

בחינה לדוגמה בתוכנה 1

משך הבחינה שלוש שעות.

יש לסמן את התשובה הטובה ביותר בתשובון וכן בטופס 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. יחד עם זאת, ככלל הנימוקים יבדקו רק במקרים של ערעור או של חשד להעתקה. 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, 20 נקודות

השאלות הבאות מתייחסות לקוד הבא, כאשר בכל שאלה התווים `/**/` בשגרה `main` מוחלפים בקטע קוד אחר.

```
public class B {  
  
    private int f = 0;  
  
    public static void main(String[] args) {  
        B b1 = new B();  
        B b2 = new B();  
        Object b3 = b1;  
  
        System.out.println(***/);  
    }  
  
    @Override  
    public boolean equals(Object obj) {  
        f++;  
        B b = (B) obj;  
        return b.f == (f - 1);  
    }  
  
    public boolean eq(B b) {  
        return super.equals(b);  
    }  
}
```

1. `b1.equals(null)`
 - a. יודפס `true`
 - b. יודפס `false`
 - c. שגיאת קומפילציה
 - d. יזרק חריג בזמן ריצה

נימוק ל-1:

2. `b1.equals(b2)`
 - a. יודפס `true`
 - b. יודפס `false`
 - c. שגיאת קומפילציה
 - d. יזרק חריג בזמן ריצה

נימוק ל-2:

3. `b3.equals(b3)`
 - a. יודפס `true`
 - b. יודפס `false`
 - c. שגיאת קומפילציה
 - d. יזרק חריג בזמן ריצה

נימוק ל-3:

4. `b2.eq (b2)`
 - a. יודפס `true`
 - b. יודפס `false`
 - c. שגיאת קומפילציה
 - d. יזרק חריג בזמן ריצה

נימוק ל-4:

5. `b1.eq(b3)`
 - a. יודפס `true`
 - b. יודפס `false`
 - c. שגיאת קומפילציה
 - d. יזרק חריג בזמן ריצה

נימוק ל-5:

שאלה 2, 33 נקודות

יש להניח שקטעי הקוד הבאים מופיעים כחלק משירותים אלא אם הם עטופים באופן מפורש בהגדרה של מחלקה.

```
6. double d = 1.0 / 0.0;
   if (d > 1.0)
       System.out.println("X");
   else
       System.out.println("Y");
```

-
- a. לא יתקמפל בגלל חלוקה באפס
 - b. יתקמפל אבל התוכנית תזרוק חריג כאשר היא תנסה לחלק באפס
 - c. יודפס X
 - d. יודפס Y

נימוק ל-6:

```
7. int i;
   System.out.println(i);
```

-
- a. לא יתקמפל
 - b. יתקמפל אבל יזרוק חריג null pointer exception
 - c. ידפיס 0
 - d. ידפיס i

נימוק ל-7:

```
8. int d=1;
   switch(d) {
       case 0: System.out.print("A");
       case 1: System.out.print("B");
       case 2: System.out.print("C");
       default: System.out.print("D");
   }
```

-
- a. B
 - b. BC
 - c. BCD
 - d. לא יתקמפל

נימוק ל-8:

```
9. public class A {
    public static void main(String[] arr) {
        System.out.println(arr[2]);
    }
}
```

We run this program with the arguments "hello world".

- a. לא יתקמפל
- b. ידפיס hello
- c. יזרוק חריג NullPointerException
- d. יזרוק חריג אחר

נימוק ל-9:

```

10. public class A {
    private int i;
    public void copy(final A a, int i) { a.i = i; }
    public static void main(String[] arg) {
        A a = new A();
        a.copy(a,4);
        System.out.println(a.i);
    }
}

```

- a. FinalModificationException יזרק חריג
 b. 4 יודפס
 c. לא יתקמפל בגלל השורה השלישית
 d. לא יתקמפל בגלל השורה השביעית

נימוק ל-10:

```

11. public class Outer {
    private int i;

    private class Inner {
        private int j = i++;
        public int g() { return j; }
    }

    public Object f(){
        Inner inner = new Inner();
        return inner;
    }
}

public class A {
    public static void main(String[] args){
        Object o = new Outer().f();
        Outer.Inner inner = (Outer.Inner) o;
        System.out.println(inner.g());
    }
}

