

בחינה בתוכנה 1

גרסה 1

מסטר ב' תשע"ה, מועד ב', 30 באוגוסט 2015
יעל אמסטרדמר, סיון טולדו, דביר נתנאלי

משך הבחינה שלוש שעות.

יש להניח שהקוד שמופיע במבחן הוא בגרסה 8 של ג'אווה.

יש לסמן את התשובה הטובה ביותר בתשובון. לא יינתן ניקוד על סימון תשובה בטופס הבחינה או במחברת הבחינה. משקל כל שאלה 4 נקודות.

יש לנמק כל תשובה בטופס הבחינה. נימוק חסר או לא נכון עלול לגרום לאי קבלת נקודות על שאלה גם אם התשובה המסומנת היא הטובה ביותר. המקום המיועד לנימוקים בגוף הבחינה מספיק לתשובות תמציתיות.

יש לצרף את טופס המבחן למחברת הבחינה. מחברת ללא טופס עזר תפסל. תשובות במחברת הבחינה לא תיבדקנה.

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.

© כל הזכויות שמורות למחברים. מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן במאגר מידע, בכל דרך שהיא, בין מכנית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

בהצלחה!

שאלה 1

באיזה מחלקת אוסף נשתמש כדי לאחסן אובייקטים ללא חזרות כאשר אין חשיבות לסדר וכאשר אנחנו מעוניינים שהשירות contains יפעל ביעילות המירבית (בממוצע, לא במובן של worst case)?

- א. HashSet
- ב. LinkedList
- ג. TreeSet
- ד. ArrayList

Comment[1s]: יעילות גבוהה בבדיקת שייכות אבל לא שומר על סדר כלשהו.

השאלות הבאות מתייחסות לקוד הבא ולחזרה החלקי שמוגדר עבורו. המימוש של הבנאים אינו נתון.

```
public class Point2D {
    /**
     * @imp_inv x+y != 0
     */
    private int y, x;
    public Point2D() { ... }
    public int divBySum(int z) { return z/(x+y); }
    public void moveX(int d) { x += d; }
    public void moveY(int d) { y += d; }
    public int x() { return x; }
    public int y() { return y; }
}

public class Printable2D extends Point2D {
    public Printable2D() {...}
    public void print() {...}
}
```

שאלה 2

בהנחה שהבנאי של Point2D תקין (מבטיח את קיום המשתמר), איזה מהשירותים האחרים של Point2D אינם תקינים בהינתן המשתמר? בשאלה זו נחשיב חלוקה באפס בתור תקלה.

- א. divBySum
- ב. moveX, MoveY
- ג. x, y
- ד. divBySum, moveX, moveY

Comment [2s]: נימוק: עלולים להזיז את x ו/או y כך שערך הסכום יהיה אפס ויפר את המשתמר.

נימוק:

שאלה 3

נניח שהשירותים הבעייתיים בשאלה הקודמת תוקן/תוקנו, ושהמחלקה Printable2D נכתבה תוך שימוש נכון בחוזים. מייד לאחר יצירה של עצם מהמחלקה Printable2D (כלומר מייד לאחר הקריאה לבנאי), המשתמר של Point2D

- א. מתקיים
- ב. אינו מתקיים (אף פעם)
- ג. אולי מתקיים ואולי לא

Comment [3s]: נימוק: מחלקה מרחיבה חייבת לכבד את המשתמרים והחזרה של מחלקת הבסיס.

נימוק:

עמוד 3 מתוך 16 מספר סידורי: _____ מספר ת"ז: _____

שאלה 4

```
Stream<String> names = Arrays.asList("xxx", "yyy", "aaa").stream();
names.sorted((a, b) -> b.compareTo(a) )
    .forEach( System.out::println );
```

הקוד

- לא יתקמפל בגלל שחסרה הגדרת טיפוסים ל-b, a
- לא יתקמפל שחסרים סוגריים מסולסלים מסביב ל-b.compareTo(a)
- תשובות א' ו-ב' נכונות (כלומר יש בקוד לפחות את שתי הבעיות הללו)
- יתקמפל ואם ירוץ בלי לזרוק חריג, ידפיס את איברי הקבוצה בסדר לקסיקוגרפי
- יתקמפל ואם ירוץ בלי לזרוק חריג, ידפיס את איברי הקבוצה בסדר לקסיקוגרפי הפוך

[4s]Comment: נימוק: אין בקוד באגים, וה-comparator משקף מסדר לקסיקוגרפי הפוך.

