

מס' שאלון - 510 27  
בינואר 2022

מס' מועד 61

סמסטר 2022א

20554 / 4

שאלון בחינת גמר

20554 - תכנות מתקדם בשפת Java

משך בחינה: 3 שעות
-------------------

בשאלון זה 12 עמודים

מבנה הבחינה:

בבחינה ארבע שאלות.

עליכם לענות על כולן.

משקל כל שאלה מופיע בגוף השאלון.

יש לענות על גבי טופס הבחינה.

חומר עזר:

ספר הקורס: Java: How to Program, 11th edition, Pearson Ed., 2018

ומדריך הלמידה בלבד.

שימוש באינטרנט ובסביבת הפיתוח ובתוכנות Java אחרות אסורה בהחלט!!!

בהצלחה !!!

שאלה 1 (28%) (יש לכתוב את התשובה על גבי טופס הבחינה)

א. (10%) נתונות המחלקות הבאות (כל אחת בקובץ נפרד ושתייהן באותו המארז):

```
public class Parent {
    public void func(Parent a) {
        System.out.print("123 ");
    }
}
public class Child extends Parent {
    // הוסיפו כאן את המתודה
    public static void main(String[] args) {
        Child b = new Child();
        Parent a = new Parent();
        b.func(b);
        b.func(a);
    }
}
```

לגבי כל אחת מהמתודות הבאות, האם ניתן להוסיף אותן (כל אחת בנפרד) למחלק B (במקום המסומן)? אם לא, הסבירו מדוע, ואם כן - מה יודפס כתוצאה מהרצת המתודה main במחלקה B. **בחרו את התשובה הנכונה ובמקרה של שגיאה פרטו מה השגיאה:**

1.

```
protected void func(Parent a) {
    System.out.print("456 ");
}
```

א. 123 123    ב. 456 123    ג. 456 456    ד. שגיאת קומפילציה

תשובה:

2.

```
public void func(Parent a) {
    System.out.print("456 ");
    throw new java.io.IOException();
}
```

א. 123 123    ב. 456 123    ג. 456 456    ד. שגיאת קומפילציה

תשובה:

3.

```
public void func(Parent a) {
    System.out.print("456 ");
}
```

א. 123 123    ב. 456 123    ג. 456 456    ד. שגיאת קומפילציה

תשובה:

4.

```
public void func(Child b) {
    System.out.print("456 ");
}
```

א. 123 123    ב. 456 123    ג. 456 456    ד. שגיאת קומפילציה

תשובה:

5.

```
public void func(Object o) {
    System.out.print("456 ");
}
```

א. 123 123    ב. 456 123    ג. 456 456    ד. שגיאת קומפילציה

תשובה:

ב. (4%) נתונות המחלקות הבאות (כל אחת בקובץ נפרד ושתייהן באותו המארג):

```
public class One {
    public One() {
        aa();
        bb();
    }
    private void aa() {
        System.out.println("aa One");
    }
    void bb() {
        System.out.println("bb One");
    }
}
public class Two extends One {
    public void aa() {
        System.out.println("aa Two");
    }
    public void bb() {
        System.out.println("bb Two");
    }
}
public class Main {
    public static void main(String[] args) {
        One one = new Two();
    }
}
```

האם הקוד מתקמפל? אם לא הסבירו מדוע, אם כן, מה יודפס כתוצאה מהרצת המתודה main במחלקה Main?

תשובה:

ג. (4%) בחרו את הטענה הנכונה :

1. שדות סטטיים מאותחלים עם טעינת המחלקה, יש עותק יחיד בתכנית, והם נגישים ממתודות סטטיות וממתודות מופע.
2. שדות סטטיים מאותחלים עם טעינת המחלקה, יש עותק יחיד בתכנית, והם נגישים ממתודות סטטיות בלבד.
3. שדות סטטיים מאותחלים עם יצירת אובייקט, יש עותק יחיד בתכנית, והם נגישים ממתודות סטטיות בלבד.
4. שדות סטטיים מאותחלים עם יצירת אובייקט, יש עותק יחיד עבור כל מופע, והם נגישים ממתודות סטטיות וממתודות מופע.

