

## מבחן בתוכנה 1

### ליאור שפירא, אורנית דרור

מועד ב' סמסטר קיץ תשס"ו, 8 בדצמבר 2006

#### הוראות

משך הבחינה שלוש שעות.

יש לענות על כל השאלות (מספר הנקודות לא משקף בהכרח את הקושי או הזמן הדרוש).  
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. 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 (מספר מחברת) ומספר ת"ז על כל דף של טופס הבחינה.

אסור השימוש בחומר עזר כלשהו, כולל מחשבוניו או כל מכשיר אחר פרט לעט.

ז	ו	ה	ד	ג	ב	א		
		5	5	5	5	5	1	
			2.5	2.5	2.5	2.5	2	
					5	10	3	
					10	10	4	
5	5	5	5	5		5	5	
							100	סה"כ



## שאלה מספר 1 (25 נקודות)

נתונים ממשק ומחלקה:

```
public interface I {
    public I f(I o) throws ClassCastException;
}

public abstract class A implements I {
    public void g(I o) {...};
}
```

ברצוננו לכתוב מחלקה בשם B המרחיבה את A:

עבור כל אחד מהסעיפים הבאים ציינו ונמקו האם המחלקה המתקבלת תקינה. כל שינוי הוא נפרד (אלה לא שינויים מצטברים).

א.

```
public abstract class B extends A{}
```

**חוקי, B מופשטת ולכן לא חייבת לממש שום דבר**

ב.

```
public class B extends A {
    public I f(B in) throws Exception {
        throw new ClassCastException();
    }
}
```

**B איננה מממשת את הממשק I (חתימה שונה לשרות), לא תקין!**

ג.

```
public class B extends A {
    public void g(I o) {...}
}
```

**לא תקין, B מרחיבה את A ולכן חייבת לממש את I. אך אין מימוש של F**

. ד

```
public class B extends A implements I {
    public I f(I in) {
        return null;
    }
    public void g(B o) {...}
}
```

תקין. B מרחיבה את A (ומממשת בהכרח את I), יש מימוש של f, החוזה מחוזק ע"י הורדת החריג. השירות g הממומש כאן לא דורס את השירות של A (חתימה שונה)

. ה

```
public abstract class B extends A {
    static public A f(A in) throws Exception {
        return null;
    }
}
```

~~לא תקין. אסור להסתיר שירות מופע public עם שירות מחלקה, כמובן שאין מימוש חוקי של המנשק I.~~



## שאלה מספר 2 (10 נקודות)

א. בהינתן ההגדרות במסגרת, אילו מההגדרות הבאות דורשות `explicit cast` (סמנו את כל התשובות הנכונות)

```
byte myByte;
int myInt;
long myLong;
char myChar;
float myFloat;
double myDouble;
```

1. `myInt = myByte;`
2. `myInt = myLong;`
3. `myByte = 3;`
4. `myInt = myChar;`
5. `myFloat = myDouble;`
6. `myFloat = 3;`
7. `myDouble = 3.0;`

ב. אם המערך `arr` מכיל רק ערכים חיוביים, מה עושה הפונקציה הבאה (תשובה אחת):

```
public int guessWhat( int arr[] )
{
    int x= 0;
    for( int i = 0; i < arr.length; i++ )
        x = x < arr[i] ? arr[i] : x;
    return x;
}
```

1. מחזירה את האינדקס של הערך הגבוה ביותר במערך
2. מחזירה `true/false` אם יש ערכים חוזרים במערך
3. מחזירה כמה ערכים זוגיים במערך
4. מחזירה את הערך הגבוה במערך

ג. בהינתן הקוד במסגרת, אילו מהשורות מודפסות אם `fragile()` זורק חריג מסוג `IllegalArgumentException` (סמנו בין 0 ל-5 תשובות)

```
void myMethod()
{
    try
    {
        fragile();
    }
    catch( NullPointerException npex )
    {
        System.out.println(
            "NullPointerException thrown " );
    }
    catch( Exception ex )
    {
        System.out.println(
            "Exception thrown " );
    }
    finally
    {
        System.out.println(
            "Done with exceptions " );
    }
    System.out.println( "myMethod is done" );
}
```

1. "NullPointerException thrown"
2. "Exception thrown"
3. "Done with exceptions"
4. "myMethod is done"
5. כלום לא מודפס

ד. האם הקוד במסגרת תקין?

```
1: class Test
2: {
3:     void show()
4:     {
5:         System.out.println(
6:             "non-static method in Test");
7:     }
8: }
9: public class Q3 extends Test
10: {
11:     static void show()
12:     {
13:         System.out.println(
14:             "Overridden non-static method in Q3");
15:     }
16: }
17: public static void main(String[] args)
18: {
19:     Q3 a = new Q3();
20: }
```

1. שגיאת קומפילציה בשורה 3
2. שגיאת קומפילציה בשורה 10
3. שגיאה בזמן ריצה בשורה 3
4. שגיאה בזמן ריצה בשורה 10
5. אין שגיאה בקוד

### שאלה 3 (15 נקודות)

בהינתן הקוד הבא:

<pre>class A {     protected int x,y;      public A(int x, int y) {         this.x = x % 10;         this.y = (y/10) % 10;     }      public A(int x) {         this();         this.x = x % 10;     }      private A() {         x = 0;         y = 0;     }      public void f() {         x = x*x;     } }</pre>	<pre>public class B extends A{     int z;      public B(int x, int y) {         super(x,y);         z = (x / 100) % 10;     }      public B(int x) {         super(x,1);         z = this.x * this.y + x;     }      public void g() {         System.out.println(x + y + z);     } }</pre>
---	---

א. ציינו ונמקו אילו מהשינויים הבאים למחלקה הם תקינים

1. הוספת השירות הבא ל-B: `protected void f() { ... }`

לא תקין, אסור להסתיר את f

2. הוספת המלה השמורה `abstract` בהגדרת המחלקה A

תקין, A כעת מופשטת

3. הוספת השדה `private double x;` למחלקה B

תקין, x מסתיר משתנה באותו שם המוגדר ב-A.

ב. אילו פרמטרים צריך להעביר לבנאי של B כדי שקריאה לשירות g תדפיס 12? (תיתכן יותר מתשובה אחת אפשרית) נמקו!

```
B b = new B(606,0);
```

נקרא בנאי עם 2 פרמטרים, קורא לבנאי עם 2 פרמטרים A (עם 606 ו-0). X מקבל את ספרת האחדות, ז"א 6. y מקבל 0. חזרה לבנאי של B שם z מקבל את ספרת המאות מהפרמטר x, ז"א 6. קריאה ל-g() תחבר את שלושתם ותדפיס 12 (הערה: יש המון המון אפשרויות נוספות)

## שאלה מספר 4 (20 נקודות)

### חלק א'

נתון קטע הקוד הבא הבונה ממשק גראפי:

```
public static void createGUI() {

    Display display = new Display ();
    final Shell shell = new Shell (display);
    Label label = new Label (shell, SWT.NONE);
    label.setText ("Name:");
    final Text text = new Text (shell, SWT.BORDER);
    Button ok = new Button (shell, SWT.PUSH);
    ok.setText ("OK");
    ok.addListener(new SelectionAdapter() {
        public void widgetSelected(SelectionEvent e) {
            System.out.println("Your name is " + text.getText());
            shell.close();
        }
    });

    shell.setLayout (new RowLayout ());
    shell.pack ();
    shell.open ();
    while (!shell.isDisposed ()) {
        if (!display.readAndDispatch ()) display.sleep ();
    }
    display.dispose ();
}
```

1) איזה ממשק ממשק SelectionAdapter? מדוע אנו משתמשים בו ולא מממשים את הממשק ישירות בעצמנו?

ממשק את SelectionListener, מחלקה זו הינה מחלקת עזר. בממשק ישנם מספר שירותים באין ברצוננו לממש (חוץ מ-widgetSelected) ולכן Adapter חוסך לנו קוד

2) כיצד מתנהג הממשק הגראפי? בהנחה שאני רושם "moshe" בתוך text ולוחץ OK, מה מודפס?

