

# פתרון הבחינה בתוכנה 1

סמסטר א', מועד א', תשס"ט  
15/02/2009

דן הלפרין, אוהד ברזילי, ליאור שפירא, מתי שמרת

הוראות (נא לקרוא!)

- משך הבחינה **שלוש שעות** - חלקו את זמנכם ביעילות.
- יש לענות על כל השאלות.
- בשאלות שבהן יש צורך לנמק, תשובה ללא נימוק לא תזכה באף נקודה.
- יש לענות על כל השאלות בגוף הבחינה במקום המיועד לכך. המקום המיועד מספיק לתשובות מלאות (ויותר). יש לצרף את טופס המבחן למחברת הבחינה. מחברת ללא טופס עזר תפסל. תשובות במחברת הבחינה לא תיבדקנה. במידת הצורך ניתן לכתוב בגב טופס הבחינה.
- יש למלא מספר סידורי (מס' מחברת) ומספר ת.ז על כל דף של טופס הבחינה.
- ניתן להניח לאורך השאלה שכל החבילות הדרושות יובאו, ואין צורך לכתוב שורות import.
- במקומות בהם תתבקשו לכתוב מתודה (שירות), ניתן לכתוב גם מתודות עזר.
- אסור השימוש בחומר עזר כלשהוא, כולל מחשבוני או כל מכשיר אחר פרט לעט. בסוף הבחינה צורף לנוחותכם נספח ובו תיעוד מחלקות שימושיות.

שאלה	א	ב	ג	ד	ה	ו	ז	ח	סה"כ
1	2	2	2	2	2	2	12	12	35
2	6	5	12	6	6				35
3	12								12
4	2	2	2	2					8
5	5								5
6	5								5
	ציון בחינה:								100

בהצלחה!

## שאלה 1 (35 נקודות)

נתונה סידרה של אברים, ללא חזרות, מטיפוס כלשהו. נגדיר את "סידרת העוצמה" (PowerList) להיות קבוצת כל תת הסדרות של הסידרה המקורית, כולל תת הסידרה הריקה. סדר הופעת תת-הסדרות בקבוצה אינו משנה. לעומת זאת, מעצם הגדרת תת-סידרה, סדר הופעת האיברים בה נקבע על פי הסידרה המקורית. כלומר, אם אבר מסוים הופיע אחרי אבר אחר בסידרה המקורית לא ייתכן כי בתת סידרה הוא יופיע לפניו. שימו לב כי סידרה אינה בהכרח ממוינת.

לדוגמא, בהינתן סידרת השלמים [2,9,5] קבוצת תת-הסדרות היא:  
 { [], [2], [9], [5], [2,9], [2,5], [9,5], [2,9,5] }

נסמן ב- T את טיפוס אברי הסידרה המקורית. מוטלת עלינו המשימה לממש את המחלקה PowerList המממשת את הממשק IList. הנתון.

```
public interface IList<T> extends Iterable< *1* > {
    public int bitFlag(List<T> subseq); // הסבר בהמשך
}
```

למחלקה שנממש יהיה שדה פרטי השומר את הסידרה המקורית וכן בנאי אחד שמקבל סידרה ושומר כבר בעת הבניה את רשימת כל תת-הסדרות.

נתון מימוש חסר של המחלקה PowerList:

```
public class PowerList < *2* > implements IList <T> {
    private List< *3* > originalList;
    private Set< *4* > powerList = new HashSet< *5* >();

    // The constructor.
    public PowerList(List<T> list) {
        ...
    }

    // The iterator.
    @Override
    public Iterator< *1* > iterator() {
        ...
    }

    // Transform a subsequence into bit flag representation
    @Override
    public int bitFlags(List<T> subseq) {
        ...
    }
}
```

וכן נתונה דוגמה להפעלת הבנאי כאשר T הוא Integer :

```
List<Integer> list = new ArrayList<Integer>();
// populate list ...

IPowerList<Integer> powerList = new PowerList<Integer>(list);
```

ענו על כל שמונת הסעיפים הבאים המתייחסים למנשק ולמחלקה כפי שהוגדרו לעיל:

א. (2 נקודות) האם ניתן להכריז על הבנאי במנשק? אם כן, כיצד? אם לא, מדוע לא?