הטענה הנכונה היא :

ד. (4%) נתון הקוד הבא : (כל אחת בקובץ נפרד ושתייהן באותו מארז) :

```
public class Foo {
    private int data;
    class Fool {
        private int d = data++;
        public int getD() { return d; }
    }
    public Object f(){
        Fool fool = new Fool();
        return fool;
    }
}

public class FooTest {
    public static void main(String[] args){
        Object o = new Foo().f();
        Foo.Fool fool = (Foo.Fool)o;
        System.out.println(fool.getD());
    }
}
```

מה יודפס כתוצאה מהרצת המחלקה Main? בחרו את התשובה הנכונה ונמקו :

1. יודפס 0
2. יודפס 1
3. שגיאת קומפילציה
4. שגיאת זמן ריצה

תשובה :

ה. (6%) נתונה המתודה הבאה:

```
1. public class Test {  
2.     public static void test(List<Integer>lst){  
3.         for(Integer i: lst){  
4.             try{  
5.                 System.out.print(i);  
6.             }catch(NullPointerException e){  
7.                 System.out.print("1");  
8.             }finally{  
9.                 System.out.print("0");  
10.            }  
11.        }  
12.        System.out.println("\n");  
13.    }  
14. }
```

איזה פלט יתקבל עבור הקריאה למתודה עם רשימה המכילה את האיברים הבאים:

7, null, 8

בחרו את התשובה הנכונה:

1. תודפס המחרוזת: 70 ואחרי זה יזרק מצב חריג. במידה והקורא החיצוני לא טיפל במצב החריג, תודפס הודעת שגיאה.
2. תודפס המחרוזת: 70
3. תודפס המחרוזת: 70null80
4. תודפס המחרוזת: 7null18
5. תודפס המחרוזת: 70null080
6. אף תשובה אינה נכונה

תשובה:

אם נשנה את הטיפוס בשורה 3 באופן הבא::

```
for(int i: lst){
```

איזה פלט יתקבל עבור אותה הקריאה? בחרו מתוך התשובות שהוצגו לעיל:

תשובה:

## שאלה 2 (23%) (יש לכתוב את התשובה על גבי טופס הבחינה)

כתבו תכנית המשמשת לתרגול קואורדינציה הקשורה בשימוש בעכבר. ממשק המשתמש יכלול משטח ציור ריק שעליו יוצגו בזה אחר זה  $n$  ריבועים במקומות אקראיים. כל ריבוע יוצג במשך  $x$  מילישניות ולאחר מכן יעלם. המשתמש צריך ללחוץ עם העכבר על כל ריבוע שמופיע על המסך. המטרה היא ללחוץ על כמה שיותר ריבועים, אך לא לבצע לחיצות מיותרות או מחוץ לתחומי הריבוע המוצג. התכנית תספור את מספר הלחיצות המוצלחות (לחיצות שפגעו בריבוע) ואת מספר הלחיצות שהחטיאו.

בסיום הצגת  $n$  הריבועים התכנית תציג תיבת דו-שיח הכוללת ציון סופי המחושב באופן הבא: מספר הלחיצות שפגעו בריבוע  $2 * \text{פחות מספר הלחיצות שהחטיאו}$ .

המשתנים  $n$  ו- $x$  יוגדרו כקבועים בתכנית. הריבועים שיוצגו יהיו בגודל קבוע של  $10 * 10$  וגודל החלון יוגדר כ-  $400 * 400$ .

