

תוכנה 1 - תרגיל מספר 5

בשתי השאלות הראשונות נדון במחלקות שמייצגות מספרים רציונליים, כגון $1/3$ או $99/100$. תזכורת מתמטית קצרה: כדי לצמצם שברים צריך לחלק את המונה והמכנה בגורם המשותף הגדול ביותר (GCD באנגלית). למשל, הגורם המשותף הגדול ביותר של 42 ו-56 הוא 14, ולכן:

$$\frac{42}{56} = \frac{3 \cdot 14}{4 \cdot 14} = \frac{3}{4}.$$

כדי לחבר או לחסר שברים, צריך למצוא את המכנה המשותף (LCM באנגלית). למשל, המכנה המשותף של 21 ו-6 הוא 42, ולכן:

$$\frac{2}{21} + \frac{1}{6} = \frac{4}{42} + \frac{7}{42} = \frac{11}{42},$$

שאלה 1

א. בין המתכנתים ביל ואומה התפתח ויכוח האם **טרם** לכתובת המחלקה המייצגת מספרים רציונליים יש לכתוב **מנשק** המתאר את הפונקציונליות של המחלקה. ביל טען כי המחלקה פשוטה דיה וכי אין הצדקה לתאר את אותו הדבר גם ע"י מנשק וגם ע"י מחלקה. ציינו 3 טיעונים **נגדיים** המצדדים בשימוש במנשק. נמקו או הדגימו בקצרה את טיעונכם.

ב. לאחר שביל השתכנע בנחיצות המנשק, החל דיון בינו ובין אומה על נחיצות השרותים הבאים. עבור כל אחד מהם ציינו האם יש לו מקום במנשק, במחלקה, בשניהם או באף אחד מהם. נמקו בקצרה:

1. add - לחיבור שני מספרים רציונליים
2. subtract - לחיסור שני מספרים רציונליים
3. multiply - להכפלת שני מספרים רציונליים
4. divide - לחלוקת שני מספרים רציונליים
5. equals - להשוואת שני מספרים רציונליים
6. gcd - לחישוב הגורם המשותף הגדול ביותר של שני שלמים
7. lcm - לחישוב מכנה משותף של שני שלמים
8. normalize - להבאת שבר פשוט למצב מצומצם
9. toString - לייצוג המספר כמחרוזת של שבר פשוט (למשל לצורכי הדפסה)
10. toDecimalString - לייצוג המספר כמחרוזת של שבר עשרוני (למשל לצורכי הדפסה)

ג. מה לגבי השרותים: `getNumerator` ו-`getDenominator` להחזרת המונה והמכנה של השבר בהתאמה? ביל ואומה הסכימו כי הדבר תלוי בהקשרי השימוש של הטיפוס החדש. ציינו באילו הקשרים יש הצדקה להכללת השרותים במנשק ובאילו אין. נמקו או הדגימו את תשובתכם.

שאלה 2

ביל ואומה מתלבטים לגבי הקבעון (mutability) של הטיפוס החדש. טיפוס מקובע (immutable) הוא טיפוס שלא ניתן לשנות את ערכי העצמים שלו לאחר שנוצרו. ראינו בתרגול את הטיפוסים String ו-StringBuffer המממשים גרסא מקובעת ושאינה מקובעת למחרוזות.

א. ציינו את היתרונות שיש לטיפוס רציונלי מקובע.

ב. ציינו את היתרונות שיש לטיפוס רציונלי שאינו מקובע.

ג. הגדירו את המנשק MutableRational לתאור טיפוס רציונלי שאינו מקובע (יש להכליל את הקובץ MutableRational.java בקובץ ה- zip). על המנשק להכיל את החתימות המלאות של כל השרותים שציינתם בשאלה 1 סעיף ב' שהם חלק מהמנשק. יש לציין את **המצב המופשט** (abstract state) של הטיפוס במקום המיועד לכך מעל הגדרת המנשק. כמו כן, עבור **כל אחד מהשרותים** יש לציין את משמעותו בעזרת המצב המופשט (יצויין בהערה מעל חתימת השרות).