**לא, לא ניתן להכריז על בנאי במנשק.  
בנאי קשור למימוש המחלקה ולא למנשק הציבורי שלה**

ב. (2 נקודות) מה צריך לכתוב במקום \*1\* ? List<T>

ג. (2 נקודות) מה צריך לכתוב במקום \*2\* ? T

ד. (2 נקודות) מה צריך לכתוב במקום \*3\* ? T

ה. (2 נקודות) מה צריך לכתוב במקום \*4\* ? List<T>

ו. (2 נקודות) מה צריך לכתוב במקום \*5\* ? List<T>

ז. (12 נקודות) ממשו את הבנאי של המחלקה.

אלגוריתם אפשרי ליצירת רשימת תת-הסדרות בהינתן סדרה S הוא:  
אתחל רשימה חדשה והכנס לתוכה את תת-הסדרה הריקה.  
לכל איבר s בסדרה S בצע הפעולות הבאות:

- העתק את רשימת תת-הסדרות שיצרת עד כה והוסף לכל תת-סדרה בעותק את s
- מזג את שתי הרשימות, המקורית והעותק ששינינו, לרשימה אחת

```
public PowerList(List<T> list) {
    originalList = new ArrayList<T>(list);

    powerList.add(new ArrayList<T>());

    Set<List<T>> helpSet = new HashSet<List<T>>();

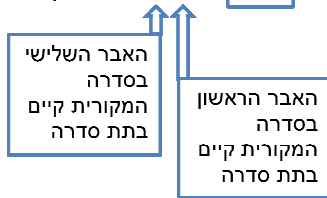
    for (T t : originalList) {
        for (List<T> l : powerList) {
            List<T> lCopy = new ArrayList<T>(l);
            lCopy.add(t);
            helpSet.add(lCopy);
        }
        powerList.addAll(helpSet);
        helpSet.clear();
    }
}
```

#### שגיאות נפוצות:

- "העתקה ע"י השמה"
- אי העתקה של הארגומנט list (אלא רק השמה של ההפניה)
- העתקה רדודה (shallow) במקום העתקה מלאה (deep)
- new List() (List הוא ממשק!)
- משתנה מקומי שהוגדר private (private מיועד לשדות בלבד!)

ח. (11 נקודות) כתבו את המתודה bitFlags המקבלת תת-סידרה ומחזירה מספר שלם מטיפוס int. מספר זה מייצג את תת-הסדרה כך שכל ביט בו מייצג האם אבר מסוים מהסדרה המקורית נמצא בתת הסדרה הנתונה. למשל:

סדרה מקורית  $[2,9,5]$  נפעיל את bitFlags([2,5]) ונקבל  $5 = \dots 000101$  (ייצוג בינארי)



סדרה מקורית  $[4,8,10,12]$  נפעיל את bitFlags([8,10,12]) ונקבל  $14 = 0001110$  (ייצוג בינארי)

תנאי הקדם של המתודה bitFlags דורש כי הסידרה המקורית אינה מכילה יותר משלושים ושניים אברים וכי תת הסידרה subseq המועברת למתודה היא אכן תת-סידרה של הסידרה המקורית.

```
public int bitFlag(List<T> subseq) {
    int result = 0;
    int bitmask;

    for (T t : subseq) {
        bitmask = 1 << originalList.indexOf(t);
        result |= bitmask;
    }

    return result;
}
```

שגיאות נפוצות:

- Math.pow מחזירה double ולכן יש להמיר הערך שהיא מחזירה ל - int
- הדלקת הביט ה - 32 ע"י Math.pow(2,32)
- שגיאה לוגית: לקדם אינדקס i ב - 1 כל פעם כשמתקדמים ב - subseq ולפי i זה להדליק ביטים בדגל (במקום ש - i יספור את המקום על פני ה - originalList)

## שאלה 2 (35 נקודות)

שאלה זו עוסקת בתכנון ובהגדרת טיפוס נתונים עבור פולינום (polynom – רב איבר) שמקדמיו שלמים. בשאלה נדון בטיפוס מקובע (immutable).

