

האוניברסיטה העברית בי"ס להנדסה ולמדעי המחשב

בחינה בקורס מבוא לתכנות מונחה עצמים 67125:

מועד א' - סמסטר ב' - תשס"ו 2006
המרצה: מר תומר הרץ

מילוי פרטים אישיים:

201 מספר מחברת:

040639429 מספר ת"ז (9 ספרות):

7/7

תאריך: 24.7.2006
משך הבחינה: שעתיים

הנחיות:

1. בתום הבחינה יש להגיש שאלון זה יחד עם מחברת הבחינה.
2. חובה לענות על חלק א' במבחן על גבי טופס המבחן עצמו. משקל כל שאלה 5 נקודות (סה"כ 35 נק.).
3. יש לענות על 2 מתוך 3 השאלות בחלק ב' (2 בלבד). משקל כל השאלות בחלק זה זהה (32 נק.).
4. יש לכתוב פתרון לכל אחת מהשאלות שבחרת בחלק ב' במחברת נפרדת. יש לציין על כריכת המחברת לאיזה שאלה המחברת מתייחסת, באופן קריא וברור.
5. הקפידו הקפדה יתרה על סדר וכתב יד קריא. בחינה בלתי קריאה לא תזכה את כותבה במלוא הנקודות.
6. יש למחוק טיוטות באופן ברור.
7. יש להשאיר שוליים.
8. אסור להשתמש בעט אדום.
9. הקפידו על סגנון תכנותי נאות: כתבו באופן מודולרי, אל תכפילו קוד וכו'.
10. תעדו כל תכנית כך שניתן יהיה להבינה בנקל (מותר לתעד בעברית).
11. פתרון מיטבי לשאלה יחשב פתרון פשוט ויעיל ככל האפשר. תוכנית מסורבלת או מסובכת לא תתקבל כתשובה מלאה, גם אם היא נכונה!
12. אין להשתמש במבני נתונים פרט לאלו שמוגדרים בנספח לבחינה. מותר כמובן להשתמש במשתנים בודדים, או במערכים שגודלם אינו תלוי בגודל הקלט.
13. בכל השאלות מותר להניח שהקלט חוקי (ומקיים את תנאי השאלה).

בחירת השאלות לבדיקה:

יש לסמן v ליד השאלות שברצונך שייבדקו: לכל היותר 2 סימני v; לא תיבדקנה שאלות שלא מסומן לידן v.

לשימוש הבודק	מספר שבה השאלה	המחברת מסמן v ליד שאלה מופיעה שברצונך שתיבדק	מספר שאלה
			1
	1	✓	2
	2	✓	3

070634424

חלק א'

כל סעיפי שאלה זו מתייחסים לשפת התכנות Java. חובה לענות על כל הסעיפים. משקל כל הסעיפים זהה. בכל סעיף יש להקיף תשובה אחת בלבד (אם שיניתם תשובתכם יש למחוק את התשובה הקודמת באופן ברור שאינו משתמע לשני פנים).

מונחים בהם נעשה שימוש בחלק זה:

מינוח	מתודה	המרה	בנאי	החלפה	העמסה	תכנון/עיצוב	ממשק	מחלקה	מצביע
פירוש	method	cast	constructor	override	overload	design	interface	class	reference

א. ע"פ עיקרון הבחירה היחידה (single choice principle):

- (i) - אסור לקבל החלטות ביותר ממקום אחד בתוכנית.
- (ii) - אסור להגביל את כמות המידע הפתוחה בפני כל מודול במערכת.
- (iii) - צריך לדאוג שהכרעות בנושא ספציפי תתקבלנה כולן במקום אחד בתוכנית.
- (iv) - מאפשר לדאוג לכך שהוספה של אפשרויות חדשות לתוכנית תגרוור שינוי של מודול יחיד בתוכנית
- (v) - תשובות (iii) ו-(iv) נכונות

ב. תבנית העיצוב Decorator מאפשרת

- (i) - להגדיר מחלקה שיורשת ממחלקה אחת, ומכילה מחלקה אחרת.
- (ii) - לקשט מידע גולמי ע"י שימוש במחלקה אשר מרחיבה את הממשק של המחלקה שמייצרת את המידע הגולמי.
- (iii) - לבנות פילטרים שמקשטים את הנתונים הגולמיים, שהטיפול בפילטרים הוא בהכרח שונה מהטיפול באובייקטים שאותם הם מקשטים.
- (iv) - יוצרת שקיפות בשימוש בין המחלקה הגולמית לבין המחלקה המקשטת.
- (v) - תשובות (i) ו-(iv)
- (vi) - תשובות (ii) ו-(iv)

