

בחינה בתוכנה 1

גרסה 1

סמסטר ב' תשע"ה, מועד א', 13 יולי 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.

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 מכונית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

בהצלחה!

שאלה 1 (4 נק')

באיזה מחלקת אוסף נשתמש כדי לשמור עצמים כאשר הם ממויינים על פי סדר מסויים?

- א. HashSet
- ב. LinkedHashSet
- ג. Array
- ד. TreeSet

נימוק:

שאלה 2 (4 נק')

איזה מהבאים אינו יתרון של מחלקה פנימית אנונימית?

- מספרת קריאות הקוד ע"י הצמדת קוד המחלקה הפנימית למקום השימוש
- מאפשרת גישה לשדות פרטיים במחלקה העוטפת וחוסכת כתיבת קוד
- מספרת את זמן הריצה של התוכנית היות ולא נדרשים שלבי האתחול של מחלקה רגילה
- מאפשרת לצמצם את מספר קבצי הקוד בפרוייקט ומקילה על ההתמצאות בתיקיית הפרוייקט

נימוק:

שאלה 3 (4 נק')

נתון הקוד הבא:

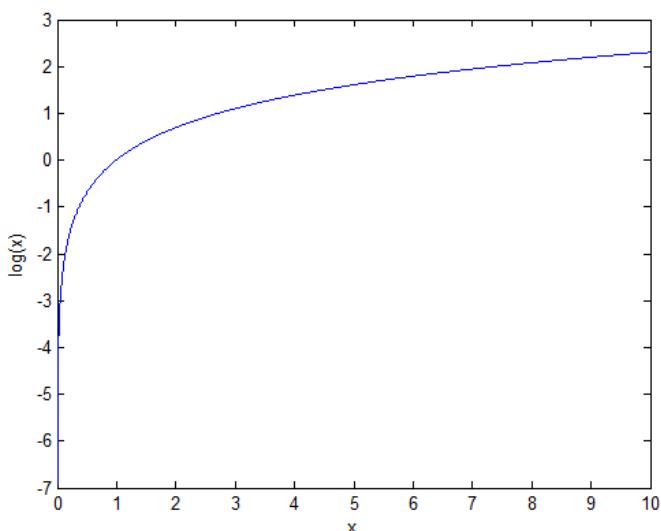
```
try {
    Scanner scanner = new Scanner(System.in);
    while (scanner.hasNextLine()) {
        System.out.println(scanner.nextLine());
    }
} catch (Exception e) {}
```

כאשר נריץ את הקוד,

- הקוד יקרא את כל הקלט מן המשתמש ויסתיים כאשר הוא יקיש Enter
- הקוד יקרא את כל הקלט מקובץ ויסתיים כאשר יגיע לסוף הקובץ
- הקוד לא יתקמפל
- אף אחת מהתשובות האחרות לא נכונה

נימוק:

השאלות הבאות מתייחסות לקוד הבא שהשירותים שמוגדרים בו אמורים לחשב קירוב ללוגריתם של מספר אי שלילי. בשאלות יש לבחור תשובות שמשמרות עד כמה שאפשר את הכוונה של מפתח הקוד, כלומר לשמר את היכולת לחשב קירובים ללוגריתמים. למשל, תשובה שמבוססת על שינוי תנאי הקדם כך שלעולם לא יתקיים איננה תשובה טובה. הגרף של פונקציית הלוגריתם מופיע משמאל.



```

public class LogAsLong {
    /** @pre x >= 1
     * @post $ret <= log(x) <= $ret + 1
     */
    public static long log(int x) {...}
    ...
}

public class LogAsInt extends LogAsLong {
    /** @pre x >= 1
     * @post $ret <= log(x) <= $ret + 1
     */
    public static int logOfNonNegative(int x) {...}
    ...
}

```

שאלה 4 (4 נק')

השאלה הזאת תקפה באותה מידה לשתי המחלקות.