```

- a. 0 יודפס
 b. 1 יודפס
 c. לא יתקמפל
 d. יזרק חריג

נימוק ל-11:

איזה מהפלטים בהמשך אי אפשר לקבל מהקוד הבא. 12.

```
public static void randomPrint(){
    Random rand = new Random();
    List<Integer> l1 = new ArrayList<Integer>();
    List<Integer> l2 = new ArrayList<Integer>();
    for (int i = 1; i < 5; i++){
        if (rand.nextDouble() > 0.5){
            l1.add(i);
        }
        else{
            l2.add(i);
        }
    }
    Iterator<Integer> it1 = l1.iterator();
    Iterator<Integer> it2 = l2.iterator();
    while (it1.hasNext() || it2.hasNext()){
        if(rand.nextDouble() > 0.5 && it1.hasNext()){
            System.out.print(it1.next() + " ");
        }
        else if (it2.hasNext()){
            System.out.print(it2.next() + " ");
        }
    }
}
```

a. 1 2 3 4
b. 1 3 2 4
c. 1 4 3 2
d. 2 1 4 3
e. 3 4 1 2

נימוק ל-12:

13. מה הפלט של השורות הבא.

```
public void foo(){  
    int sum = 0;  
    for (int i = 0; i < 5; i++){  
        sum += (i++);  
    }  
    System.out.println(sum);  
}
```

- a. 6
- b. 9
- c. 10
- d. 12
- e. 15

נימוק ל-13:

שאלה 3, 20 נקודות

נתונות שתי המחלקות הבאות:

```
public class A {
    public void f(A a) {
        System.out.print("in A ");
    }
}

public class B extends A {
    public static void main(String[] args) {
        B b = new B();
        A a = new A();
        b.f(b);
        b.f(a);
    }
}
```

לגבי כל אחד מהשירותים הבאים: האם השירות יכול להיות מוגדר כשירות של המחלקה B בקוד למעלה? אם לא (כלומר אם לדעתך הקוד לא יתקמפל), יש להסביר מדוע. אם כן, יש להסביר מה קורה כאשר מריצים את `B.main`.

```
14. protected void f(A a) {
    System.out.print("in B ");
}
```

- a. לא יתקמפל
- b. יודפס in A in A
- c. יודפס in B in A
- d. יודפס in B in B

נימוק ל-14:


```
15. public void f(A a) {  
    System.out.println("in B");  
    throw new java.io.IOException();  
}
```

-
- a. לא יתקמפל
 - b. יודפס in A in A
 - c. יודפס in B in A
 - d. יודפס in B in B

נימוק ל-15:

```
16. public void f(B b) {  
    System.out.println("in B");  
}
```

-
- a. לא יתקמפל
 - b. יודפס in A in A
 - c. יודפס in B in A
 - d. יודפס in B in B

נימוק ל-16:

```
17. public void f(Object o) {  
    System.out.println("in B");  
}
```

-
- a. לא יתקמפל
 - b. יודפס in A in A
 - c. יודפס in B in A
 - d. יודפס in B in B

נימוק ל-17:

שאלה 4, 27 נקודות

בשאלה זו נעסוק בתוכנית המייצרת ממשק משתמש גרפי הכולל חלון ובו 3 כפתורי לחיצה (ראו תמונה למטה). במקטע הקוד המובא בהמשך, המתודה `createShell` מכילה מקטע קוד חסר. בכל אחד מהסעיפים הבאים, יחליף מקטע קוד מסוים את מקטע הקוד החסר.



```
public class GUIMain {
    private static Display display;
    private static int num = 0;

    public static void main(String[] args) {
        Shell shell = createShell();
        eventLoop(shell);
    }

    private static void eventLoop(Shell shell) {
        shell.pack();
        shell.open();
        while (!shell.isDisposed()) {
            if (!display.readAndDispatch()) {
                display.sleep();
            }
        }
        display.dispose();
    }

    private static Shell createShell() {
        display = Display.getDefault();
        Shell shell = new Shell(display);
        shell.setLayout(new RowLayout(SWT.HORIZONTAL));

        /* missing code, multiple completions below */

        return shell;
    }
}
```