ג. השאלה הבאה מתייחסת לקטעי הקוד הבאים:

```
public static Command createCommand (CommandNames commandName) {
    switch (commandName) {
        case RUN:
            return new RunCommand();
        case HALT:
            return new HaltCommand();
        default:
            return null;
    }
}
```

```
public void executeCommand (CommandNames commandName) {
    Command c = createCommand (commandName);
    if (c instanceof RunCommand)
        ((RunCommand) c).run();
    if (c instanceof HaltCommand)
        ((HaltCommand) c).halt();
}
```


201
040639429

אלו מהטענות הבאות נכונות:

- (i) - בהנחה שה - new היחיד לאובייקטים מסוג Command נמצא בפונקציה createCommand היא מממשת את עיקרון ה - single-choice.
- (ii) - הפונקציה executeCommand כתובה באופן שאיננו עושה שימוש בעקרון הפולימורפיזם.
- (iii) - שימוש בפקודה instanceof הוא שגוי מיסודו, ומוטב לא להשתמש בפקודה כלל.
- (iv) - המחלקה RunCommand נורשה מהמחלקה Command.
- (v) - תשובות (i), (iii) ו-(iv).
- (vi) - תשובות (i), (ii) ו-(iv).

ד. נתון קטע הקוד הבא:

```
A a1 = new B();
a1.execute();
A a2 = new C();
a2.execute();
```

אילו מהטענות הבאות נכונות:

- (i) ✓ - יתכן כי המחלקה A היא מחלקה אבסטרקטית והמחלקות B ו-C יורשות אותה.
- (ii) - לא יתכן כי המחלקה A אבסטרקטית שאם כן, לא היינו יכולים ליצור מצביעים מסוגה.
- (iii) ✓ - יתכן כי A הינו ממשק אשר המחלקות B ו-C מממשות.
- (iv) - חייב להיות קשר הורשה (extends) בין המחלקות A ו-B.
- (v) - תשובות (i) ו-(iv).
- (vi) - תשובות (i) ו-(iii).

Nodes

ה. נרצה להגדיר מחלקה שתציג גרף לא מכון. הגרף יכיל צמתים שלכל אחד שתי תכונות (data members): (1) מזהה (2) נתון - שתיהן מוגדרות עם הרשאת גישה protected. למחלקה אין שיטות כלל.

אילו מהטענות הבאות נכונות:

- (i) - בד"כ נכון להגדיר את המחלקה Node כ-Nested Class.
- (ii) - בד"כ נכון להגדיר את המחלקה Node כ-~~Inner~~ Inner Class.
- (iii) - אם ממשים את הגרף בעזרת רשימת שכנויות עדיף ש Node יהיה Nested Class, אם בעזרת מטריצת שכנויות עדיף ש Node יהיה Inner class.
- (iv) - אם ממשים את הגרף בעזרת רשימת שכנויות עדיף ש Node יהיה Inner Class, אם בעזרת מטריצת שכנויות עדיף ש Node יהיה Nested Class.

ו. תבנית העיצוב Composite מממשת את עקרונות עיצוב מונחה העצמים הבאים:

- (i) ✓ - תכנת לממשק ולא למימוש.
- (ii) - עקרון הבחירה היחידה.
- (iii) ✓ - עקרון הפתיחות-סגירות.
- (iv) - תשובות (i) ו-(iii) נכונות.
- (v) - כל התשובות נכונות.

ז. מה מהבאים נכון:

- (i) - איטרטור תמיד יהיה Inner class, של מבנה הנתונים שעליו הוא מוגדר.
- (ii) - מבנה הנתונים מגדיר את סדר המעבר של האיטרטור על האברים, אלא אם האיטרטור משתמש במבנה עזר.
- (iii) - איטרטור תמיד יצטרך גישה מיוחדת, למבנה הפנימי של מבנה הנתונים עליו הוא עובר.
- (iv) - ב' וג' נכונים.
- (v) - אף תשובה איננה נכונה.

חלק ב'

בחלק זה עליך לבחור 2 מתוך 3 השאלות הבאות. משקל כל השאלות זהה. בכל השאלות ניתן להשתמש בכל מבני הנתונים אשר מצורפים לבחינה בנספח א'. מותר להשתמש אך ורק בפעולות המופיעות בממשק המצורף לכל מבנה נתונים. קראו את ההנחיות המופיעות בסוף כל שאלה, לפני פתירתה.