- א. החוזה של \log תקין כמו שהוא (כלומר מאפשר מימוש של השירות כך שיקיים את החוזה)
- ב. החוזה יהיה תקין אם נוסיף לו עוד תנאי שמגביל את הארגומנט x
- ג. החוזה יהיה תקין אם נוסיף לו תנאי, אבל אין צורך להגביל את x בתנאי הזה

נימוק:

שאלה 5 (4 נק')

תנאי האחר בשירות של `LogAsInt`

- א. מחזק את תנאי האחר של השירות ב-`LogAsLong`
- ב. מחליש את תנאי האחר של השירות ב-`LogAsLong` ומכיון ש-`LogAsInt` מרחיבה את `LogAsLong`, החלשה כזאת היא לא תקינה
- ג. מחליש את תנאי האחר של השירות ב-`LogAsLong`
- ד. אינו מחליש ואינו מחזק את תנאי האחר של השירות ב-`LogAsLong`

נימוק:

שאלה 6 (4 נק')

השאלה הבאה מתייחסת לקוד הבא.

```
Stream<String> stream = ...;
stream.filter( s -> s.contains("X") )
    .sorted()
    .forEach( System.out::println );
```

בהנחה שהקוד שמאתחל את ה-Stream<String> תקין (עובר קומפילציה ולא זורק חריג בזמן ריצה). שאר הקוד

- א. לא יתקמפל בגלל בעיה בשורה שקוראת ל-filter
- ב. לא יתקמפל בגלל בעיה בשורה שקוראת ל-foreach
- ג. יתקמפל ואם ירוץ בלי לזרוק חריג, בכל שורה בפלט יופיע התו X
- ד. יתקמפל אבל גם אם ירוץ בלי לזרוק חריג, בפלט עלולה להופיע שורה בלי התו X

נימוק:

שאלה 7 (4 נק')

השאלה הבאה מתייחסת לקוד הבא.

```
public class Test {
    private class Ex extends RuntimeException {}

    public void f() {
        throw new Ex();
    }
}
```

- א. הקוד לא יתקמפל מכיון שהשירות זורק חריג ממחלקה פרטית; אפשר לתקן את הבעיה הזאת על ידי שינוי private ל-public
- ב. הקוד לא יתקמפל מכיון שהשירות לא תופס את החריג ולא מכריז עליו; אפשר לתקן את הבעיה הזאת על ידי הוספת פסוק throws להגדרה של f
- ג. גם א' וגם ב' נכונות
- ד. הקוד יתקמפל כפי שהוא

נימוק:

שאלה 8 (4 נק')

השאלה הבאה מתייחסת לקוד הבא. הקוד שמאתחל את x יוצר HashSet של 3 קבוצות שבראשונה 2 מחרוזות (שהן x, y), בשניה מחרוזת אחת (z), והשלישית ריקה.

```
Set< Set<String> > sss = ...;
int n=0;
for (Set<String> s: sss) {
    Iterator<String> i = s.iterator();
    while (i.hasNext())
        System.out.printf("set %d string %s\n",n,i.next());
    n++;
}
```

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

א. לא יתקמפל

ב. יתקמפל וידפיס תמיד את שלוש השורות

```
set 0 string x
set 0 string y
set 0 string z
```

ג. יתקמפל וידפיס תמיד את שלוש השורות

```
set 0 string x
set 0 string y
set 1 string z
```

ד. אף תשובה אחרת לא נכונה

נימוק

שאלה 9 (4 נק'):

נתון הממשק הבא

```
public interface I {
    public void foo(String s);
}
```

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

```
int foo(Object o); /* A */
void foo(Object o2); /* 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

נימוק:

שאלה 10 (4 נק')

השאלה הבאה מתייחסת לקוד הבא