בתאור המצב המופשט ניתן להשתמש במבנים הבאים:

- this – המצב הקונקרטי של העצם לאחר ביצוע השרות
- AF(<exp>) - המצב המופשט של <exp>
- \$prev(<exp>) – מצב \ ערך <exp> לפני ביצוע השרות
- \$ret – הערך המוחזר של השרות

ד. הגדירו את המנשק ImmutableRational לתאור טיפוס רציונלי מקובע (יש להכליל את הקובץ ImmutableRational.java בקובץ ה- zip). על המנשק להכיל את החתימות המלאות של כל השרותים שציינתם בשאלה 1 סעיף ב' שהם חלק מהמנשק. יש לציין את **המצב המופשט** (abstract state) של הטיפוס במקום המיועד לכך מעל הגדרת המנשק. כמו כן, עבור **כל אחד מהשרותים** יש לציין את משמעותו בעזרת המצב המופשט (יצויין בהערה מעל חתימת השרות).

בתאור המצב המופשט ניתן להשתמש במבנים הבאים:

- this – המצב הקונקרטי של העצם לאחר ביצוע השרות
- AF(<exp>) - המצב המופשט של <exp>
- \$prev(<exp>) – מצב \ ערך <exp> לפני ביצוע השרות
- \$ret – הערך המוחזר של השרות

להלן דוגמת שימוש במנשקים :

```
Course c1 = new ... ; // a course of 3 units, and max number of students 40  
Course c2 = new ... ; // a course with 4 units, and max number of students 40
```

```
Student s1 = new ... ;  
Student s2 = new ... ;
```

```
System.out.println(s1.totalUnits());
```

```
s1.register(c1);  
s2.register(c1);  
s1.register(c2);
```

```
System.out.println(s1.totalUnits());  
s1.drop(c1);  
System.out.println(s1.totalUnits());
```

הפלט של סדרת הפעולות יהיה :

```
0  
7  
4
```

א. כתבו את החוזה של המנשק Course : לכל שרות כתבו תנאי קדם (precondition) ותנאי אחר (postcondition) באופן המקובל (ביטויים בוליאניים שיכולים להשתמש בשאילתות). במידת הצורך, הוסיפו במילים תנאים שלא ניתנים לביטוי בצורה הרגילה. יש להוסיף את הערות החוזה לקובץ העזר Course.java המצורף לתרגיל.

ב. כתבו את החוזה של המנשק Student : לכל שרות כתבו תנאי קדם (precondition) ותנאי אחר (postcondition) באופן המקובל (ביטויים בוליאניים שיכולים להשתמש בשאילתות). במידת הצורך, הוסיפו במילים תנאים שלא ניתנים לביטוי בצורה הרגילה. יש להוסיף את הערות החוזה לקובץ העזר Student.java המצורף לתרגיל.

ג. המחלקה SimpleCourse אמורה לממש את הממשק Course בצורה פשוטה, תוך שימוש במערך לייצוג הסטודנטים הרשומים לקורס. נתון חלק מהקוד – כל השדות, הבנאי, ומימוש של שרות אחד.

```
public class SimpleCourse implements Course {  
  
    private int maxNumStudents;  
    private int top;  
    private int units;  
    private int level;  
    private Student[] students;  
  
    public SimpleCourse(int maxNumStudents, int units, int level) {  
        this.maxNumStudents = maxNumStudents;  
        students = new Student [maxNumStudents];  
        this.units = units;  
        this.level = level;  
        top = -1;  
    }  
  
    public void register(Student s) {  
        students[++top] = s;  
    }  
  
    // more code omitted  
}
```

1. הגדירו את משתמר הייצוג (representation invariant) של המחלקה SimpleCourse

2. השלימו את הקוד של המחלקה SimpleCourse.

הוראות הגשה:

1. קראו בעיון את קובץ נוהלי הגשת התרגילים אשר נמצא באתר הקורס.
2. הגשת התרגיל תעשה ע"י המערכת VirtualTAU (<http://virtual2002.tau.ac.il/>).
3. הגשת התרגיל היא ע"י יצירת קובץ zip אשר יכיל את התשובות המילוליות, וקובצי העזר בצירוף החוזה והמצב המופשט.
 - א. יש לקרוא לקובץ ה-zip לפי שם המשתמש שלכם, למשל: zvainer.zip
 - ב. יש לצרף קובץ פרטים אישיים בשם details.txt שיכיל את שמכם ומספר תעודת הזהות שלכם
 - ג. נוסף על הקבצים הבודדים יש להגיש קובץ טקסטואלי נוסף (doc/txt/rtf) שיכיל את כל תשובותיכם

בהצלחה