נימוק:

שאלה 5

```
public class Test {
    private class Ex extends java.io.IOException {}

    private Ex e = new Ex();

    public void f() throws Ex {
        throw e;
    }
}
```

הקוד הנתון

- לא יתקמפל מכיון שהשירות זורק חריג ממחלקה פרטית; אם נשנה ל-private public הבעיה הזאת תסתדר
- לא יתקמפל מכיון שהשירות זורק חריג ממחלקה פרטית; אם נשנה ל-throws Ex - גם א' וגם ב' נכונות (כלומר הקוד לא יתקמפל, אבל כל אחד מהתיקונים המוצעים יפתור את הבעיה)
- לא יתקמפל כי אי אפשר לזרוק חריג שהוא שדה במחלקה; צריך לייצר עצם שמייצג את החריג בפקודה שזורקת אותו, כלומר לשנות את ל-throw new Ex() throws Ex
- הקוד יתקמפל והוא יתקמפל גם אם נמחק את throws Ex
- הקוד יתקמפל כפי שהוא, אבל לא יתקמפל אם נמחק את throws Ex

[5s]Comment: ניתן ניקוד גם על התשובה (הלא נכונה) הזאת מכיון שההנחה (הלא נכונה) שאסור להכריז על זריקת חריג פרטי היא הנחה סבירה.

[6s]Comment: נימוק: הקוד תקין כפי שהוא ויש להצהיר על זריקת החריג כי אינו RuntimeException או Error.

נימוק:

שאלה 6

```
Set< Set<String> > x = ...;  
for (String s: x) System.out.printf("string is %s\n",s);
```

בהנחה שהקוד שמאתחל את x תקין (עובר קומפילציה ולא זורק חריג בזמן ריצה). שאר הקוד

Comment[7s]: נימוק: האיברים של x אינם מחרוזות אלא קבוצות של מחרוזות ולכן לולאת ה- for אינה תקינה.

- א. לא יתקמפל
- ב. יתקמפל אבל יזרוק NullPointerException אם s הוא null
- ג. יתקמפל ולא יתכן ש-s יהיה null כי אי אפשר להכניס null ל-Set
- ד. אף תשובה אחרת אינה נכונה

נימוק:

שאלה 7

נתונה המחלקה

```
public abstract class C {  
    public abstract void foo (String s);  
}
```

איזה משני השירותים הבאים ניתן להוסיף למחלקה כך שהקוד יתקמפל ללא שגיאות?

```
public abstract int foo (String s2); /* A */  
public void foo () { foo(null); } /* B */
```

- א. אי אפשר להוסיף אף אחד מהם
- ב. רק את השירות המסומן A
- ג. רק את השירות המסומן B
- ד. אפשר להוסיף את A ואפשר להוסיף את B אבל לא את שניהם ביחד
- ה. אפשר להוסיף כל אחד מהם או להוסיף את שניהם ביחד
- ו. אפשר להוסיף את A וגם את B אבל לא אחד מהם לבד

Comment[8s]: נימוק: B הוא העמסה בגלל העדר הארגומנט. A משנה רק את הטיפוס המוחזר וזה לא מהווה העמסה.

נימוק:

נתונות המחלקות הבאות:

```
public class A {
    protected Integer getSizeofLargestFile (HashSet<String> files)
        throws IOException {...}
}

public class B extends A {
    // 1
    private Integer getSizeofLargestFile (HashSet <String> files,
        String folderName) throws Exception {...}

    // 2
    public Integer getSizeofLargestFile (HashSet < String > mySpecialist)
        throws FileNotFoundException {...}
}

public class C extends A {
    // 3
    protected int getSizeofLargestFile (HashSet < String > files)
        throws IOException {...}

    // 4
    protected Integer getSizeofLargestFile (HashSet < String > files)
        throws Exception{...}
}
```

שאלה 8

אלו מהמתודות (הממוספרות על פי ההערה שמעליהן) יכולות להיכלל במחלקה B מבלי לגרום לשגיאת קומפילציה?