הערות:

- ניתן להניח שהקוד (לאחר השלמת המקטע החסר) עובר קומפילציה ורץ ללא זריקת חריג.
- ה- `layout` בקוד `RowLayout(SWT.HORIZONTAL)` יגרום ליצירת הכפתורים משמאל לימין.
- רצף לחיצה מתאר מצב שבו לאחר פתיחת החלון של הממשק הגרפי, המשתמש לוחץ על הכפתורים בסדר מסוים. למשל, רצף הלחיצה $3 \rightarrow 2 \rightarrow 1$ מתאר: לחיצה בודדת על הכפתור הראשון (השמאלי ביותר), לאחר מכן על הכפתור השני (האמצעי) ולבסוף לחיצה על הכפתור השלישי (הימני ביותר).

בסעיפים 18-26, עליכם לציין מהו מצב הכפתורים לאחר רצף הלחיצות הנתון בכל סעיף מבין האפשרויות a-h הבאות:

| | | | |
|----|----------|----------|----------|
| a. | Clicked! | Clicked! | Clicked! |
| b. | ? | Clicked! | Clicked! |
| c. | ? | ? | ? |
| d. | Clicked! | ? | ? |
| e. | ? | ? | Clicked! |
| f. | Clicked! | ? | Clicked! |
| g. | Clicked! | Clicked! | ? |
| h. | ? | Clicked! | ? |

אם נציב במקטע הקוד החסר במתודה createShell את מקטע הקוד הבא, מה תהיה התוצאה של רצפי הלחיצה שבהמשך:

```

for (int i = 0; i < 3; i++) {
    Button b = new Button(shell, SWT.PUSH);
    b.setText(" ? ");
    b.addListener(new SelectionAdapter() {
        boolean clicked = false;

        public void widgetSelected(SelectionEvent e) {
            if (!clicked) {
                Button b = (Button) e.getSource();
                b.setText("Clicked!");
                clicked = true;
            }
        }
    });
}

```

- 18. 1→2→3 _____: התוצאה
- 19. 3→2→2 _____: התוצאה
- 20. 1→1→1 _____: התוצאה

נימוק ל-18-20:

אם נציב במקטע הקוד החסר במתודה createShell את מקטע הקוד הבא:

```
final Button[] barr = new Button[3];
barr[0] = new Button(shell, SWT.PUSH);
barr[0].setText(" ? ");
for (int i = 1; i < 3; i++) {
    SelectionAdapter a = new SelectionAdapter() {
        boolean clicked = false;
        public void widgetSelected(SelectionEvent e) {
            if (!clicked) {
                Button b = (Button) barr[barr.length - 1];
                b.setText("Clicked!");
                clicked = true;
            }
        }
    };
    barr[i] = new Button(shell, SWT.PUSH);
    barr[i].setText(" ? ");
    barr[i].addSelectionListener(a);
}
```

- 21. 1→2→3 _____: התוצאה
- 22. 3→2→2 _____: התוצאה
- 23. 1→1→1 _____: התוצאה

נימוק ל-22-24:

אם נציב במקטע הקוד החסר במתודה createShell את מקטע הקוד הבא:

```
final Button[] barr = new Button[4];
for (int i = 0; i < 3; i++) {
    barr[i] = new Button(shell, SWT.PUSH);
    barr[i].setText(" ? ");
    barr[i].addListener(new SelectionAdapter() {
        boolean clicked = false;

        public void widgetSelected(SelectionEvent e) {
            Button b = (Button) e.getSource();
            if (barr[num] == b) {
                b.setText("Clicked!");
                num++;
            } else if (b.getText().equals("Clicked!")) {
                b.setText(" ? ");
            }
        }
    });
}
```

24. $1 \rightarrow 2 \rightarrow 3$ _____: התוצאה:
 25. $3 \rightarrow 2 \rightarrow 2$ _____: התוצאה:
 26. $1 \rightarrow 1 \rightarrow 1$ _____: התוצאה:

נימוק ל-24-26: