

**מבחן בתוכנה 1 - פתרונות**

**ליאור שפירא, אורנית דרור**

**מועד א' סמסטר קיץ תשס"ו, 20 בספטמבר 2006**

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

**שאלה מספר 1 (25 נקודות) (5 נקודות לסעיף) (פחות 2 נק' בלי נימוק)**

נתון המנשק הבא:

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

ברצוננו לכתוב מחלקה בשם A הממשת את המנשק I:

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

א.

```
public abstract class A implements I{}
```

המחלקה המתקבלת הינה תקינה, A מחלקה מופשטת ולכן לא חייבת לממש את המנשק

ב.

```
public class A implements I {
    public A f(I o) throws Exception {
        throw new Exception();
    }
}
```

המחלקה המתקבלת תקינה, השירות f ב-A מממש את שירות f של המנשק. הערך החוזר הינו A, מימוש של I ולכן תקין

ג.

```
public class A implements I {
    public I f(A a) throws Exception {
        throw new Exception();
    }
}
```

לא תקין. השירות f המוגדר ב-A אינו מממש את השירות של המנשק (חתימה שונה)

. ד

```
public class A implements I {
    public I f(I o) {
        return null;
    }
}
```

המימוש תקין. השירות f ב-A מממש את השירות של הממשק I , אומנם מימוש זה לא זורק חריג, אך הדבר רק מחזק את החוזה ולכן תקין.

. ה

```
public class B {
    public B(int i) {}
    public I f(I o) throws Exception{
        throw new Exception();
    }
}
public class A extends B implements I {}
```

המימוש אינו תקין, A אינה מגדירה בנאי ולא קוראת מפורשות לבנאי של B ולכן ביצירת מופע של A ייקרא בנאי ברירת המחדל של B. אין כזה ולכן לא תקין.

**שאלה מספר 2 (10 נקודות) (2 נק' לשאלה, 1/2 נק' לסעיף בד')**

א. סמן את המשפט הנכון

1. מחלקה (class) היא מופע (instance) של עצם (object)
2. למחלקה (class) יש מצב רגעי, המוגדר ע"י השדות שלה (המשתנים השייכים לה)

3. **עצמים (Objects) מכילים שירותים (מתודות) הפועלים על השדות של העצם. ערך השדות מייצג את המצב הרגעי של העצם (state)**
4. עצמים (Objects), מחלקות (Classes) ומנשקים (Interfaces) הינם שמות נרדפים לאותו דבר

- ב. אם אנו רוצים להגדיר שירות (מתודה) במחלקה, כך שלא ניתן יהיה לדרוס (override) את השירות במחלקות-בנות, ושניתן יהיה להפעיל את השירות מתוך מחלקה זו ובנותיה (sub classes), כיצד נגדיר שירות זה?

1. static protected
2. **final protected**
3. private
4. static public

ג. איזה משפט לא נכון? אנו משתמשים בירושה (inheritance) בכדי

1. ליצור התמחויות (specializations) של המחלקה המורשת
2. לשנות התנהגות של מחלקה ע"י דריסת שירותים (Overriding)
3. **לסמל שהמחלקה המורשת הינה אבסטרקטית ולא ניתן להשתמש בה ישירות. נשתמש רק במחלקות המורשות (sub classes)**
4. להוסיף התנהגות ויכולות למחלקה המורשת (להוסיף מידע ומתודות חדשים)

ד. כל אחת מארבעת המתודות בקטע הקוד הבא מוגדרות בתוך מחלקה Test2. סמנו עבור כל אחת אם היא

- מעמיסה את המתודה ב-Test1 (overload)
- דורסת את המתודה ב-Test1 (override)
- לא חוקית

1. **מעמיס**/דורס/לא חוקי
2. **מעמיס**/דורס/לא חוקי
3. מעמיס/דורס/**לא חוקי**
4. מעמיס/**דורס**/לא חוקי

**שאלה 2 - סעיף ד'**

```
public class Test1 {
    public void f1(float a, String b) {...}
}
public class Test2 extends Test1 {
    //code here
    1. public float f1(float a, float b) {...}
    2. public void f1(int a, int b, int c) {...}
    3. public float f1(float p, String q) {...}
    4. public void f1(float a, String b) {...}
}
```