שאלה 1

חברת מעוף-נעים מפתחת תוכנה לניהול דלפקי בידוק (צ'ק-אין) עבור שדות תעופה. בכל שדה תעופה יש אוסף מוגדר וסופי של דלפקי בידוק. בהתאם לצרכים המשתנים של חברות התעופה השונות, מקצה שדה התעופה לכל חברת תעופה אוסף של דלפקים. בתקופות של לחצי שיא בשדה, לעתים מקצה שדה התעופה את אותו דלפק לשתיים או יותר חברות תעופה. כידוע, בכל דלפק בידוק יתכן שיווצר תור של נוסעים הממתנים לקבל שירות.

תוכנת דלפקי הבידוק אמורה לשלוט בהשמה של נוסעים לדלפקי הבידוק. בכל פעם שנוסע מגיע הוא נשלח לדלפק שבו מספר הנוסעים הממתנים הוא מינימלי. בשאלה זו עליך להגדיר את המחלקה Airline אשר אחראית על ניהול דלפקי הבידוק בשדה התעופה.

לרשותך המחלקה הבאה:

Public class Passenger {	המחלקה מייצגת נוסע שאמור לטוס משדה התעופה. לכל נוסע יש שם, מספר דרכון ואת שם חברת התעופה שעימה הוא אמור לטוס.
//data members private String _name; private long _passportNumber; private String _airline;	
public Passenger(String name, long pasNum, String airline){ _name = name; _passportNumber = pasNum; _airline = airline; }	בואי איתחול
public void getAirline(){ return _airline; }	הפעולה מחזירה את שם החברה שעימה אמור הנוסע לטוס.

א. בחר והגדר מבני נתונים מתאימים למימוש הממשק המופיע מטה. מבני הנתונים חייבים להבטיח מימוש יעיל ככל הניתן של ניהול השמת הנוסעים לדלפקי הבידוק. הסבר בקצרה מדוע בחרת במבנה הנתונים הספציפי למימוש המחלקה.

ב. ממש את הפעולות שמסומנות בכתב מודגש.

Public class AirportCounterManager{	
//data members	עליך להגדיר מבני נתונים מתאימים לייצוג הבעיה
public AirportCounterManager (int numCounters){ ... }	בואי אשר מאתחל את מבני הנתונים המתאימים. הפרמטר מייצג את מספר הדלפקים הקיימים בשדה התעופה הנתון.
public void assignCounter(String airline, int counterNumber)	פעולה אשר מקצה את הדלפק שמספרו counterNumber לחברה בשם airline.
public void assignPassenger (Passenger p){ ... }	פעולה אשר מצרפת את הנוסע לדלפק של חברת התעופה שאיתה הוא טס שבו מספר הנוסעים הממתנים הוא מינימלי. אם מספר הנוסעים המתאימים בכמה דלפקים זהה, הנוסע יתווסף לאחד מהדלפקים הללו.

המשך בעמוד הבא...

ג. הסבר בקצרה את מדוע בחרת דווקא במבנה הנתונים הספציפי למימוש המחלקה בסעיף א'.

ד. נתח את זמן הריצה של הפעולה assignPassenger שאותה מימשת בסעיף ב'.

הנחיות:

- מותר להגדיר מחלקות ושיטות עזר כרצונכם.
- בהגדרת מימוש המחלקה מותר להשתמש אך ורק במבני נתונים אשר מופיעים בנספח א', וכן במשתנים פרימיטיביים ובמערכים.
- מימוש הפעולה של צירוף נוסע לדלפק חייב להיות יעיל ככל הניתן, יעילות ההשמה תלויה כמובן בבחירת מבני הנתונים שבעזרתם תייצגו את דלפקי הבידוק.

שאלה 2

צבא ההגנה של מדינת ליליפוט החליט לבנות כלי אוטומטי שיצנזר את מכתביו של גוליבר אל אנגליה מולדתו. בעת הפעלת הכלי יכולים הגמדים להזין רשימה של מילים מצונזרות כגון "ליליפוט", "גמדים" וכו'. גוליבר כותב את מכתביו כקבצי טקסט. כלי הצנזור קורא את הקובץ ויוצר קובץ חדש אשר במקום כל מילה מצונזרת מופיעה מחרוזת באותו האורך שכולה בנויה מהאות X.

לדוגמא עבור המילה "ליליפוט" תופיע המחרוזת "XXXXXXXX".

עליכם לממש את כלי הצנזור האוטומטי של גמדי ליליפוט המתואר מטה:

<pre>public class LiliputCensorReader extends BufferedReader {</pre>	
<pre>//data members private ??? _censorList;</pre>	עליכם לבחור מבנה נתונים רלוונטי למימוש רשימת הצנזורה
<pre>public LiliputCensorReader(Reader in) throws IOException { super(in); <i>List</i> _censoredReader = null; }</pre>	בנאי אשר מקבל כקלט שם של צינור קלט שאותו יש לצנזר.
<pre>public void setCensorList(String censorListFileName) throws IOException{ ... }</pre>	הפעולה קוראת את רשימת המילים המופיעה בקובץ, ושומרת את רשימת המילים במבנה הנתונים שבחרתם לעיל. ניתן להניח כי קובץ רשימת המילים מכיל מילה בכל שורה.
<pre>public String readLine() throws IOException { ... }</pre>	פונקציה אשר קוראת שורה מצינור הקלט ומחזירה אותה כמחרוזת אשר כל מופע של מילה שנמצאת ברשימת המילים המצונזרות מוחלף במחרוזת באותו האורך אשר מכילה את האות X.

המשך בעמוד הבא ...

בהנתן כלי הצנזורה האוטומטי, כתבו הגמדים את התוכנית הבאה אשר מצנזרת את מכתביו של גוליבר:

<pre> public class GulliverCensor { public static void main(String[] args) throws IOException{ LiliputCensorReader cIn = new LiliputCensorReader(new FileReader("GulliverLetter.txt")); FileWriter fileOut = new FileWriter("GulliverCensLetter.txt"); cIn.setCensorList("censorList.txt"); String line; while((line=cIn.readLine())!=null) fileOut.write(line + "\n"); fileIn.close(); fileOut.flush(); fileOut.close(); } </pre>	התוכנית משתמשת בכלי הצנזורה האוטומטי בכדי לייצר תחת שם קובץ הפלט את הקובץ המצונזר. הקובץ censorList.txt מכיל את רשימת המילים שאותן יש לצנזר, בפורמט המתואר לעיל.
---	--

לדוגמא: אם רשימת המילים המצונזרות (הקובץ censorList.txt) כוללת את המילים הבאות:
Island, Liliputs אזי עבור הקלט הבא:

Hi, its Gulliver
If you get this letter please come and save me
I'm on an **Island** and have been captured by some creatures called
Liliputs
Please help me!

יתקבל הפלט הבא:

Hi, its Gulliver
If you get this letter please come and save me
I'm on an **xxxxxxx** and have been captured by some creatures called
xxxxxxx
Please help me!

א. בחרו מבנה נתונים לשמירת רשימת המילים המצונזרות, אשר יאפשר לצנזר מילים בצורה יעילה ככל הניתן. נמקו בקצרה את בחירתכם.

ב. ממשו את הפעולה setCensorList תוך שימוש במבנה הנתונים שבחרתם בסעיף א'.

ג. ממשו את הפעולה readLine אשר מחזירה שורה מצונזרת מן הקובץ.

ד. הסבירו בקצרה מהי תבנית העיצוב שמממשת המחלקה שכתבתם, ומה היתרון שבה?

הנחיות:

- שימו לב כי כל הפעולות זורקות IOException ולכן אין צורך לספל מפורשות בחריגות.
- ניתן להניח כי הקלט תקין, כלומר גם רשימת המילים המצונזרות וגם הקובץ שאותו מצנזרים ניתנים בפורמט תקין.
- שימו לב כי בפונקציה readLine עליכם לספל במקרה שבו נקרא null. במקרה כזה על הפונקציה להחזיר null.
- ההשוואה בין מחרוזות למטרת צנזורה צריכה להיות case-insensitive. לצורך כך אפשר להשתמש בפונקציה toLowerCase של המחלקה String אשר מחזירה את המחרוזת הנתונה באותיות קטנות.

שאלה 3

נתון הממשק הבא :

<pre>public interface Function {</pre>	ממשק זה מתאר פונקציה לא אנליטית שבה הקשר בין $y - 7 \times$ הינו שרירותי.
<pre> public boolean add(Number <i>Integer</i> x, Integer y);</pre>	מוסיפה את הערך x לתחום של הפונקציה, ומתאימה לו את הערך y. אם הערך x קיים בתחום, אינה עושה דבר. המתודה מחזירה ערך $true$ אם ורק אם x הוסף לתחום, $false$ אחרת.
<pre> public Number <i>Integer</i> evaluate(Integer x);</pre>	מחזירה את הערך המתאים ל x. ניתן להניח כי x קיים בתחום.

בשאלה זו עליך לתכנן ולממש שלוש מחלקות אשר מממשות את הממשק הנתון לעיל:

1. **GeneralFunction** - מחלקה שתתאר פונקציה כללית.

2. **UniqueFunction** - מחלקה שתתאר פונקציה חד-חד ערכית. פונקציה הינה חד-חד ערכית אם לכל $x_1 \neq x_2$ המקיימים $x_1 \neq x_2$ מתקיים $evaluate(x_1) \neq evaluate(x_2)$.

3. **ComplexFunction** - מחלקה שתתאר פונקציה שהיא הרכבה של מספר פונקציות.

להלן מספר תכונות של פונקציית ההרכבה:

(i) לפונקציה זו לא ניתן להוסיף איברים לתחום, כלומר הפעולה `add` איננה נתמכת, ובמקרה שהיא נקראת יש לזרוק חריגה מסוג `UnsupportedOperationException`.

(ii) נגדיר את סדר ההרכבה : בהנתן שהפונקציה $g(x)$ היא הראשונה בסדר והפונקציה $f(x)$ היא השנייה בסדר, ההרכבה תוגדר להיות $f(g(x))$.

(iii) בעת בניית המחלקה המתארת פונקציה שהיא הרכבה של מספר פונקציות, עליכם לכלול את הבנאי הבא:

```
public ComplexFunction(LinkedList List)
```

סדר האיברים ברשימה הוא הסדר המבוקש של הרכבת הפונקציות.

א. הציעו Design למערכת, ע"פ העקרונות שנלמדו בכיתה. שרטטו תרשים שמבהיר את היחס בין המחלקות השונות בתוכנית. נמקו בקצרה את העיצוב שבחרתם. עליכם לתאר את היחסים בין המחלקות וכן את הנתונים והממשק של כל אחת מהמחלקות.

ב. כיתבו קוד המממש את ה- Design שהצעתם.

ג. איך הייתם מרחיבים את המימוש שלכם כך שיתמוך בהוספת איברים לתחום של פונקציות מורכבות?

בהצלחה!

עבודה 2

א. מבנה הנתונים המבנה קבוצת מילים שחבר

כ. Set. מיון מילים יחס 130 בין המילים (הסדר

המילים) ניתן מילים את הקיבוצים עם יציאה

SortedSet. מילים את התכונה String מילה genericity

// data Members

private Set<String> - censorList;

~~private Set<String> - censorList;~~

```
public void setCensorList(  
    String censorListFileName)  
    throws IOException {
```

מילה בין מילים

- censorList = new SortedSet<String>();

```
Reader words = new BufferedReader(  
    new FileReader(  
        censorListFileName));
```

String word;

```
while ((word = words.readLine()) != null) {
```

```
    - censorList.add(word.toLowerCase());
```

```
words.close();
```

```
}
```

//

מחזור ק Set

אם לא קורה אקסצפציה

ה add

②

Public String ReadLine() throws IOException {

String line = ^{super} in.readLine();

if (line == null)

return null;

String output = new String();

```
Scanner words = new Scanner(line);
```

while (words.hasMoreNext()) {

String word = words.next();

```
if (-censorList.Contains(low word.ToLowerCase))
```

output = output + stringToXXX(word);

else

Output = output + word;

if (words.hasNext())

$$\text{output} = \text{output} + " ";$$

३

Return Output:

3

```
private String Geting to XXX (String word) { // } 2015
```

String result = new String(); // e-x String s = "Hello"

```
for (int i=0; i<word.length(); i++)
```

result = result + "X";

return result;

3

③ המעצב מממש את תבנית הויזוק Decorator.
היא מממנת את הממשק של bufferedReader
המלווה, מאוספה לו בוקצונות קצונה של סילר
(צנלונה).

היתרון:

א. המשתמש עם ציבור של עצמים אחדים מממשק חדש.

ב. יש חברים שהעלף את סילר הצנלונה בסילר אחר.

השינוי המעצב בקוד הוא מנצחי (סוג הצנלונה - single-choice)

ג. תכנות ממשק, ועליו עמיתים, (צומח עליו).

ד. ניתן לשכנע את הסילר עם כל סילר יחד.

נניח סילר שהוא אופיות מכלולות לוגיות,

ה. עומת עליו בקוד של הנישנים (reusability).

(composability).

הערה: במתחזה (ב) השתמשתי בקנייט לשונה מה super

בני זקני יא השונה הרשיה בקטל. הושלמה לי ביתה

מסביב הכורה עמית ייכיה למבוי ציבור בקטל עליו.