- א. 1
- ב. 2
- ג. 1 וגם 2
- ד. אף אחת מהן

Comment [9s]: 1 הוא העמסה (תוספת ארגומנט) ו-2 הוא דריסה עם הרחבת נראות וצמצום חריג.

נימוק:

שאלה 9

אלו מהמתודות (הממוספרות על פי ההערה שמעליהן) יכולות להיכלל במחלקה C מבלי לגרום לשגיאת קומפילציה?

- א. 3
- ב. 4
- ג. 3 וגם 4
- ד. אף אחת מהן

Comment [10s]: 3 אינו מעמיס אבל משנה את טיפוס הערך המוחזר לטיפוס שאינו תת-טיפוס; זה אסור. 4 גם הוא אינו מעמיס אבל מרחיב את החריג; שוב אסור.

נימוק:

שאלה 10

נתון קוד שתי המחלקות הבאות

```
public abstract class Agent1 {
    public abstract void actOn(Object o);

    public void repeat(Object o, int i) {
        for (int j = 1; j <= i; j++) {
            actOn(o);
        }
    }
}
```

```
public abstract class Agent2 {
    public abstract void actOn(Object o, int i);

    public void repeat(Object o, int i) {
        for (int j = 1; j <= i; j++) {
            actOn(o, j);
        }
    }
}
```

ניתן להבחין בשכפול קוד מסוים בין המחלקות, ונרצה למנוע אותו במידת האפשר. מה מבין האפשרויות הבאות יביא לחיסכון קוד מקסימלי, אך עדיין ישמור על השימוש המקורי בכל אחת מן המחלקות? (למשל, לא ידרוש שינוי של מחלקות קיימות היורשות מהן, או של קוד לקוח קיים המשתמש בהן)

- א. נהפוך את Agent1 למחלקה היורשת מ-Agent2, נוסיף ב-Agent1 מימוש ל-actOn(o, i) המשתמש ב-actOn(o) ונמחק ממנה את repeat.
- ב. נהפוך את Agent2 למחלקה היורשת מ-Agent1, נוסיף ב-Agent2 מימוש ל-actOn(o) המשתמש ב-actOn(o, i) ונמחק ממנה את repeat.
- ג. נכתוב מחלקה שלישית ממנה יירשו שתי המחלקות, ונעביר אליה את המימוש ל-repeat.
- ד. נכתוב מחלקה שלישית שתירש משתי המחלקות ותספק מימוש משותף ל-repeat ול-actOn.

נימוק:

Comment[11s]: נימוק: פתרון אפשרי ומינימליסטי.

שאלה 11

```
public class MyClass {
    public void myMethod(Other a) {
        a.otherMethod();
    }
}
```

בהינתן המחלקה שלעיל, מה יבטיח כי בזמן ריצה יקרא השירות otherMethod שהוגדר במחלקה Other? הניחו כי קיים במחלקה Other שירות בשם זה שאינו מקבל ארגומנטים.

- א. נגדיר את otherMethod כ-final
- ב. נגדיר את otherMethod כ-private
- ג. נדאג שבמחלקה Other לא יהיו שירותים נוספים
- ד. תשובות א' וב' נכונות

[12s]Comment: נימוק: הגדרה כ-final מונעת דריסה ולכן השירות הנתון יופעל גם במחלקות יורשות.

נימוק:

שאלה 12

מה הפלט של הקוד הבא?

```
public class Test11 {
    public static void main(String a[]) {
        Integer i1 = new Integer(127);
        Integer i2 = new Integer(127);
        Long l = new Long(127);
        System.out.print((i1 == i2) + " ");
        System.out.print((i1.equals(i2)) + " ");
        System.out.print((i1.equals(l)) + " ");
    }
}
```

- א. false true true
- ב. true true true
- ג. false true false
- ד. false false true
- ה. false false false
- ו. true true false
- ז. true false true
- ח. true false false
- ט. הקוד לא יתקמפל

[13s]Comment: תשובה לא נכונה שקיבלה ניקוד בגלל שהיא סבירה למדי (טיפוסים שונים, אבל ערכים שונים ובלתי ניתנים לשינוי בהמשך).

