

בחינה בתוכנה 1

מסטר א' תשע"ז, מועד א', 14 בפברואר 2017
ליאור וולף, תומר עזרא, לנה דנקין

משך הבחינה שלוש שעות.

יש להניח שהקוד שמופיע במבחן מתאים לגרסה 7 של ג'אווה, אלא אם כן צויין אחרת בשאלה (יש שאלה/שאלות שמתייחסות ספציפית לג'אווה 8).

בבחינה זו מופיע קוד שבחלקו אינו מתקמפל, אינו רץ או שנוגד את הסטנדרטים של java כפי שנלמדו בקורס, וזאת מתוך מטרה לבחון ידע והבנה של נושאים מסויימים. אין לראות בקטעי קוד אלה דוגמא לכתיבה נכונה ב java.

יש לסמן את התשובה הטובה ביותר בתשובון. לא יינתן ניקוד על סימון תשובה בטופס הבחינה או במחברת הבחינה.

ניתן לנמק את התשובות בטופס הנימוקים המצורף בלבד – נימוקים בגוף הבחינה לא יתקבלו. ניקוד ינתן ללא קשר לנימוק אלא במקרה בו במהלך ערעור יטען שתשובה מסויימת נובעת מפרשנות של השאלה שהיא סבירה למרות שלא התכוונו אליה. אנחנו עושים כל מאמץ כדי שלא יהיו מצבים מעין אלו, אבל אם יהיה, על הנימוק להיות תואם לפרשנות זו. המקום המיועד לנימוקים בטופס 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 מחשבוני 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 מכונית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

בהצלחה!

1. שאלה 1:

בחן/י את הקוד הבא ובחרי/י את התשובה הטובה ביותר.

```
public class MyClass {  
    public static void main(String[] args){  
        int[] arr = new int[]{1,1,1};  
        int[] another_arr = arr;  
        func(arr, another_arr);  
        System.out.println(Arrays.toString(arr));  
    }  
  
    public static void func(int[] arr1, int[] arr2){  
        if (arr1 == arr2){  
            arr1[0] = 10;  
        }  
        arr1 = new int[]{3,3,3};  
        if (arr1 != arr2){  
            arr1[2] = 10;  
        }  
    }  
}
```

בהרצת התוכנית יודפס:

- א. [1,1,1]
- ב. [3,3,3]
- ג. [10,1,1]
- ד. [10,3,3]
- ה. [10,1,10]
- ו. [1,1,10]
- ז. [3,3,10]

2. שאלה 2:

```
public void func(A a){  
    B b = (B)a;  
    // the rest of the code  
}
```

התבוננו בטענות הבאות, הנטענות לכל A ו B, בהנחה שידוע כי גם A וגם B אינן מנשקים:
טענה 1: ניתן לקרוא ל func רק עם משתנה מטיפוס סטטי A.
טענה 2: ניתן לקרוא ל func רק עם משתנה מטיפוס דינאמי A.
טענה 3: הפונק' func תתקמפל רק אם B הוא צאצא של A בעץ הירושה.
טענה 4: אם הקוד מתקמפל, לא תיתכן שגיאת זמן ריצה בשלב ה casting מ B ל A.
בחר/י את התשובה הטובה ביותר:

- א. רק טענה 1 נכונה
- ב. רק טענה 2 נכונה
- ג. רק טענה 3 נכונה
- ד. טענות 3+4 נכונות
- ה. טענות 1+3 נכונות
- ו. טענות 2+4 נכונות
- ז. טענות 1+2 נכונות
- ח. אף טענה אינה נכונה

3. שאלה 3:

בחר/י את הטענות הבאות ובחר/י את התשובה הטובה ביותר.
טענה 1: כל הקבועים והשירותים במנשק יהיו בעלי ניראות public.
טענה 2: מחלקה יכולה לממש יותר ממנשק אחד.
טענה 3: מנשק יכול לממש מנשק אחר או לרשת ממחלקה אבסטרקטית, בתנאי שהמחלקה האבסטרקטית אינה מכילה מימושים.
טענה 4: כל מחלקה אשר עושה implements למנשק חייבת לממש את כל השירותים שהוא מגדיר.

