIndustrialist** - תעשיין משוגע אשר מוריש לכל אחד מיורשיו הפוטנציאליים אחוזים רנדומאליים מכספו (כשסכום האחוזים מסתכם למאה). לדוגמא, הוא יכול להחליט לתת X אחוזים ליורש הראשון, Y אחוזים לשני, ולשלישי והאחרון X-Y-100 אחוזים.
- **LonerRichIndustrialist** - תעשיין בודד שאין לו יורשים פוטנציאליים, והוא מוריש במותו את כל כספו לצדקה. נייצג את הארגון אליו תורם התעשיין הבודד ע"י מחרוזת (שם הארגון).

**הערות**

- כל אחד משבעת הטיפוסים הנ"ל יגדיר את המתודה `public void die()`
- כל תעשיין עשיר מקבל, בעת יצירתו (בבנאי), שם וסכום כסף התחלתי. התעשיין הבודד מקבל גם את שם ארגון הצדקה אליו יתרום במותו.
- במותו, כל תעשיין מחלק את כל כספו, ובפרט לא נותר לו כסף.
- במותו, כל תעשיין מדפיס לפלט הסטנדרטי הודעה כמתואר למטה. התעשיין הבודד מדפיס גם כמה תרם לצדקה.
- היעזרו במחלקה `java.util.Random` ובפרט במתודה `nextDouble()` כדי לבחור אחוזים רנדומליים עבור תעשיינים משוגעים.
- אתם רשאים להגדיר מחלקות עזר לפי הצורך (הקפידו להוסיף אותן לתרשים)
- תשובתכם צריכה לבטא **שימוש חוזר** ברכיבי תוכנה **ולמזער את שכפול הקוד**. הדבר יעשה, בין השאר, ע"י בחירה נכונה של מחלקות קונקרטיות, מחלקות מופשטות ומנשקים. קוד עובד הוא תנאי הכרחי אך לא מספיק במקרה זה.

אופן מימוש המחלקות המדויק נתון לבחירתכם, אך עליכם לממש את המתודות והבנאים המופיעים בקוד הלקוח הבא, באופן שיתקבל פלט בדומה לפלט הנתון.

```
public static void main(String[] args) {
    // 1st generation
    CrazyRichIndustrialist uma = new CrazyRichIndustrialist("Uma", 30.0);
    FairRichIndustrialist bill = new FairRichIndustrialist("Bill", 20.0);

    // 2nd generation
    UnfairRichIndustrialist sean = new UnfairRichIndustrialist("Sean", 1.0);
    uma.addSuccessor(sean);
    bill.addSuccessor(sean);

    CrazyRichIndustrialist lucy = new CrazyRichIndustrialist("Lucy", 1.0);
    uma.addSuccessor(lucy);
    bill.addSuccessor(lucy);

    // 3rd generation
    LonerRichIndustrialist max = new LonerRichIndustrialist("Max", 1.0,
        "SOS animals");
    sean.addSuccessor(max);

    FairRichIndustrialist rita = new FairRichIndustrialist("Rita", 1.0);
    sean.addSuccessor(rita);

    uma.die(); // Sean and Lucy get random percentages
    sean.die(); // Max gets everything and Rita gets nothing
    bill.die(); // Lucy gets everything since Sean died
    max.die(); // SOS animals get everything

    System.out.println("####");
    System.out.println(uma.getName() + " has " + uma.getHoldings()
        + " millions");
    System.out.println(bill.getName() + " has " + bill.getHoldings()
        + " millions");
    System.out.println(sean.getName() + " has " + sean.getHoldings()
        + " millions");
    System.out.println(lucy.getName() + " has " + lucy.getHoldings()
        + " millions");
    System.out.println(max.getName() + " has " + max.getHoldings()
        + " millions");
}
```

```

System.out.println(rita.getName() + " has " + rita.getHoldings()
                    + " millions");
}

```

להלן פלט אפשרי. שימו לב שבגלל האלמנט הרנדומאלי בירושה של תעשיין משוגע, הסכומים המודגשים בצהוב עשויים להיות שונים, אך סכומם צריך להיות 53 (סכום הכספים של אומה, ביל, שון, לואי ומקס. ייתכן אי דיוק קטן בגלל השימוש ב-double).

```

Uma has passed away.
Sean has passed away.
Bill has passed away.
Max has passed away.
27.485007755241206 millions donated to SOS animals.
####
Uma has 0.0 millions
Bill has 0.0 millions
Sean has 0.0 millions
Lucy has 25.514992244758794 millions
Max has 0.0 millions
Rita has 1.0 millions

```

**בהצלחה !**