[14s]Comment: נימוק: עצמים שונים, ערך וטיפוס זהים, טיפוסים שונים.

נימוק:

עמוד 8 מתוך 16 מספר סידורי: _____ מספר ת"ז: _____

השאלות הבאות מתייחסות לקוד הבא.

```
public class Base {
    private static boolean goo = true;
    protected static boolean foo() {
        goo = !goo;
        return goo;
    }
    public String bar = "Base:" + foo();
    public static void main(String[] args) {
        Base base = new Sub();
        /**/
    }
}
public class Sub extends Base {
    public String bar = "Sub:" + foo();
}
```

בכל אחד מהסעיפים הבאים מוחלף סימון ה- `/**/` בפונקציית ה-`main` בקטע קוד, ובסעיף ה-`main` גם נוסף קוד למחלקה `Sub`. הנכם מתבקשים לציין מהו הפלט של הרצת פונקציית ה-`main`.

שאלה 13

`System.out.println(base.bar);` מוחלף ב-`/**/`

- א. יודפס `Base:true` ב. יודפס `Base:false` ג. יודפס `Sub:true`
- ד. יודפס `Sub:false` ה. תתקבל שגיאת קומפילציה ו. תתקבל שגיאה בזמן ריצה

נימוק:

שאלה 14

`System.out.println("Base:" + goo);` מוחלף ב-`/**/`

- א. יודפס `Base:true` ב. יודפס `Base:false` ג. יודפס `Sub:true`
- ד. יודפס `Sub:false` ה. תתקבל שגיאת קומפילציה ו. תתקבל שגיאה בזמן ריצה

נימוק:

[15s]Comment: הניסוח תוקן במהלך הבחינה ל-"שאלות 16 ו-17"

[16s]Comment: `goo` משנה את ערכו פעם אחת והשדה `bar` הוא במחלקה `Base`.

[17s]Comment: `goo` מחליף את ערכו פעמיים, באתחול במחלקה `Base` ובאתחול במחלקה `Sub`.

עמוד 9 מתוך 16 מספר סידורי: _____ מספר ת"ז: _____

שאלה 15

```
Sub sub = base;
System.out.println(sub.bar);
```

*** / מוחלף ב- /

א. יודפס Base:true ב. יודפס Base:false ג. יודפס Sub:true

Comment [18s]: השמה לתת-טיפוס.

ד. יודפס Sub:false ה. תתקבל שגיאת קומפילציה ו. תתקבל שגיאה בזמן ריצה

נימוק:

שאלה 16

```
protected static boolean foo() {
    return false;
}
```

מוסיפים ל-Sub את השירות {

```
System.out.println("Base:" + base.foo());
```

*** / מוחלף ב- /

Comment [19s]: המתודה foo ב-Sub מסתירה את המתודה foo ב-Base. לכן נקראת פעמיים, באתחול השדה bar ובשורה הנתונה.

א. יודפס Base:true ב. יודפס Base:false ג. יודפס Sub:true

ד. יודפס Sub:false ה. תתקבל שגיאת קומפילציה ו. תתקבל שגיאה בזמן ריצה

נימוק:

שאלה 17

```
public String foo(String s) {
    return bar;
}
```

מוסיפים ל-Sub את השירות {

```
System.out.println(base.foo(null));
```

וגם / *** / מוחלף ב- /

א. יודפס Base:true ב. יודפס Base:false ג. יודפס Sub:true

Comment [20s]: foo עם ארגומנט לא מוגדרת עבור הטיפוס Base.