בחר/י את התשובה הטובה ביותר:

- א. כל הטענות נכונות
- ב. רק טענות 2+3+4 נכונות
- ג. רק טענות 1+2 נכונות
- ד. רק טענות 1+2+3 נכונות
- ה. רק טענות 1+4 נכונות
- ו. רק טענות 2+3 נכונות
- ז. רק טענות 2+4 נכונות
- ח. רק טענות 1+3+4 נכונות

4. שאלה 4:

בהנתן ממשק A, נרצה להגדיר אוסף MyList גנרי שיכיל רק עצמים מטיפוסים המממשים את הממשק A האוסף יממש את הממשק הגנרי List כמו שהוא, מבלי להוסיף שירותים חדשים. לדוגמא, עבור שתי מחלקות B ו C המממשות את A נרצה שהקוד של Test יתקמפל, ובנוסף, אם נוציא את השורה האחרונה מההערה, הקוד לא אמור להתקמפל.

```
public interface A{
}
public class B implements A{
}
public class C implements A{
}
public class Test{
    public static void main(String[] args){
        MyList<B> listOfB = new MyList<>();
        listOfB.add(new B());
        //listOfB.add(new C()); - this shouldn't compile
    }
}
```

java.util

Interface List<E>

Type Parameters:

E - the type of elements in this list

All Superinterfaces:

Collection<E>, Iterable<E>

add

boolean add(E e)

Appends the specified element to the end of this list (optional operation).

מהי ההגדרה הנכונה של המחלקה MyList?

- א. `MyList<T extends A> implements List<T>`
- ב. `MyList<T> implements List<T extends A>`
- ג. `MyList<T extends A> implements List<A>`
- ד. `MyList<A> implements List<T extends A>`
- ה. `MyList<A> implements List<T>`
- ו. `MyList<T> implements List<A>`
- ז. `MyList implements List<T extends A>`
- ח. במחלקה גנרית לא ניתן לצמצם את הטיפוס הגנרי בתוך הגדרת המחלקה

5. שאלה 5:

בחנ'י את הקוד הבא ובחר'י את התשובה הטובה ביותר:

```
public class CC {  
    public static void main(String[] args){  
        AA a = new BB();  
        a.foo(a.i);  
    }  
}  
  
public class AA{  
    public int i = 5;  
  
    public void foo(int x){  
        System.out.println("AA.foo: " + i + " ; " + x);  
    }  
}  
  
public class BB extends AA{  
    public int i = 8;  
  
    public void foo(int x){  
        System.out.println("BB.foo: " + i + " ; " + x);  
    }  
}
```

בהרצת התוכנית CC יודפס:

- א. AA.foo: 8 ; 5
- ב. AA.foo: 8 ; 8
- ג. AA.foo: 5 ; 5
- ד. AA.foo: 5 ; 8
- ה. BB.foo: 8 ; 5
- ו. BB.foo: 8 ; 8
- ז. BB.foo: 5 ; 5
- ח. BB.foo: 5 ; 8

6. שאלה 6:

להלן מספר טענות על חריגים:

טענה 1: לא חייבים להצהיר בחתימת המתודה על זריקת חריגים שאינם נבדקים (unchecked).

טענה 2: חריגים שאינם נבדקים אינם יכולים להיתפס בבלוק try/catch ויש לטפל בהם בצורה אחרת, למשל לבדוק קלט לפני קריאה למתודה שעלולה לזרוק אותם.

טענה 3: מדיניות ההצהרה על חריגים נועדה למנוע שגיאות זמן ריצה. במידה ונוריד מהקוד הצהרה על חריג, הקוד עלול לעוף בזמן ריצה.