- פולינום הוא ביטוי מהצורה:  $c_0 + c_1x + c_2x^2 + c_3x^3 + \dots + c_nx^n$ .
- $c_0, c_1, \dots, c_n$  הם מספרים שלמים, והם נקראים **המקדמים** של הפולינום (coefficients).
- המעריך הגדול ביותר בפולינום (של איבר שהמקדם שלו אינו אפס) נקרא **הדרגה** (degree) של הפולינום.

לדוגמא: הביטוי  $2x^3 - 5x^2 + 13$  הוא פולינום שדרגתו 3 ומקדמיו הם: 2, -5, 0, 13.

א. (6 נקודות) נתון המנשק IPolynomial ובו שרותים המתוארים בצורה מילולית. עבור כל שרות הוסיפו תיעוד של השרות במונחי **המצב המופשט** של הטיפוס כפי שמתואר בראש המנשק. את תשובותיכם מלאו במקום המיועד מעל לחתימת כל אחת מהשרותים:

```

/** פולינום מקובע (immutable) שמקדמיו שלמים
 * @abst:  $c_0 + c_1x + c_2x^2 + c_3x^3 + \dots + c_nx^n$ 
 */
public interface IPolynomial {

    /** השרות מחזיר את דרגת הפולינום
     *
     * @abst: $ret = n
     */
    public int degree ( );

    /** השרות מחזיר את המקדם של האיבר עם מעריך d
     *
     * @abst: $ret =  $c_d$ 
     */
    public int coeff (int d);

    /** השרות מחזיר את הסכום של הפולינום הנוכחי ושל הארגומנט q
     *
     * @abst: $ret = AF(this) + AF(q)
     */
    public IPolynomial add (IPolynomial q);

    /** השרות מחזיר את המכפלה של הפולינום הנוכחי ושל הארגומנט q
     *
     * @abst: $ret = AF(this) * AF(q)
     */
    public IPolynomial mul (IPolynomial q);

    /** מחזיר את תוצאת חיסור q מהפולינום הנוכחי
     *
     * @abst: $ret = AF(this) - AF(q)
     */
    public IPolynomial sub (IPolynomial q);

    /** מחזיר פולינום שמקדמיו נגדיים (מינוס) למקדמי הפולינום הנוכחי
     *
     * @abst: $ret = -AF(this)
     */
    public IPolynomial minus ( );
}

```

ב. (5 נקודות) אומה הסתומה עדיין לא יודעת איך לממש את המנשק הנתון, אולם ביל הדביל מציע לה למרות זאת לממש במחלקה מופשטת (abstract class) את אחד השרותים, בצורה שלא תהיה תלויה במימוש שיבחר לבסוף. ממשו את השרות הזה בקוד המחלקה המופשטת:

```
public abstract class AbstPolynom implements IPolynom {

    public IPolynom sub(IPolynom q) {
        return add(q.minus());
    }
}
```

טעויות נפוצות:

• נסיון להגדיר את minus בעזרת mul(-1). זו טעות מכיוון שהפונקציה mul מקבלת IPolynom ולא int

ג. (12 נקודות) בסעיף הבא אתם מתבקשים לממש את המחלקה Polynom.

להלן תזכורת מתמטית:

**סכום של שני פולינומים** הוא פולינום שהמקדמים שלו הינם סכומים של המקדמים המתאימים (לפי חזקות) של שני הפולינומים הנתונים. למשל, עבור  $a(x) = 6x^3 + 3x^2 + 8$  ו-  $b(x) = 2x^2 + 4x + 1$  נקבל:  $a(x) + b(x) = 6x^3 + 5x^2 + 4x + 9$ .

**מכפלה של שני פולינומים** הינה פולינום, אשר מתקבל כתוצאה של "פתיחת סוגריים" רגילה בביטוי:

$$a(x) \cdot b(x) = (a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0) \cdot (b_m x^m + b_{m-1} x^{m-1} + \dots + b_1 x + b_0)$$

לדוגמה, עבור  $a(x) = 3x^2 + x + 2$  ו-  $b(x) = x^3 + 4x$  נקבל:

$$a(x) \cdot b(x) = (3x^2 + x + 2) \cdot (x^3 + 4x) = 3x^{2+3} + 3 \cdot 4x^{2+1} + x^{1+3} + 4x^{1+1} + 2x^3 + 8x = 3x^5 + 12x^3 + x^4 + 4x^2 + 2x^3 + 8x = 3x^5 + x^4 + 14x^3 + 4x^2 + 8x.$$