מודפס "your name is moshe" והתכנית יוצאת

3) תנו דוגמה לעוד סוג של סידור פקדים אפשרי והסבירו כיצד היה נראה הממשק הגראפי החדש

כרגע הפקדים מסודרים בשורה, ניתן היה לסדר אותם ב-GridLayout, במקרה זה היינו מסדרים אותם בעמודות, אחת ליד השנייה (כמו טבלה)...

חלק ב'

(1) מה עושה השירות הבא?

```

public void bar(String in, String out) throws ... {
    BufferedReader reader = new BufferedReader(new FileReader(in));
    BufferedWriter writer = new BufferedWriter(new FileWriter(out));

    String s;
    List l = new LinkedList();
    while ((s = reader.readLine()) != null)
        l.add(s);
    Collections.sort(l);
    Collections.reverse(l);
    for (Object o : l)
        writer.write((String)o);
}

```

קורא קובץ טקסט שורה שורה לתוך רשימה מקושרת. ממיין את הרשימה בסדר לקסיקוגרפי יורד וכותב חזרה לקובץ חדש

(2) מבין החריגים הבאים, אילו עלולים להיזרק בקטע הקוד שבסעיף 1? לכל חריג ציין אם יש להכריז עליו ב-throws של השירות

java.lang.RuntimeException

תמיד עלול להיזרק, אין צורך להכריז עליו

java.lang.ClassCastException

מבצעים Casting של אובייקט למחרוזת, אז טכנית עלול להיזרק. אך אנו יודעים בוודאות ש-o הוא מחרוזת ולכן אין סכנה. בכל מקרה לא צריך להכריז עליו

java.lang.ClassNotFoundException

לא עלול לקרות, לא צריך להכריז

java.io.IOException

עלול לקרות (בפתיחת קובץ, קריאה וכתובה) ויש להכריז עליו

## שאלה 5 (30 נקודות)

התבקשתם לכתוב מערכת המנהלת ספר מתכונים. לכל מתכון מספר מרכיבים שיחד מרכיבים מנה. במערכת יש צורך להוסיף מתכונים חדשים, להוסיף רכיבים לכל מתכון, לחשב מה עלות הכנה של מתכון ועוד...

נתונים המנשקים הבאים:

```
package recipes;

public interface Recipe {

    // return name of recipe, for example "mama's chocolate chip cookies"
    public String name();

    // add the given amount of an ingredient to the recipe
    public void addIngredient(Ingredient ingredient, int amount);

    // calculate the cost of the recipe (sum of (ingredient's
    // costs * cost per unit))
    public double cost();

    // if an Ingredient with the same name as ingredient exists in
    // the recipe, return its amount in the recipe. Otherwise return 0.
    public int contains(Ingredient ingredient);
}

public interface Ingredient {
    public String name(); //name of ingredient

    public String color(); //main color

    public boolean isSweet(); //is it sweet

    public double pricePerUnit(); //price per unit
}
```

להלן דוגמת שימוש במנשקים:

```
Ingredient chicken = new ...
Ingredient lemon = new ...
Ingredient parsley = new ...

Recipe r = new ...
r.addIngredient(chicken, 1);
r.addIngredient(lemon, 3);
r.addIngredient(parsley, 3);

System.out.println("Cost of recipe " + r.name() + " is " + r.cost());
System.out.println(r.name() + " contains how much chicken? " +
    r.contains(chicken));
```

במקרה הספציפי הזה יודפס:

Cost of recipe Roast Chicken is 32.5  
 Roast Chicken contains how much chicken? 1



א. כיתבו את החוזה של המנשק Recipe: לכל שרות כיתבו תנאי קדם (precondition) ותנאי אחר (postcondition) באופן המקובל (ביטויים בוליאניים שמשמשים בשאילתות). במידת הצורך, הוסיפו במילים תנאים שלא ניתנים לביטוי בצורה הרגילה.

```
Post:Return name of recipe
```

```
public String name();
```

```
pre:ingredient not null, amount > 0
```

```
post: contains(ingredient) == amount
```

```
post: prev@cost() + ingredient.pricePerUnit()*amount = cost()
```

```
public void addIngredient( Ingredient ingredient, int amount);
```

```
post: cost() == sum of contains(ingredient) for each ingredient in the recipe
```

```
public double cost();
```

```
pre: ingredient != null
```

```
post: 0 if ingredient in recipe, otherwise amount given in addIngredient
```

```
public int contains(Ingredient ingredient);
```

ב. המחלקה RecipeArray אמורה לממש את הממשק Recipe בצורה פשוטה באמצעות מערך (של רכיבים ושל כמויות). למען הפשטות נניח כי אין יותר מ-10 רכיבים במתכון.

```
public class RecipeArray implements Recipe {
    public RecipeArray(String name) {
        this.name = name;

        ingredients = new Ingredient[10];
        amounts = new int[10];
        count = 0;
    }

    public String name() {
        return name;
    }

    public void addIngredient(Ingredient ingredient, int amount) {
        ...//omitted
    }

    public double cost() {
        ...//omitted
    }

    public int contains(Ingredient ingredient) {
        ... //omitted
    }

    private String name;
    private Ingredient[] ingredients;
    private int[] amounts;
    private int count;
}
```

ג. הגדירו את משתמר המחלקה של RecipeArray:

(בקצרה ולא פורמלית)

**name, ingredients, amounts, count not null**  
**count equal to number of ingredients**  
**amounts[i] >= 0 for each i**

ד. השלימו את הקוד של שני השירותים addIngredient ו-contains:

```
public void addIngredient(Ingredient ingredient, int amount) {
    ingredients[count] = ingredient;
    amounts[count] = amount;
    count++;
}
public int contains(Ingredient ingredient) {
    for (int i=0;i<count;i++) {
        if (!ingredients[i].name().compareTo(ingredient.name()))
            return amounts[i];
    }
    return 0;
}
```

ה. נתגלתה בעיה במערכת. כאשר מחפשים מתכונים עם רכיב מסוים (צנון למשל), לוקח הרבה זמן לחפש בכל מתכון האם הרכיב קיים. הציעו (אל תממשו, רק תפרטו את הרעיון, כולל מחלקות שבהם תשתמשו) מימוש אחר של Recipe בו נוכל למצוא אם רכיב מסוים קיים במתכון במהירות

ניתן לשמור את הרכיבים Map במקום במערך, כאשר המפתח הינו שם הרכיב. נוכל לחפש את שם הרכיב ובמהירות לדעת אם הוא קיים במתכון או לא (יותר מהר מלעבור על כל המערך ולבדוק)

ו. ישנה דרישה לעשות מימוש חדש של מתכון – AllergicRecipe, מימוש זה יקבל בבנאי רכיב אליו המשתמש אלרגי. נמקו האם השינויים הבאים בחוזה חוקיים:

<p>נימוק: לא חוקי, החוזה דורש שאם נוסף רכיב הוא יהיה חלק מהמתכון</p>	<p>כאשר יקראו ל-addIngredient על המתכון עם הרכיב הנ"ל, לא באמת נוסף אותו למתכון.</p>
<p>נימוק: מפר את חוזה המנשק ולכן לא חוקי</p>	<p>נשנה את addIngredient של המחלקה כך שיחזיר false אם מנסים להוסיף את הרכיב האלרגני, ו-true לכל רכיב אחר</p>

ז. מעוניינים לכתוב שני מימושים של המגשק Ingredient, אחד עבור רכיבים בשריים ואחד עבור רכיבים צמחיים.

```
public class Vegetable implements Ingredient {
    ...

    private boolean isGoodForYou;
    private String name;
    private String color;
    private boolean isSweet;
    private double pricePerUnit;
}

public class Meat implements Ingredient {
    ...

    private String animal;
    private boolean isKosher;
    private String name;
    private String color;
    private boolean isSweet;
    private double pricePerUnit;
}
```

נראה כי רוב השירותים והשדות יהיו משותפים למחלקות אלו. כיצד ניתן למנוע שכפול קוד? מה הייתם ממליצים לעשות?

ניתן לכתוב מחלקה מופשטת משותפת אשר גם Vegetable וגם Meat יירשו ממנה, מחלקה זו תכיל את כל השדות המשותפים ותגדיר כמה שיותר שירותים משותפים