בחר'י את התשובה הטובה ביותר:

- א. אף טענה אינה נכונה
- ב. רק טענה 1 נכונה
- ג. רק טענה 2 נכונה
- ד. רק טענה 3 נכונה
- ה. רק טענות 1+2 נכונות
- ו. רק טענות 1+3 נכונות
- ז. רק טענות 2+3 נכונות
- ח. כל הטענות נכונות.

7. שאלה 7:

לפניכם קוד אשר מיועד לקרוא שורה אחת מקובץ הטקסט input.txt, ומספר טענות הקשורות אליו.
הקוד הנתון משתמש במחלקה BufferedReader אותה ראיתם בתרגול, ועובדת בדומה למחלקה MyBufferedReader אותה מימשתם באחד מתרגילי הבית.
הניחו כי הקובץ input.txt קיים בנתיב ההרצה ומכיל יותר משורת טקסט אחת.

```
BufferedReader bR = new BufferedReader(new FileReader("input.txt"));  
System.out.println(bR.readLine());
```

- טענה 1: השימוש ב BufferedReader מבטיח לנו כי עבור כל קריאה ל readLine נקרא מהקובץ input.txt פעם אחת לכל היותר.
- טענה 2: שימוש ב BufferedReader מבטיח לנו שלא יקרו שגיאות IOException בזמן הקריאה מקובץ.
- טענה 3: אם נרצה לחסוך בגישות לקובץ ממנו אנו קוראים, נרצה שגודל ה buffer של ה BufferedReader יהיה כמה שיותר גדול.
בחר/י את התשובה הטובה ביותר:
- א. אף טענה אינה נכונה.
 - ב. כל הטענות נכונות.
 - ג. רק טענה 1 נכונה.
 - ד. רק טענה 2 נכונה.
 - ה. רק טענה 3 נכונה.
 - ו. רק טענות 1+2 נכונות.
 - ז. רק טענות 1+3 נכונות.
 - ח. רק טענה 2+3 נכונות.

8. שאלה 8:

לפניכם קוד המחלקה Gen_Class, מימוש המתודה main ובנוסף, שלושה מימושים אפשריים של הפונקציה func המופיעים בהערה. הקוד אינו מתקמפל כיוון ש main קוראת ל func עבורה אין מימוש.

```
public class Gen_Box<T> {  
    /*  
    public void func(List<T> lst){ //version 1  
    }  
  
    public void func(List<?> lst){ //version 2  
    }  
  
    public void func(List lst){ //version 3  
    }*/  
  
    public static void main(String[] args){  
        Gen_Box<String> box = new Gen_Box<>();  
        List<String> myList = new ArrayList<>();  
        box.func(myList);  
    }  
}
```

מבין שלושת המימושים האפשריים ל func (מספור כל פונקציה מופיעה בהערה), הוספת איזה מימוש תאפשר לקוד של main להתקמפל? הניחו כי אנו מתייחסים להוספת כל אחד משלושת המימושים בנפרד.

- א. מימוש מספר 1 בלבד.
- ב. מימוש מספר 2 בלבד.
- ג. מימוש מספר 3 בלבד.
- ד. מימוש מספר 1 או מימוש מספר 2 בלבד.
- ה. מימוש מספר 1 או מימוש מספר 2 בלבד.
- ו. מימוש מספר 2 או מימוש מספר 3 בלבד.
- ז. הוספת כל אחד מבין המימושים תגרום לקוד להתקמפל.
- ח. הקוד לא יתקמפל עבור הוספת אף אחד מבין שלושת המימושים.

9. שאלה 9:

לפניך מספר טענות כלליות על ארכיטקטורת Java. בחר/י את הטענה שאינה נכונה.

- א. המהדר (compiler) קורא קבצי Java ומייצרי קבצי byte code.
- ב. ה JVM מכיל את המהדר (compiler) והמפרש (interpreter) של Java.
- ג. ניתן לקמפל ב Java 8 קוד שנכתב בסינטקס של Java 7.
- ד. אם על מחשב לינוקס מותקן מפרש (interpreter) של java 7, נוכל להריץ עליו תוכנית שהודרה (compiled) בגירסת java 7 על מחשב windows.

10. שאלה 10:

נתונה המחלקות A ו B. נרצה לבדוק אם ניתן יהיה להוריד חלק מהבנאים הממומשים ב A כך ש B תתקמפל מבלי לשנות את המימוש שלה. הבנאים של A ממוספרים בהערות המופיעות ליד ההצהרה על כל בנאי. שימו לב שלשלושת הבנאים של A מימוש ריק והם אינם תלויים אחד בשני.

```
public class A{
    public A(){ //constructor 1
    }

    public A(int i){ //constructor 2
    }

    public A(int i, int j){ //constructor 3
    }
}

public class B extends A{
    public B(int i, int j, int k){
        this(i, j);
    }

    public B(int i, int j){
        super(i);
    }

    public B(){
    }
}
```

בחר/י את התשובה הטובה ביותר:

- א. הקוד של B אינו מתקמפל ועבור כל בנאי שנסיר מ A הוא ימשיך לא להתקמפל.
- ב. הקוד של B אינו מתקמפל, אבל ניתן להסיר בנאים מ A כך ש B תתקמפל.
- ג. הקוד של B מתקמפל, על מנת שימשיך להתקמפל, ניתן להסיר רק את בנאי 1.
- ד. הקוד של B מתקמפל, על מנת שימשיך להתקמפל, ניתן להסיר רק את בנאי 2.
- ה. הקוד של B מתקמפל, על מנת שימשיך להתקמפל, ניתן להסיר רק את בנאי 3.
- ו. הקוד של B מתקמפל, אם נסיר את בנאים 1+2, הקוד ימשיך להתקמפל.
- ז. הקוד של B מתקמפל, אם נסיר את בנאים 1+3, הקוד ימשיך להתקמפל.
- ח. הקוד של B מתקמפל, אם נסיר את בנאים 2+3, הקוד ימשיך להתקמפל.

11. שאלה 11:

```
public class ExpTest {
    public static void main(String[] args) throws Exception{
        List<String> strings = Arrays.asList("A", "", null, "C");
        for (String str: strings){
            try{
                checkEmpty(str);
                System.out.print(str + " ");
            }
            catch (MyLengthException ex){
                System.out.print("ERROR! ");
            }
            finally{
                System.out.print("# ");
            }
        }
    }

    public static void checkEmpty(String str) throws MyLengthException{
        if (str.length() == 0){
            throw new MyLengthException();
        }
    }

    static class MyLengthException extends Exception{
    }
}
```

- א. התוכנית תדפיס: # ERROR! A ולאחר מכן תעוף על שגיאת זמן ריצה.
- ב. התוכנית תדפיס: # ERROR! A ותסתיים באופן תקין ללא שגיאות.
- ג. התוכנית תדפיס: # # ERROR! A ולאחר מכן תעוף על שגיאת זמן ריצה.
- ד. התוכנית תדפיס: # # ERROR! A ותסתיים באופן תקין ללא שגיאות.
- ה. התוכנית תדפיס: # ERROR! # A ולאחר מכן תעוף על שגיאת זמן ריצה.
- ו. התוכנית תדפיס: # ERROR! # A ותסתיים באופן תקין ללא שגיאות.
- ז. התוכנית תדפיס: # # ERROR! # A ולאחר מכן תעוף על שגיאת זמן ריצה.
- ח. התוכנית תדפיס: # # ERROR! # A ותסתיים באופן תקין ללא שגיאות.
- ט. התוכנית תדפיס: # C # ERROR! # ERROR! # A ותסתיים באופן תקין ללא שגיאות.

12. שאלה 12:

לפניך מספר טענות על המילה השמורה final. בחר/י את הטענה הנכונה:

- א. שדות שהם final חייבים להיות מוגדרים גם כ static (אך יתכנו שדות סטטיים שאינם final)
- ב. מתודה שהיא final היא מתודה שהמחלקה היורשת לא יכולה לקרוא לה.
- ג. מתודה שהיא final היא מתודה שלא דורסת אף מתודה במחלקת האב.
- ד. לא ניתן להגדיר מחלקה כ final.
- ה. גם שדות מחלקה וגם משתנים של פונקציות יכולים להיות מוגדרים כ final.

```
public void testFinally() {
    System.out.println(setOne().toString());
}

protected StringBuilder setOne() {
    StringBuilder builder = new StringBuilder();
    try {
        builder.append("Cool");
        return builder.append("Return");
    }
    finally {
        builder.append("+1");
    }
}
```

java.lang

Class StringBuilder

java.lang.Object

java.lang.StringBuilder

All Implemented Interfaces:

Serializable, Appendable, CharSequence

StringBuilder

```
public StringBuilder()
```

Constructs a string builder with no characters in it

```
public StringBuilder append(String str)
```

Appends the specified string to this character sequence.

שימו לב – פעולת ה `append` משנה את הערך של המחרוזת הנבנית ומחזירה את העצם שעליו הפעלנו את הפעולה. כך ניתן למשל לשרשר שוב ושוב:

```
StringBuilder builder = new StringBuilder();
builder.append("a").append("b").append("c");
System.out.println(builder); //prints "abc"
```

מה יקרה בהרצת המתודה `testFinally`:

- א. יודפס `CoolReturn+1`
- ב. שגיאת קומפילציה
- ג. יודפס `CoolReturn`
- ד. שגיאת זמן ריצה
- ה. יודפס `Cool+1`
- ו. יודפס `Cool`
- ז. יודפס `+1`
- ח. יודפס `Cool+1Return`
- ט. תודפס מחרוזת ריקה

```
public class SizeSet {  
  
    public static void main(String[] args) {  
        EnumSet largeSize = EnumSet.of(Size.XL,Size.XXL,Size.XXXL);  
        for(Iterator it = largeSize.iterator() ; it.hasNext() ; ){  
            Size size = (Size)it.next();  
            System.out.println(size);  
        }  
    }  
}  
  
enum Size {  
    S, M, L, XL, XXL, XXXL;  
}
```

java.util

Class EnumSet<E extends Enum<E>>

java.lang.Object

java.util.AbstractCollection<E>

java.util.AbstractSet<E>

java.util.EnumSet<E>

All Implemented Interfaces:

Serializable, Cloneable, Iterable<E>, Collection<E>, Set<E>

מה יקרה עם ההרצה:

- א. שגיאת קומפילציה בשורה של for
- ב. שגיאת זמן ריצה בשורה של it.next()
- ג. יודפסו שלוש מידות XL XXL XXXL בשורות נפרדות
- ד. יודפסו שש מידות בשורות נפרדות

15. שאלה 15:

```
public abstract class My {
    public void myMethod() {
        System.out.print("Abstract");
    }
}

public class Poly {
    public static void main(String a[]) {
        My m = new My() {};
        m.myMethod();
    }
}
```

מבין הטענות הבאות, אילו טענות נכונות? בחר/י את התשובה הטובה ביותר:

- A. מתקיים `m instanceof My`.
- B. הטיפוס הדינאמי של `m` הוא `My`.
- C. הטיפוס הדינאמי של `m` יורש מ-`My`.
- D. הטיפוס הסטטי של `m` הוא `My`.

- א. A,C,D
- ב. A,B,C,D
- ג. D
- ד. A,C
- ה. A,B

16. שאלה 16:

```
public class B extends A {
    public static void main(String[] argv){
        B b = new B();
        A a = new A ();
        System.out.print((b instanceof A) + ",");
        System.out.print((a instanceof B) + ",");
        System.out.print(b.getClass() == a.getClass());
    }
}
```

מה יודפס בשורות נפרדות משמאל לימין בהרצת התוכנית? הניחו כי קיים מימוש כלשהו למחלקה A והקוד מתקמפל.

- א. true, false, false
- ב. true, true, false
- ג. false, true, false
- ד. false, false, true
- ה. false, false, false
- ו. true, true, true

שאלה זו מתייחסת לגירסת Java 8.

```
public class Java8Tester {  
  
    final static String salutation = "Hello! ";  
  
    public static void main(String args[]) {  
        g(message -> System.out.println(salutation + message));  
    }  
  
    public static void g(GreetingService greetService1) {  
        greetService1.sayMessage("Mahesh");  
    }  
  
    interface GreetingService {  
        ****  
    }  
  
}
```

עבור תוכנית תקינה שעוברת קומפילציה מה צריך להיות במקום ****?

- א. void sayMessage(String a);
- ב. String sayMessage(String a);
- ג. void sayMessage();
- ד. String sayMessage();
- ה. void sayMessage(String x, String y);

התבוננו בקוד הבא:

```
public class D{
    public static void main(String[] args){
        B b = new C();
        b.f();
        ((B)new C()).f();
    }
}
public class A {
    static void f() {
        System.out.print("A");
    }
}
public class B extends A {
}
public class C extends B {
    static void f(){
        System.out.print("C");
    }
}
```

מה יודפס בהרצת התוכנית D?:

- א. AC
- ב. AA
- ג. CA
- ד. CC
- ה. הקוד לא יתקמפל
- ו. תתרחש שגיאה בזמן ריצה

```
public abstract class HttpServlet {  
  
    private static java.util.Map? ii = new java.util.TreeMap<>();  
  
    static HttpServlet getInstance(String class_name) {  
        HttpServlet servlet = ii.get(class_name);  
        if (servlet != null)  
            return servlet;  
  
        try {  
            java.lang.reflect.Constructor constructor =  
Class.forName(class_name).getConstructor(new Class[] {});  
            servlet = (HttpServlet) constructor.newInstance(new  
Object[] {});  
        }  
        catch (Exception e) {  
            return null;  
        }  
        ii.put(class_name, servlet);  
        return servlet;  
    }  
  
    protected HttpServlet() {  
    }  
  
    protected abstract void doGet(HttpServletRequest req,  
HttpServletResponse resp);  
}
```

מה צריך להיות a? בשורה השנייה על מנת שהקוד יתקמפל?

- א. <String,HttpServlet>
- ב. <String,String>
- ג. <HttpServlet,String>
- ד. < HttpServlet>
- ה. <String>

```
public class Q{
    static class A{}
    static class B extends A{}
    static class C extends B{}

    static void f(A a, B b) {} /**
    static void f(B a, A b) {} /**
    static void f(B a, B b) {} /**

    public static void main(String []args){
        f(new C(),new C());
    }
}
```

לפניך מספר טענות המתייחסות לקוד הנתון:

טענה 1: הקוד אינו מתקמפל

טענה 2: אם נמחק את השורה המסומנת ב **/**** הקוד לא יתקמפל

טענה 3: בכל מצב בו נשאיר בקוד יותר משורה אחת מבין השורות המסומנות

ב **/**,/**,/**** הקוד לא יתקמפל

אילו מבין הטענות הן נכונות? בחר/י את התשובה הטובה ביותר:

- א. אף טענה אינה נכונה
- ב. רק טענה 1 נכונה
- ג. רק טענה 2 נכונה
- ד. רק טענה 3 נכונה
- ה. רק טענות 1,2 נכונות
- ו. רק טענות 1,3 נכונות
- ז. רק טענות 2,3 נכונות
- ח. כל הטענות נכונות

21. שאלה 21:

מה יודפס בהרצת התוכנית הבאה?

```
public class Q{
    public static class MyException extends Exception {}
    public static class MyException2 extends MyException {}

    static public class A {
        public void f() throws MyException2 {
            throw new MyException2();
        }
    }

    public static void main(String []args){ /*
        A a = new A();
        try {
            a.f();
        }
        catch (MyException2 e){
            System.out.print("2");
        }
        catch (MyException e){
            System.out.print("1");
        }
    }
}
```

- א. 2
- ב. 1
- ג. 21
- ד. ריצה תקינה שלא תדפיס כלום
- ה. התוכנית תעוף בזמן ריצה

22. שאלה 22:

מה יהיה הפלט של התוכנית הבאה?

```
public class Q{
    static int i;
    public static void main(String []args){
        int s=0;
        for (int i=0; i++<7; i++){
            s+=i++;
        }
        System.out.println("i="+i+" s=" +s);
    }
}
```

- א. i=0 s=5
- ב. i=0 s=7
- ג. i=0 s=12
- ד. i=0 s=15
- ה. i=7 s=5
- ו. i=7 s=7
- ז. i=7 s=12
- ח. i=7 s=15

23. שאלה 23:

לפניכם מימוש שתי מחלקות, A ו B.

```
public class A{
    private void f1(){}
    private void f2(){}
    protected void f3(){}
    protected void f4(){}
}

public class B extends A{
    private void f1(){}
    protected void f2(){}
    void f3(){}
    public void f4(){}
}
```

בהגדרת אילו מבין המתודות במחלקה B תהיה שגיאת קומפילציה?

- א. f1,f2
- ב. f2,f3,f4
- ג. f3
- ד. f1,f2,f3
- ה. f2,f4
- ו. f3,f4

24. שאלה 24:

```
public class HelloWorld {
    private int x1;
    public static int x2;
    static class A{
        private int x3;
        static private int x4;

        class B {
            void f() {
                System.out.println(?x?);
            }
        }
    }
}
```

אילו ערכים מבין הערכים {x1, x2, x3, x4} ניתן להציב במקום ?x? על מנת שהקוד יתקמפל?
בחר/י את התשובה הטובה ביותר:

- א. x1,x2,x3,x4
- ב. x2,x4 בלבד
- ג. x1,x3 בלבד
- ד. x2,x3,x4 בלבד
- ה. x4 בלבד
- ו. x3,x4 בלבד

```
public class MyGui{
    public static void main(String[] args){
        MyGui myGui = new MyGui();
        myGui.runApplication();
    }
    private void runApplication() {
        Display display = Display.getDefault();
        Shell shell = new Shell(display);
        shell.setLayout(new FillLayout(SWT.VERTICAL));
        Button ok = new Button(shell, SWT.PUSH);
        ok.setText("0");
        ok.addSelectionListener(new ButtonHandler());
        ok.addSelectionListener(new ButtonHandler());
        ok.addSelectionListener(new ButtonHandler());
        shell.pack();
        shell.open();
        while (!shell.isDisposed()) {
            if (!display.readAndDispatch())
                display.sleep();
        }
        display.dispose();
    }
    public class ButtonHandler implements SelectionListener {
        public void widgetSelected(SelectionEvent e) {
            if (e.getSource() instanceof Button) {
                Button b = (Button) e.getSource();
                b.setText(b.getText()+1);
            }
        }
        public void widgetDefaultSelected(SelectionEvent e){}
    }
}
```

איזה טקסט יופיע על הכפתור לאחר לחיצה אחת עליו בהרצת התוכנית?

- א. 3
- ב. 0
- ג. 111
- ד. 0111
- ה. 01

org.eclipse.swt.widgets

Class Button

java.lang.Object

org.eclipse.swt.widgets.Widget

org.eclipse.swt.widgets.Control

org.eclipse.swt.widgets.Button

addSelectionListener

```
public void addSelectionListener(SelectionListener listener)
```

Adds the listener to the collection of listeners who will be notified when the control is selected by the user, by sending it one of the messages defined in the SelectionListener interface.