ובמקרה הכללי:

$$\begin{aligned} p(x) &= p_0 + p_1 x + p_2 x^2 + \dots + p_m x^m \\ q(x) &= q_0 + q_1 x + q_2 x^2 + \dots + q_n x^n \\ p(x)q(x) &= p_0 q_0 + (p_0 q_1 + p_1 q_0)x + (p_0 q_2 + p_1 q_1 + p_2 q_0)x^2 \\ &\quad + (p_0 q_3 + p_1 q_2 + p_2 q_1 + p_3 q_0)x^3 \\ &\quad + (p_0 q_4 + p_1 q_3 + p_2 q_2 + p_3 q_1 + p_4 q_0)x^4 + \dots \\ &\quad + (p_0 q_{n+m} + p_1 q_{n+m-1} + p_2 q_{n+m-2} \dots \\ &\quad \quad + p_{n+m-1} q_1 + p_{n+m} q_0)x^{n+m}. \end{aligned}$$

ממשו את המחלקה Polynom (בעמוד הבא). אם לצורך הגדרת המחלקה יש צורך בשרותי עזר נוספים, שדות, בנאים או טיפוסים חדשים ממשו גם אותם.

```
public class Polynom extends AbstPolynom {

    private int[] trms;

    /**
     * @pre coeffs.length > 0
     */
    public Polynom(int... coeffs) {
        int len = 0;
        for (int i = 0; i < coeffs.length; i++) {
            if (coeffs[i] != 0)
                len = i;
        }

        trms = new int[len + 1];
        for (int i = 0; i < trms.length; i++)
            trms[i] = coeffs[i];
    }

    // Zero Polynom
    public Polynom() {
        this(0);
    }

    /**
     * Returns the degree of this, i.e., the largest exponent with a non-zero
     * coefficient. Returns 0 if this is the zero polynom.
     */
    public int degree() {
        return trms.length - 1;
    }

    /**
     * Returns the coefficient of the term of this whose exponent is d
     */
    public int coeff(int d) {
        if (d < 0 || d > degree())
            return 0;
        else
            return trms[d];
    }

    /**
     * If q is null throws NullPointerException;
     * else returns the polynom this + q.
     */
    public Polynom add(IPolynom q) {
        int[] result = new int[Math.max(degree(), q.degree()) + 1];
        for (int i = 0; i < result.length; i++) {
            result[i] = coeff(i) + q.coeff(i);
        }
        return new Polynom(result);
    }
}
```



```

/**
 * Returns the polynom -(this)
 */
public Polynom minus() {
    int[] result = new int[degree() + 1];
    for (int i = 0; i < result.length; i++) {
        result[i] = -trms[i];
    }
    return new Polynom(result);
}

/**
 * If q is null throws NullPointerException;
 * else returns the polynom this * q.
 */
public Polynom mul(IPolynom q) {
    int[] result = new int[degree() + q.degree() + 1];
    for (int i = 0; i <= degree(); i++)
        for (int j = 0; j <= q.degree(); j++)
            result[i + j] += coeff(i) * q.coeff(j);
    return new Polynom(result);
}
}

```

#### שגיאות נפוצות:

- לא לבדוק בבנאי כי המקדמים הגבוהים של הפולינום אינם אפס. אם לא בודקים זאת יש לבדוק זאת בפונקציה `degree`
- אם לא בודקים כי המקדמים הגבוהים של הפולינום אינם אפס לא בבנאי ולא בפונקציה `degree`, יש לבדוק זאת בסוף כל פעולת חיבור, חיסור וכפל
- שינוי הפולינום הנוכחי במהלך פעולות חיבור/חיסור/כפל
- אי הגדרת בנאי נוח (למשל בנאי שמקבל Map כארגומנט אינו נוח לשימוש)
- ניתן היה להגדיר תנאי קדם לבנאי, אולם היה צורך לבדוק אותם כאשר קוראים לבנאי ממקום אחר בקוד
- הגדרת פונקציית עזר ציבורית שמחזירה את שדה המקדמים (למשל המערך). הפונקציה חושפת את המימוש הפנימי!
- קבלת ארגומנט מערך בבנאי והשמה שלו לשדה (במקום העתקה שלו)
- אי בדיקת תנאי קצה בארגומנט של הפונקציה `coeff`
- התייחסות לארגומנט `q` כאילו שהוא מטיפוס `Polynom`

ד. (6 נקודות) מהי פונקצית ההפשטה של המימוש אותו בחרתם?

```
AF (this) = trms[0] + trms[1] · x + ... + trms[trms.length - 1] · xtrms.length-1
```

ה. (6 נקודות) מהו משתמר הייצוג של המימוש אותו בחרתם?

```
@representation_inv: trms != null &&  
    trms.length > 0 &&  
    trms.length == degree() + 1 &&  
    trms[i] == coeff(i) &&  
    trms[trms.length-1] == 0 $implies  
        this.equals(new Polynom() )
```

### שאלה 3 (12 נקודות)

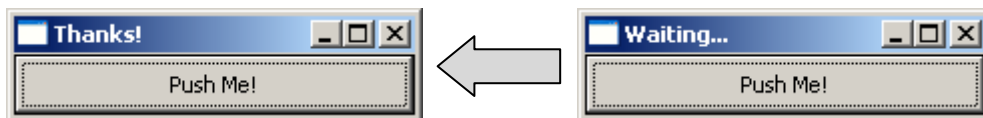
נתונה המחלקה הבאה המגדירה חלון ובו כפתור יחיד.

```

1. public class SimpleButton {
2.     public static void main(String[] args) {
3.         Display display = new Display ();
4.         Shell shell = new Shell (display);
5.         shell.setText("Waiting...");
6.
7.
8.         Button button = new Button(shell, SWT.PUSH);
9.         button.setText ("Push Me!");
10.        button.setLocation(0,0);
11.        button.setSize(200,30);
12.
13.        shell.pack ();
14.        shell.open ();
15.        while (!shell.isDisposed ()) {
16.            if (!display.readAndDispatch ())
17.                display.sleep ();
18.        }
19.        display.dispose ();
20.    }

```

עליכם לשנות את הקוד כך שלאחר הלחיצה הראשונה על הכפתור תוחלף כותרת החלון מ "Waiting..." ל- "Thanks!" , כפי שמופיע בדוגמה:



מותר לכם לשנות את הקוד בכל דרך הנראית לכם - הוספת מתודות / מחלקות, שינוי המתודה main, הוספת שדות, שימוש בירושה וכו' - ובלבד שתתקבל התוצאה המבוקשת. אם הינכם משנים / מוסיפים קוד במחלקה לעיל ציינו, בעזרת מספרי השורות, את מקום השינוי / הוספה.

**שגיאות נפוצות (פתרון מוצע בעמוד הבא):**

- הוספת מאזין ל- shell במקום לכפתור
- מימוש המנשק SelectionListener ללא מימוש כל המתודות הרלוונטיות
- אי הוספת המצייין final למשתנה shell
- שינוי כיתוב הכפתור במקום כיתוב מסגרת החלון

**השינויים הנדרשים מודגשים בצהוב:**

```
public class SimpleButton {  
  
    public static void main(String[] args) {  
  
        Display display = new Display ();  
  
        final Shell shell = new Shell (display);  
  
        shell.setText("Waiting...");  
  
  
        Button button = new Button(shell, SWT.PUSH);  
  
        button.setText ("Push Me!");  
  
        button.setLocation(0,0);  
  
        button.setSize(200,30);  
  
        button.addSelectionListener(new SelectionAdapter() {  
  
            public void widgetSelected(SelectionEvent arg0) {  
  
                shell.setText("Thanks!");  
  
            }  
  
        });  
  
        shell.pack ();  
  
        shell.open ();  
  
        while (!shell.isDisposed ()) {  
  
            if (!display.readAndDispatch ())  
  
                display.sleep ();  
  
        }  
  
        display.dispose ();  
  
    }  
  
}
```

**שאלה 4 (8 נקודות)**

במחלקה Base מוגדרת המתודה foo שכותרתה היא:

```
void foo() throws IllegalArgumentException
```