ד. יודפס Sub:false ה. תתקבל שגיאת קומפילציה ו. תתקבל שגיאה בזמן ריצה

נימוק:

שאלה 18

הטיפוס double יכול לייצג מספרים אי-שליליים שבין 10^{-300} ל- 10^{300} . לא כל המספרים ניתנים לייצוג, אבל אם x הוא בתחום הזה ולא ניתן לייצוג, אז יש מספר ניתן לייצוג בין x ו- $2x$ (יש יותר מספרים ניתנים לייצוג אבל התנאי הנתון מספיק עבור השאלה הזאת). בהינתן ההנחות הללו, הקוד הבא, והתנאים הנתונים בקוד, האם יתכן שהשירות SmallMath.exp לא יוכל למלא את תנאי האחר שלו (e המופיע בתנאי האחר הוא מספר אוילר, בערך 2.718)? אפשר להניח שכל הקוד מקיים את כל תנאי החוזה (אין הפרות חוזה או באגים).

```
public class SmallInt {
    private final int n;
    // @pre 0 <= n <= 10
    public SmallInt(int n) { this.n = n; }
    public int value() { return n; }
}

public class SmallMath {
    // @pre none
    // @post e^n ≤ $ret ≤ 2e^n
    public static double exp( SmallInt n ) { ... }
}
```

- א. לא. מובטח שהקלט יהיה קטן מספיק כך שהשירות תמיד יוכל למלא את תנאי האחר.
- ב. כן, יתכן שהשירות יקבל קלט גדול מ-10 ובפרט מספיק גדול כך שהאקספוננט יהיה גדול מהמספר המקסימלי שאפשר לייצג ב-double, אבל רק אם הארגומנט של השירות יהיה ממחלקה שמרחיבה את SmallInt.
- ג. כן, יתכן שהשירות יקבל קלט גדול מ-10 ובפרט מספיק גדול כך שהאקספוננט יהיה גדול מהמספר המקסימלי שאפשר לייצג ב-double, אבל רק אם הארגומנט של השירות יהיה ממחלקה אנונימית שמרחיבה את SmallInt.
- ד. כן, יתכן שהשירות יקבל קלט גדול מ-10 ובפרט מספיק גדול כך שהאקספוננט יהיה גדול מהמספר המקסימלי שאפשר לייצג ב-double, וזה עלול לקרות גם אם הארגומנט של השירות יהיה מהמחלקה SmallInt.

נימוק: _____

Comment[21s]: נימוק: הארגומנט עלול להיות ממחלקה שמרחיבה את SmallInt דוֹרֵסֵת את value.

שאלה 19

נתון הקוד הבא:

```
public static void main(String[] args) throws Exception {
    Scanner scanner = null;
    do {
        scanner = new Scanner(new File("dictionary.txt"));
        System.out.println(scanner.next());
    } while (scanner.hasNext());
    if (scanner != null) {
        scanner.close();
    }
}
```

איזו מן הטענות הבאות לגבי הקוד הנ"ל אינה נכונה?

- א. אם הקובץ dictionary.txt אינו קיים, התכנית תזרוק שגיאה בזמן ריצה
- ב. אם הקובץ dictionary.txt נגיש לקריאה אך ריק, התכנית תזרוק שגיאה בזמן ריצה
- ג. אם הקובץ dictionary.txt נגיש לקריאה ומכיל רק "a", התכנית תדפיס מילה זו וייתכן שתסיים לרוץ ללא שגיאות
- ד. אם הקובץ dictionary.txt נגיש לקריאה ומתחיל ב-"a aaron", התכנית תדפיס כל מילה בקובץ בשורה נפרדת וייתכן שתסיים לרוץ ללא שגיאות (מילים מוגדרות כרצפי תווים המופרדים ביניהם ב-white spaces)
- ה. הבדיקה scanner != null מיותרת, משום שאם הגענו לנק' זו בקוד scanner בהכרח אינו null

Comment [22s]: נימוק: התוכנית מייצרת Scanner חדש בכל איטרציה של הלולאה ולכן הלולאה תדפיס שוב ושוב את המילה הראשונה בקובץ ולא תסיים לעולם.

נימוק:

שאלה 20

נתונה הפונקציה הבאה

```
public static String f(String s1, String s2) {
    s1.replaceAll(s2, "");
    s2.replaceFirst(s1, "b");
    return s2;
}
```

מה יהיה ערך ההחזרה של f("abaabaa", "baa")?

- א. מחרוזת ריקה
- ב. abaabaa
- ג. baa
- ד. bab
- ה. bba

Comment [23s]: נימוק: השירות אינו משנה את הארגומנטים בכלל ומחזיר את השני.

נימוק:

נתון קוד ג'אווה אשר מייצר חלון ממשק משתמש גרפי הכולל 3 תוויות (Label) ו-3 שדות טקסט (Text) ויראה כך מיד לאחר הרצת התוכנית:

בכל אחת מהשאלות הבאות, נתון לכם הקוד שיושלם במקטע הקוד החסר, ועליכם לציין מה יופיע ב-3 תוויות החלון, לאחר שהרצנו את התוכנית מהתחלה והקלדנו את המילה "hello" בכל אחד משדות הטקסט לפי הסדר (מלמעלה למטה).

```
public class GUIclass {

    private static Text[] texts = new Text[3];;
    private static Label[] labels = new Label[3];
    private static int counter = 0;

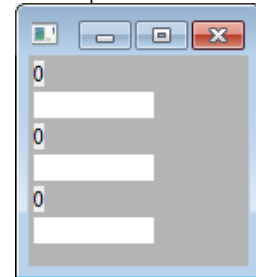
    public static void main(String[] args) {

        Display display = Display.getDefault();
        Shell shell = new Shell(display);
        shell.setLayout(new RowLayout(SWT.VERTICAL));

        for (int i=0; i<3; i++) {
            labels[i] = new Label(shell, SWT.LEFT);
            labels[i].setText("0");
            texts[i]=new Text(shell, SWT.LEFT);
        }

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

Missing code



המשך קוד המחלקה בעמוד הבא...

```

private static void addListeners1(Text text, Label label) {
    KeyAdapter keyAdapter = new KeyAdapter() {
        int counter = 0;
        public void keyPressed(KeyEvent e) {
            if (e.character == '1')
                counter++;
            label.setText(Integer.toString(counter));
        }
    };
    text.addKeyListener(keyAdapter);
}

private static void addListeners2(Text text, Label label) {
    text.addKeyListener(new KeyAdapter() {
        public void keyPressed(KeyEvent e) {
            if (e.character == '1')
                counter++;
            label.setText(Integer.toString(counter));
        }
    });
}

private static void addListeners3(Text text, Label[] labels) {
    KeyAdapter keyAdapter1 = new KeyAdapter() {
        public void keyPressed(KeyEvent e) {
            if (e.character == '1')
                counter++;
        }
    };

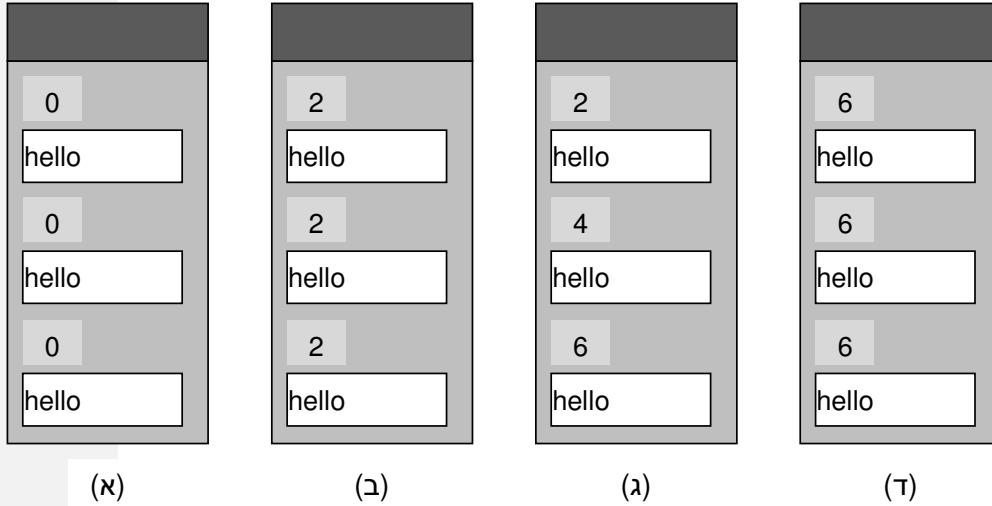
    KeyAdapter keyAdapter2 = new KeyAdapter() {
        public void keyPressed(KeyEvent e) {
            for (int i=0; i<labels.length; i++)
                labels[i].setText(Integer.toString(counter));
        }
    };

    text.addKeyListener(keyAdapter1);
    text.addKeyListener(keyAdapter2);
}
} // end of class

```

שאלה 21

```
for (int i=0; i<3; i++)
    addListeners1(texts[i], labels[i]);
```

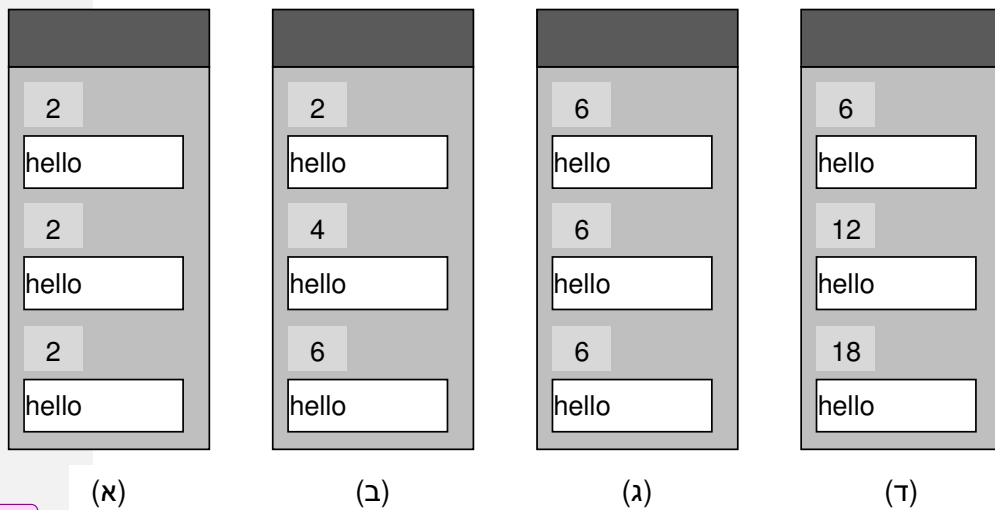


Comment [24s]: התשובה הנכונה היא ב'. עבור כל שדה טקסט מוגדר מאזין שמעדכן את התווית המתאימה לו במספר המופעים של 1 במילה hello שהוקלדה.

נימוק

שאלה 22

```
for (int i=0; i<3; i++)
    for (int j=0; j<3; j++)
        addListeners1(texts[i], labels[j]);
```

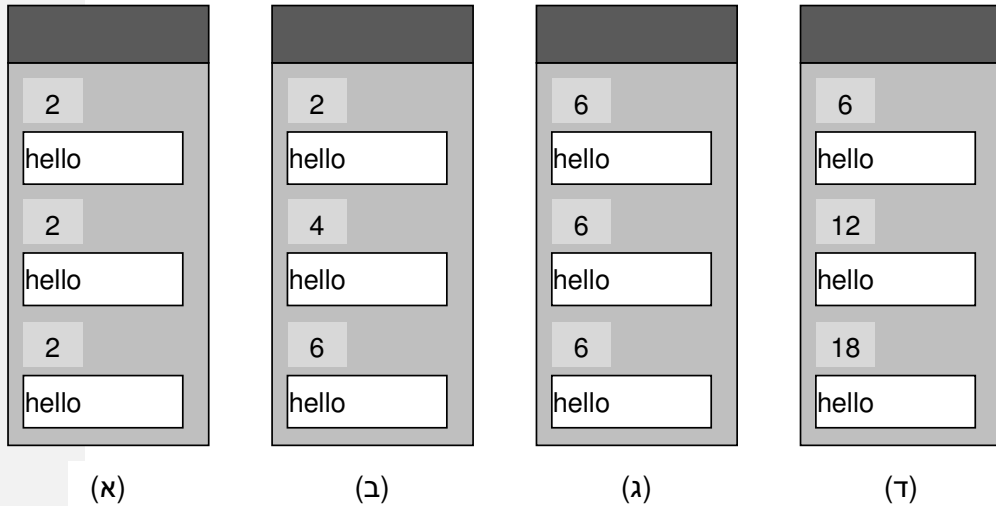


Comment [25s]: התשובה הנכונה היא א'. עבור כל שדה טקסט הוגדרו שלושה מאזינים שיעדכנו את שלושת התוויות במספר המופעים של 1 במילה שהוקלדה.

נימוק

שאלה 23

```
for (int i=0; i<3; i++)
    addListeners2(texts[i], labels[i]);
```

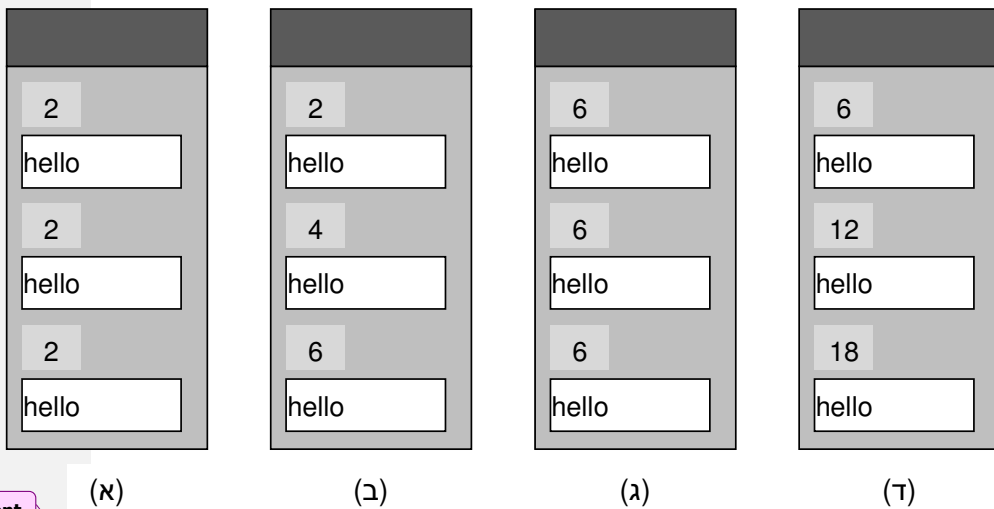


Comment [26s]: התשובה הנכונה ב'. עבור כל שדה טקסט מוגדר מאזין שמעדכן את התוית המתאימה במספר המופיעים של 1 במילה שהוקדה, אבל עם מונה גלובלי.

נימוק _____

שאלה 24

```
for (int i=0; i<3; i++)
    addListeners3(texts[i], labels);
```



Comment [27s]: התשובה הנכונה היא ג'. עבור כל שדה טקסט מוגדר מאזין שמעדכן מונה גלובלי במספר המופיעים של 1, ומאזין שני שמעדכן את כל התויות בערכו של המונה.

נימוק _____

שאלה 25

נתון הקוד הבא

```
public class MyClass {  
  
    public MyClass() {  
        this(2);  
        fld = 3 - fld;  
    }  
  
    public MyClass(int i) {  
        fld = 6 - fld;  
    }  
  
    private int fld = 10;  
  
    public static void main(String[] args) {  
        MyClass myC = new MyClass();  
        myC.fld = 18 - myC.fld;  
        System.out.println(myC.fld);  
    }  
}
```

סמנו את התשובה הטובה ביותר מבין האפשרויות הבאות

- א. התכנית תדפיס 5
- ב. התכנית תדפיס 8
- ג. התכנית תדפיס 10
- ד. התכנית תדפיס 11
- ה. התכנית תדפיס 25

ו. תהיה שגיאת קומפילציה בשורה השניה ב-main (ואולי בשורות נוספות)

נימוק

Comment[28s]: נימוק: השדה מקבל את הערך 10 לפני הבנאים, הבנאי בלי ארגומנטים נקרא וקורא מייד לבנאי עם ארגומנט אחד, ואז רץ גוף הבנאי ללא ארגומנטים. מחסרים את התוצאה מ-18 ומקבלים 11.