```
public class A extends B {
    public A(int m, int n) {
        super(m,n);
        System.out.println(m-n);
    }
}

public class B extends A {
    public B(int a, int b) {
        super(a,b);
        System.out.println(a*b);
    }
}

public static void main(String[] args) {
    A a1 = new A(3,2);
    A a2 = new B(5,4);
}
}
```

הרצה של main

- ז. תדפיס את המספרים 1 ו-20
- ח. תדפיס את המספרים 12 ו-20
- ט. תדפיס את המספרים 12 ו-20
- י. תדפיס את המספרים 1 ו-1
- יא. התוכנית תגרום לשגיאת קומפילציה
- יב. התוכנית תגרום לשגיאת זמן ריצה

נימוק:

שאלה 11 (4 נק')

עליך להמיר קוד משפת ++C שבה יש מנגנון של ירושה מרובה לשפת ג'אווה. הקוד שעליך להמיר כולל שלוש מחלקות ששתיים מהן אינן יורשות מאף מחלקה אבל השלישית יורשת משתייהן. המרה סבירה לג'אווה תתבסס על

- א. ירושה והכלה
- ב. טיפוסים מוכללים (generics), כמו (HashSet<String>)
- ג. שיגור דינמי
- ד. חריגים

נימוק:

שתי השאלות הבאות מתייחסות לקוד הבא.

```
public class A {
    public Integer findMaxFromIndex(ArrayList<Integer> list,
                                   int startingIndex)
        throws IllegalArgumentException {...}
}
public class B extends A {
    protected Integer findMaxFromIndex(ArrayList<Integer> list, // 1
                                       int startingIndex)
        throws IllegalArgumentException {...}

    public Integer findMaxFromIndex(ArrayList<Integer> myList, // 2
                                    int startingIndex) {...}
}
public class C extends A {
    public Integer findMaxFromIndex(LinkedList<Integer> list, // 3
                                   int startingIndex)
        throws Exception {...}

    public float findMaxFromIndex(List<float> myList, // 4
                                  int startingIndex)
        throws IllegalArgumentException {...}
}
```

שאלה 12 (4 נק')

איזה מהשירותים הממוספרים אפשר לכלול במחלקה B מבלי לגרום לשגיאת קומפילציה?

- א. 1
- ב. 2
- ג. 1 וגם 2 (שני השירותים יכולים להיות מוגדרים ביחד)
- ד. 1 או 2 אבל לא את שניהם ביחד
- ה. אף אחד מהם

נימוק:

שאלה 13 (4 נק')

איזה מהשירותים הממוספרים אפשר לכלול במחלקה C מבלי לגרום לשגיאת קומפילציה?

- א. 3
- ב. 4
- ג. 3 וגם 4 (שני השירותים יכולים להיות מוגדרים ביחד)
- ד. 3 או 4 אבל לא את שניהם ביחד
- ה. אף אחד מהם

נימוק:

5 השאלות הבאות מתייחסות לקוד הבא.

```
public class A {
    private void print() { System.out.print(foo() + " "); }
    public String foo() { return "AAA"; }

    public static void main(String[] args) {
        A[] arr = { new A(), new B() };
        for (int i = 0; i < 2; i++) {
            /***/
        }
    }
}

public class B extends A {
    private void print() { System.out.println("%" + foo() + " "); }
    public String foo() { return "BBB"; }
    public void bar() { print(); }
}
```

שאלה 14 (4 נק')

מה יקרה כשנריץ את main (אם התוכנית תתקמפל בכלל) אם נחליף את גוף הלולאה (מסומן ב-/***/ בקוד)

```
A a = arr[i];
a.print();
```

- א. יודפס AAA AAA
- ב. יודפס AAA BBB
- ג. יודפס AAA %BBB
- ד. יודפס %BBB %BBB
- ה. התוכנית לא תעבור קומפילציה
- ו. התוכנית תזרוק חריג בזמן ריצה

נימוק:

שאלה 15 (4 נק')

מה יקרה כשנריץ את main (אם התוכנית תתקמפל בכלל) אם נחליף את גוף הלולאה (מסומן ב-/**/) בקוד

```
arr[i + 1] = arr[i];  
arr[i + 1].print();
```

- א. יודפס AAA AAA
- ב. יודפס AAA BBB
- ג. יודפס AAA %BBB
- ד. יודפס %BBB %BBB
- ה. התוכנית לא תעבור קומפילציה
- ו. התוכנית תזרוק חריג בזמן ריצה

נימוק:

שאלה 16 (4 נק')

מה יקרה כשנריץ את main (אם התוכנית תתקמפל בכלל) אם נחליף את גוף הלולאה (מסומן ב-/**/) בקוד

```
if (i == 0) { print(); }
```

- א. יודפס AAA
- ב. יודפס %AAA
- ג. יודפס BBB
- ד. יודפס %BBB
- ה. התוכנית לא תעבור קומפילציה
- ו. התוכנית תזרוק חריג בזמן ריצה

נימוק:

שאלה 17 (4 נק')

מה יקרה כשנריץ את main (אם התוכנית תתקמפל בכלל) אם נחליף את גוף הלולאה (מסומן ב-/**/) בקוד

```
if (arr[i] instanceof A) {  
    System.out.print(arr[i].foo() + " ");  
}
```

- א. יודפס AAA
- ב. יודפס AAA AAA
- ג. יודפס AAA BBB
- ד. יודפס BBB
- ה. התוכנית לא תעבור קומפילציה
- ו. התוכנית תזרוק חריג בזמן ריצה

נימוק:

שאלה 18 (4 נק')

מה יקרה כשנריץ את main (אם התוכנית תתקמפל בכלל) אם נחליף את גוף הלולאה (מסומן ב-/**/) בקוד

```
B b = new B() {  
    public void bar() {  
        System.out.print("#" + arr[i].foo() + " ");  
    }  
};  
b.bar();
```

- א. יודפס #AAA #AAA
- ב. יודפס #AAA #BBB
- ג. יודפס %AAA %BBB
- ד. יודפס %BBB %BBB
- ה. התוכנית לא תעבור קומפילציה
- ו. התוכנית תזרוק חריג בזמן ריצה

נימוק:

שתי השאלות הבאות מתייחסות לקוד הבא.

```
public abstract class Filter<T> {
    protected abstract boolean cond(T t);

    public void filter(Collection<T> col) {
        for (Iterator<T> iter = col.iterator(); iter.hasNext();) {
            T t = iter.next();
            if (!cond(t)) iter.remove();
        }
    }
}
```

שאלה 19 (4 נק')

נתבונן על הקוד הנתון למעלה של `Filter<T>`. איזו מבין המחלקות הבאות יכולה לרשת מ-`Filter<T>` ולשמש אותנו כפילטר של מחרוזות? (קוד המחלקה אינו נתון)

- א. `public class StringFilter extends Filter<T> {...}`
- ב. `public class StringFilter<String> extends Filter<T> {...}`
- ג. `public class StringFilter<String> extends Filter<String> {...}`
- ד. `public class StringFilter extends Filter<String> {...}`

נימוק:

שאלה 20 (4 נק')

```
public class SubFilter<T> extends Filter<T> {
    protected boolean cond(T t) {
        return (t != null);
    }
}
```

נתבונן על הקוד הנתון של `SubFilter<T>` היורשת מ-`Filter<T>`. ניצור מופע שלה כך:

```
SubFilter<List<String>> sub = new SubFilter<>();
```

מה מהבאים אינו נכון לגבי `sub`?

- א. ניתן להשתמש בו כדי להסיר null מרשימות של מחרוזות
- ב. ניתן להשתמש בו כדי להסיר null מאוסף של רשימות של מחרוזות
- ג. המתודה filter של sub תשנה רק אוספים המכילים null
- ד. המתודה filter של sub לא תשנה אוספים ריקים

נימוק:

מקבץ שאלות GUI

השאלות הבאות מתייחסות לקוד הבא.