במחלקה Derived דורסים את המתודה foo

```
void foo() =MISSING_CODE=
```

**עבור כל אחת מהאפשרויות הבאות ענו האם יכולה להחליף את =MISSING\_CODE=.**  
נמקו את תשובותיכם:

א. throws NumberFormatException

חוקי. מותר לזרוק חריג שירש מהחריג שניזרק במתודה הנדרסת במחלקת הבסיס,

ב. throws IllegalArgumentException

חוקי. טריויאלי (לא שינינו דבר)

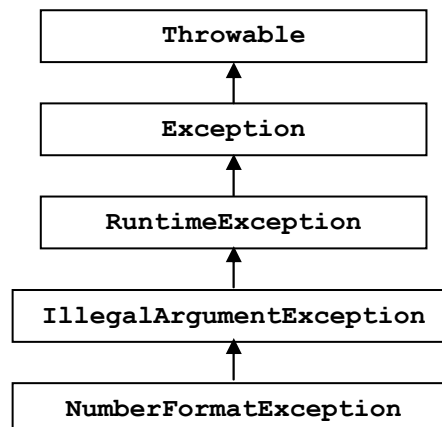
ג. throws RuntimeException

חוקי. RuntimeException אינו חריג ניבדק ולכן מותר לזרוק אותו בכל מיקרה.

ד. כלום. דהיינו לאחר ההחלפה כותרת הפונקציה תהיה void foo()

חוקי. מותר למתודה דורסת לא לזרוק חריג (לא נפתיע הלקוח עם חריג לא צפוי)

היררכיית ההורשה שבה מופיע החריג NumberFormatException היא:



שאלה 5 (5 נקודות)

נתונה התכנית הבאה:

```

01 class Rectangle {}
02
03 class ColoredRectangle extends Rectangle {}
04
05 class Base {
06     void foo(Rectangle x) {
07         System.out.println("Base: foo(Rectangle)");
08     }
09 }
10
11 class Sub extends Base {
12     @Override
13     void foo(ColoredRectangle x) {
14         System.out.println("Sub: foo(ColoredRectangle)");
15     }
16 }
17
18 public class WhatHappensHere {
19
20     public static void main(String[] args) {
21         Sub o = new Sub();
22         Rectangle rc = new ColoredRectangle();
23
24         ((Base) o).foo(rc);
25     }
26 }
    
```

בחרו את המשפט הנכון מבין האפשרויות הבאות:

א. בהרצת התוכנית WhatHappensHere תודפס למסך השורה:  
 Base: foo(Rectangle)  
 והתוכנית תסתיים בהצלחה.

ב. בהרצת התוכנית WhatHappensHere תודפס למסך השורה:  
 Sub: foo(ColoredRectangle)  
 והתוכנית תסתיים בהצלחה.

ג. קוד התוכנית לא עובר קומפילציה בגלל שורה/ות מספר 12/13 הבעיה היא התגית @Override מחייבת שהפונקציה שמתחתיה תדרוס או תממש פונקציה שהגיעה בירושה (או מהממשק). הפונקציה foo אינה דורסת את הפונקציה foo ממחלקת הבסיס מכיוון שחתימתה שונה!

ד. הרצת התוכנית תגרור חריג בזמן ריצה מטיפוס: \_\_\_\_\_ בשורה מספר \_\_\_\_ . הבעיה היא:

---



---



---

## שאלה 6 (5 נקודות)

נתונה התכנית הבאה:

```
class A {
    private int x = 5;

    public A(int x) {
        this.x = x;
    }

    public A(int x, int y) {
        this(x + y);
    }

    public String toString() {
        return Integer.toString(x);
    }
}

public class B extends A {
    private int x = 10;

    public B (int x) {
        super(x, x);
        this.x = x;
    }

    public B(int x, int y) {
        this(x * y);
    }

    public String toString() {
        return Integer.toString(x);
    }

    public static void main(String[] args) {
        A a1 = new B(4, 2);
        A a2 = new A(4, 2);
        System.out.println(a1 + " " + a2);
    }
}
```

מה יהיה הפלט של התכנית. בחרו אחת מהתשובות הבאות:

- a) 10 5
- b) 8 6**
- c) 8 5
- d) 16 16

**ושוב, בהצלחה**