ה. מחלקה המוגדרת final ניתן להגיד עליה כי

1. לא ניתן ליצור מופע שלה, ז"א ליצור עצם מסוג מחלקה זו
2. כל השדות שבה (data members) יוגדרו גם כן final
3. **לא ניתן לרשת ממחלקה זו (לא ניתן לעשות subclasses)**
4. ניתן ליצור בה רק שרותים/מתודות ציבוריים (public)

א. 3  
 ב. 2  
 ג. 3  
 ד. מעט  
 מעט  
 לא  
 דורס  
 ה. 3

## שאלה 3 (15 נקודות)

נתונות שתי המחלקות הבאות:

```
public class A{
    public static String s1="1";
    public String s2 = "2";

    public static class AA{
        public final String s3 = "3";
    }

    protected void f(){
        s2 = "5";
    }
}

public class B extends A{
    public String s2="4";

    public void g(){    }
}
```

א. **[9 נקודות]** ציינו ונמקו אלו מהשינויים הבאים במחלקה B הם תקינים:

- Removing the public modifier (i.e. class B extends A{...})  
השינוי תקין, ניתן להוריד רמת נראות למחלקה מרחיבה
- Adding the member: private String s1="2";  
השינוי תקין. הגדרת משתנה באותו שם "מסתירה" את המשתנה המוגדר ב-A
- Adding the method: public void f(){}  
השינוי תקין. השירות המוגדר ב-B דורס את השירות f המוגדר ב-A.

ב. **[6 נקודות]** הוסיפו שורות קוד למתודה void g() של מחלקה B כך שתדפיס את המחרוזת "12345" באמצעות השדות, המחלקות הפנימיות והמתודות של A ו-B בלבד.

```
public void g() {
    System.out.print(s1); //prints 1
    System.out.print(super.s2); //prints 2
    System.out.print((new A.AA()).s3); //prints 3
    System.out.print(s2); //prints 4
    f(); //changes value of super.s2 to 5
    System.out.print(super.s2); //prints 5
}
```

**שאלה מספר 4 (20 נקודות)****חלק א' – 6 נקודות**

1) מה עושה קטע הקוד הבא? (הטיפוס של shell הוא Shell).

```
foo.addSelectionListener(new SelectionAdapter () {
    public void widgetSelected(SelectionEvent e) {
        shell.close();
    }
});
```

קטע הקוד מוסיף מאזין 'בחירה' ל-widget ש-foo מתייחס אליו. כאשר יבחרו את הפקד (widget) אז החלון בו הוא נמצא ייסגר

2) לאיזה סוג עצם צריך להיות קשור השם foo?

Foo צריך להיות פקד (widget) בעל השירות .addSelectionListener.

**חלק ב' – 8 נקודות**

1) מה עושה השירות הבא?

```
public void bar(String fileName) throws ... {
    FileInputStream fileIn = new FileInputStream(fileName);
    ObjectInputStream in = new ObjectInputStream(fileIn);
    Map map = (HashMap) in.readObject();
}
```

השירות פותח את הקובץ ששמו מועבר ב-filename, עושים deserialization של אובייקטים מתוך הקובץ, ספציפית עצם מסוג Hashmap שהפנייה אליו שומרים במשתנה map

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

`java.lang.RuntimeException`

החריג עלול להיזרק, חריגים מסוג זה הינם `unchecked` ולכן אין צורך להכריז עליהם.

`java.lang.ClassCastException`

חריג זה עלול להיזרק, אנו מבצעים יציקה בשורה האחרונה של השירות ואין ודאות שהיא תצליח. חריג זה מרחיב את `RuntimeException` ולכן אין צורך להכריז עליו

`java.lang.ClassNotFoundException`

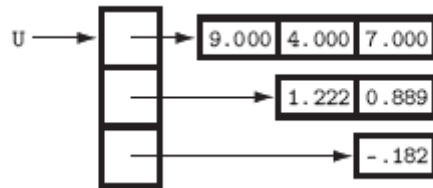
חריג זה לא יכול להיזרק. חריג זה מזרק רק כאשר מנסים לטעון מחלקה בצורה דינאמית. בשירות כל הטיפוסים מוגדרים כך שכבר בעת קומפילציה נדע אם יש בעיה אם אחד מהם. בכל מקרה אין צורך להכריז על חריג זה

`java.io.IOException`

חריג זה עלול להיזרק, אנו מבצעים פעולות על קבצים (כגון פתיחת קובץ) ובמקרים בהם הקובץ לא קיים, לא ניתן לפתוח אותו, יש טעות בקריאה מהקובץ... ייזרק חריג זה (או חריג המרחיב אותו). לכן יש צורך לטפל בחריג זה ויש להכריז עליו

### חלק ג' – 6 נקודות

(1) U הוא מערך של מערכים המייצג מטריצה משולשית עליונה (כמו שנראה בציור)



כיצד ניתן להדפיס את ערך התא שבו יושב המספר 0.889?

`System.out.println(U[1][1]);`

(2) הסבירו מהו עץ מיני-מקס (Game search tree)

Explanation in exercise 4

**שאלה מספר 5 (30 נקודות)**

נתון המנשק הבא המתאר מוצר למכירה במערכת של חנות מסוימת:

```

public interface Product {

    public int productCost( );
        //          מחיר המוצר עצמו

    public int shipCost( );
        //          עלות המשלוח

    public int totalCost( );
        //          מחיר כולל (מוצר + המשלוח)

    public void discount(int amount);
        // amount   הקטן את מחיר המוצר עצמו על ידי חיסור הסכום
        //          חוקי אם המחיר החדש לפחות 0 , ולא גדול מהמחיר המקורי
}

```

להלן דוגמת שימוש במנשק:

```

Product p = new .....
System.out.println(p.productCost( ) + " " + p.shipCost( )
+ " " + p.totalCost( ));
p.discount(2);
System.out.println(p.productCost( ) + " " + p.shipCost( )
+ " " + p.totalCost( ));

```

אם שורת ההדפסה הראשונה מדפיסה:

15 4 19

אזי ההדפסה השנייה תהיה :

13 4 17

המנשק הזה וכל שאר המחלקות והמנשקים בשאלה הזו הם חלק מהחבילה **com.superstore.products**. החבילה אינה מכילה מנשקים וחבילות אחרים פרט לאלה המוזכרים בשאלה (בפרט, שאר הקוד של מערכת המחשב של החנות נמצא בחבילות אחרות, כגון **com.superstore.ui**, **com.superstore.accounting**, **com.superstore.customers**, וכדומה).



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

```
public int productCost();
    @post this == old, $ret >= 0
    @post $ret <= totalCost()

public int shipCost();
    @post this ==old, $ret >= 0
    @post $ret <= totalCost()

public int totalCost();
    @post this ==old, $ret ==productConst() + shipCost()
    @post $ret >= 0

public void discount(int amount);
    @pre 0 <= amount <= productCost()
    @post this.productCost() ==old.productCost()- amount
```

המחלקה MyProduct אמורה לממש את המנשק בצורה פשוטה תוך שימוש בשני שדות בלבד, לייצוג מחיר המוצר ועלות המשלוח. הגוף של כל הפקודות הושמט (אבל הבנאי נתון במלואו).

```
public class MyProduct implements Product {
    protected int productCost;
    protected int shipCost;

    public MyProduct(int productCost, int shipCost) {
        this.productCost = productCost;
        this.shipCost = shipCost;
    }

    public int productCost( ) {
        // omitted
    }

    public int shipCost( ) {
        // omitted
    }

    public int totalCost( ) {
        // omitted
    }

    public void discount(int amount) {
        // omitted
    }
}
```

ב. הגדירו את המשתמר (representation invariant) של המחלקה MyProduct

`productCost >= 0`

`shipCost >= 0`

`totalCost() == productCost + shipCost`

Implementation invariant:

`productCost() == productCost`

`shipCost() == shipCost`

ג. מקרה פרטי חשוב נוגע למוצרים שהלקוח מקבל במקום, ולכן עלות המשלוח היא 0. רוצים להגדיר עבורם מחלקה NoShipProduct. השלימו את ההגדרה באופן שימזער את הצורך בכתיבת קוד חדש, וכך שהלקוח שרוצה ליצור עצם מטיפוס זה יידרש להעביר רק פרמטר יחיד – מחיר המוצר.

```
public class NoShipProduct extends MyProduct {
    public NoShipProduct(int productCost){
        super(productCost,0)
    }
}
```

ד. הנהלת החנות שוקלת מדיניות הנחות שונה עבור המוצרים ללא משלוח (כלומר שינוי בחוזה של הפקודה discount במחלקה NoShipProduct). לכל אחד מהשנויים המוצעים, ציינו אם השינוי תקין מבחינת כללי תיכון בעזרת חוזה. יש לנמק בקיצור. בנוסף, אם השינוי חוקי, כיתבו את החוזה החדש.