```
public class GUIclass {
    private final static int N = 3;
    private static Integer[] nums = new Integer[]{0,0,0};

    private static Text[] texts = new Text[N];
    private static Button[] buttons = new Button[N];

    public static void main(String[] args) {

        Display display = Display.getDefault();
        Shell shell = new Shell(display);
        // N columns, equal width
        shell.setLayout(new GridLayout(N, true));
        for (int i=0; i<N; i++) {
            texts[i]=new Text(shell, SWT.LEFT);
            texts[i].setText(Integer.toString(i));
        }

        for (int i=0; i<N; i++) {
            buttons[i]=new Button(shell, SWT.PUSH);
            buttons[i].setText("Button" + Integer.toString(i));
        }

        /** MISSING CODE **/

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

    private static void addListeners1() {
        SelectionListener sl = new SelectionAdapter() {
            int m=0;
            public void widgetSelected(SelectionEvent e) {
                for (int i=0; i<N; i++)
                    texts[i].setText(Integer.toString(nums[i]+m));
                m++;
            }
        };
        buttons[0].addSelectionListener(sl);
    }
}
```

```

private static void addListeners2() {
    for (int i=0; i<N; i++) {
        final int j= i;
        buttons[i].addSelectionListener(new SelectionAdapter() {
            public void widgetSelected(SelectionEvent e) {
                int oldNum = Integer.parseInt(texts[j].getText());
                int newNum = oldNum * j;
                texts[j].setText(Integer.toString(newNum));
            }
        });
    }
}

private static void addListeners3() {
    buttons[0].addSelectionListener(new InnerHandler());
    buttons[0].addSelectionListener(new InnerHandler());
    buttons[0].addSelectionListener(new InnerHandler());
    buttons[1].addSelectionListener(new InnerHandler());
    buttons[1].addSelectionListener(new InnerHandler());
    buttons[2].addSelectionListener(new InnerHandler());
}

private static class InnerHandler extends SelectionAdapter {
    public void widgetSelected(SelectionEvent e) {
        for (int i=0; i<N; i++) {
            nums[i]++;
            texts[i].setText(Integer.toString(nums[i]));
        }
    }
}
} // end of class

```

בכל השאלות הבאות יש לסמן את התשובה שמתאימה למצב מסך מסויים. המצבים וההתאמה שלהם לתשובות מופיעים לאחר השאלות.

שאלה 21 (4 נק')

כיצד יראה החלון לאחר הרצת התוכנית ולחיצה על שלושת הכפתורים משמאל לימין אם הקוד החסר הוא

addListeners1()

שאלה 22 (4 נק')

כיצד יראה החלון לאחר הרצת התוכנית ולחיצה על הכפתור השמאלי שלוש פעמים אם הקוד החסר הוא

addListeners1()

שאלה 23 (4 נק')

כיצד יראה החלון לאחר הרצת התוכנית ולחיצה על שלושת הכפתורים משמאל לימין אם הקוד החסר הוא

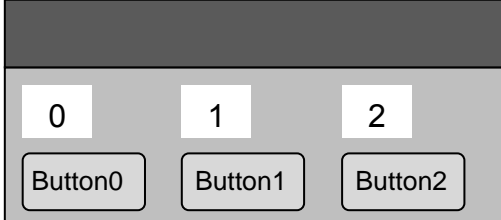
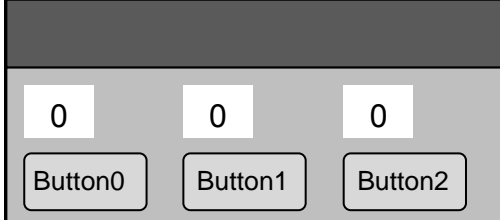
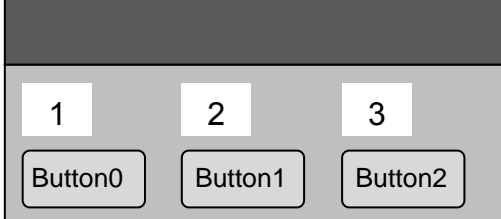
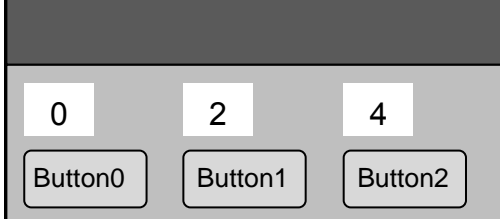
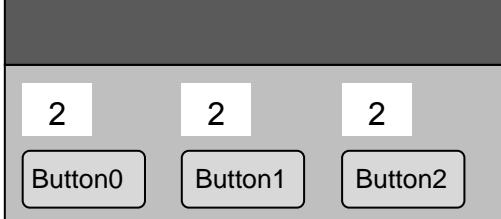
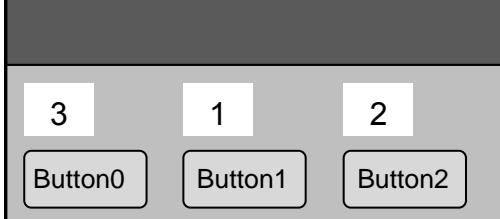
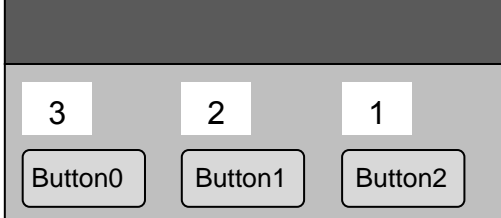
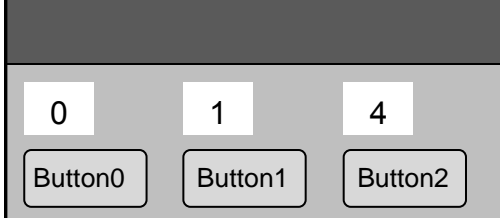
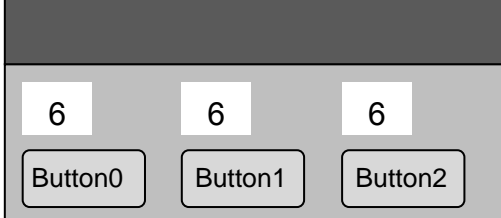
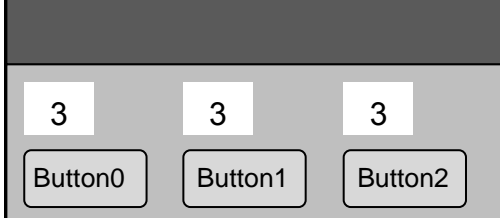
addListeners2()

שאלה 24 (4 נק')

כיצד יראה החלון לאחר הרצת התוכנית ולחיצה על שלושת הכפתורים משמאל לימין אם הקוד החסר הוא

addListeners3()

מפתח תשובות ל-4 השאלות הקודמות:

<p>ב.</p> 	<p>א.</p> 
<p>ד.</p> 	<p>ג.</p> 
<p>י.</p> 	<p>ה.</p> 
<p>נ.</p> 	<p>ז.</p> 
<p>ו.</p> 	<p>ט.</p> 

4 השאלות הבאות (נקודה אחת לכל שאלה) מתייחסות לקוד הבא. בשאלה הזאת יש לנמק רק במקרה שאתם עונים שהקוד לא מתקמפל.

```
public class Test {

    public interface If {}
    public class A implements If {}
    public class B extends A {}

    public class Helper<T> {
        public Helper(T arg) {}
    }

    private void test() {
        Helper<If> h1 = new Helper<If>(new A());
        Helper<If> h2 = new Helper<If>(new B());
        Helper<If> h3 = new Helper<A> (new A());
        Helper<A> h4 = new Helper<A> (new B());
    }
}
```

שאלה 25: השורה שמגדירה את h1 נימוק לתשובה ב':
א. מתקמפלת
ב. אינה מתקמפלת

שאלה 26: השורה שמגדירה את h2 נימוק לתשובה ב':
א. מתקמפלת
ב. אינה מתקמפלת

שאלה 27: השורה שמגדירה את h3 נימוק לתשובה ב':
א. מתקמפלת
ב. אינה מתקמפלת

שאלה 28: השורה שמגדירה את h4 נימוק לתשובה ב':
א. מתקמפלת
ב. אינה מתקמפלת