שינוי	סמנו את הנכון	נימוק (+ חוזה)
מותר לתת הנחה עד 20% בלבד	תקין / לא תקין	לא תקין (מחזק תנאי קדם)
התוכנה יכולה לקרוא לפקודה תמיד, אבל הפקודה לא תשנה את המחיר בכלל	תקין / לא תקין	לא תקין (מחליש תנאי אחר)

ה. כדי להקטין את התלות בין חלק התוכנה שמשתמש ב Product למחלקות שמממשות אותו MyProduct ו NoShipProduct הוחלט להשתמש במנגנון של בית חרושת ליצירת עצמים מטיפוס Product, על ידי כתיבת מחלקה ProductFactory. יש להשלים את המחלקה, באופן שלקוח שייקרא ל ProductFactory.getProduct יקבל עצם מהמחלקה המתאימה ביותר.

```
public class ProductFactory {

    public static Product getProduct (int productCost, int shipCost) {
        return (shipCost == 0) ? new NoShipProduct(productCost) :
            new MyProduct(productCost,shipCost);
    }

}
```

בנוסף, יש לציין במילים איזה שנויים נדרשים במחלקות NoShipProduct ו MyProduct :

צריך לשנות את נראות הבנאים של MyProduct ושל NoShipProduct מ-public ל-package.

מוצע לכתוב מחלקה חדשה BasketProduct למימוש מוצרים שבנויים מאוסף מוצרים אחרים. למחלקה זאת שדה יחיד – מערך של מוצרים. בשלב זה נדרש בנאי שמאפשר יצירה של מוצר סל כאוסף של שני מוצרים, אך ניתן יהיה להוסיף בהמשך בנאים אחרים. (הערה – הפתרון צריך לאפשר מוצר סל שחלק מהמוצרים שהוא מכיל גם הם מוצרי סל).

ו. השלימו את החלקים המסומנים במימוש של המחלקה BasketProduct. (אין לכתוב את shipCost() ו totalCost()). מותר רק להוסיף קוד בחלקים החסרים המסומנים במלבנים.

```
public class BasketProduct implements Product { <<I think some of the people will
extend MyProduct and not write something new. I think we shouldn't give full points
for that approach, waste of data members (unless they explain themselves really good)
protected Product [ ] productArr;

public BasketProduct(Product p1, Product p2) {
    productArr = new Product[]{p1, p2};
}

public int productCost() {
    int sum = 0;
    for (Product p : productArr) {
        sum += p.productCost();
    }
    return sum;
}

public int shipCost() { // omitted
}
public int totalCost() { // omitted
}
```

<< אני חושב שפה החוזה עדיין תקף, ז"א אתה לא יכול לעשות הנחה ביותר מכמה שהמוצר עולה, ז"א סך מחירי המוצרים בפנים. לכן ערך discount אם תנאי הקדם מתמלאים יהיה חוקי. מכאן נובע שניתן להוריד את הסכום המבוקש מהמוצרים (לחלק אותו ביניהם). אני מציע שנקבל את התשובות הבאות

- א. ממש מורידים את amount מהמחירים של המוצרים (מחולק ביניהם באיזושהיא צורה)
- ב. מגדירים שדה חדש – הנחה ומתחשבים בו בproductCost

```
public void discount(int amount) {
    //need another field to hold the discount (and use it in productCost)
    or:
    productArr[0].discount(amount);
    then, what if there are no products?

    Or,
    final int amnt = amount;
    productArr = copyArrAndIncreaseByOne;
    productArr[productArr.length - 1] = new Product() {
        int cost = amnt;
        ....
        // allow negative cost << I don't think so!!
    }
}
```

ז. הגדירו את המשתמר (representation inv.) של המחלקה BasketProduct

productArr != null

productCost() ==  $\sum_{0 \leq i < \text{productArr.length}-1} \text{productArr}[i].\text{productCost}()$

shipCost() ==  $\sum_{0 \leq i < \text{productArr.length}-1} \text{productArr}[i].\text{shipCost}()$

totalCost() == productCost() + shipCost()