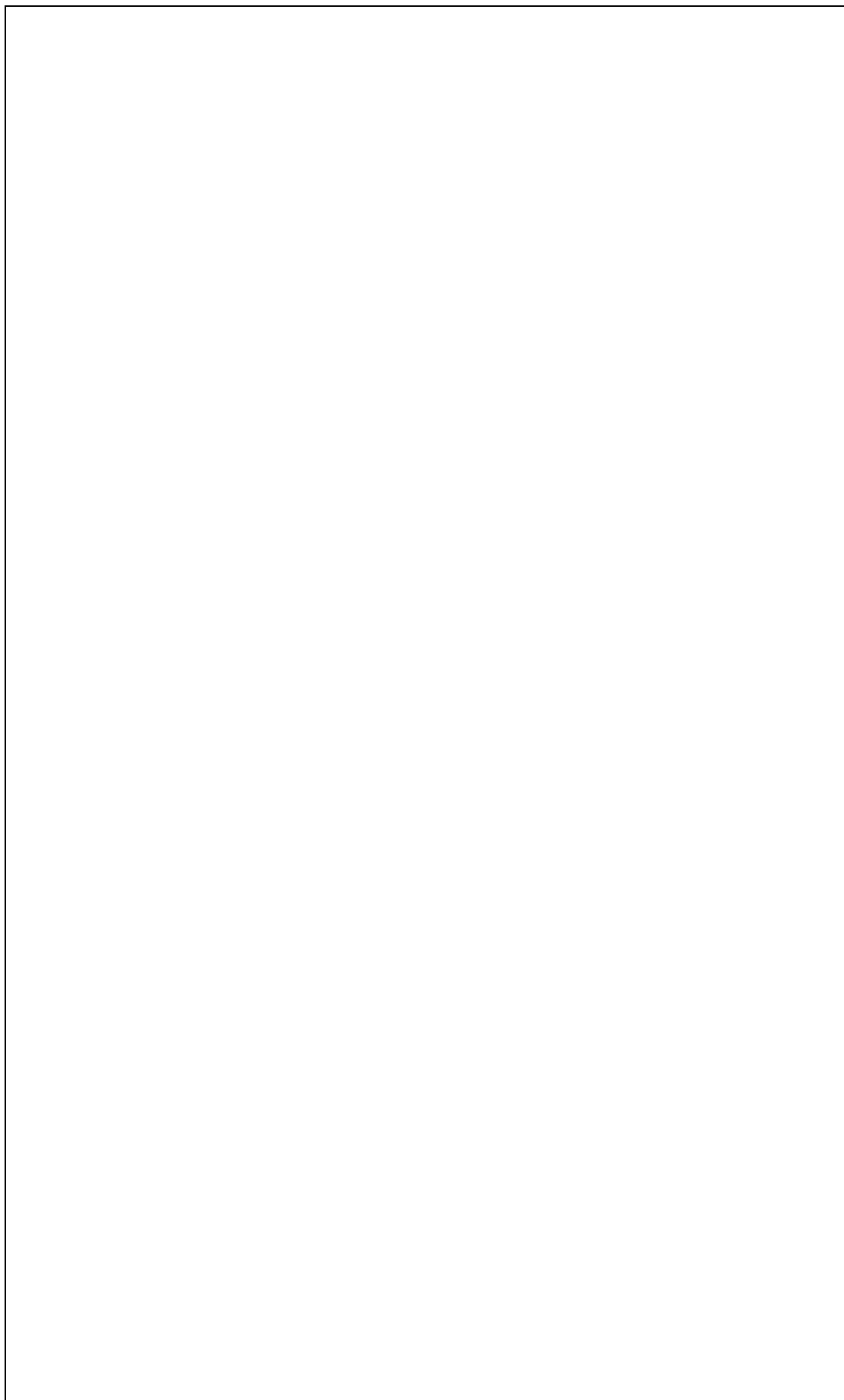
### לסטודנטים שלמדו את הקורס החל מסמסטר 2022א:

- א. פרטו במילים באיזה רכיבים אתם משתמשים וכיצד הם מאורגנים בתוך החלון. בנוסף ציינו את שמות הרכיבים (`fx:id`) ואת האירועים הרלוונטיים בממשק המשתמש הגראפי.
- ב. כתבו את מחלקת הקונטרולר המבצעת את הנדרש כפי שתואר לעיל. הניחו שממשק המשתמש נבנה ב-SceneBuilder כפי שצוין בסעיף א, וכתבו במחלקת הקונטרולר את כל מה שנדרש כולל הגדרת הרכיבים השונים ומתודות טיפול באירועים.

### לסטודנטים שלמדו את הקורס לפני 2022א:

כתבו תכנית שמבצעת את הנדרש לעיל. יש לאפשר למשתמש ללחוץ על ה- $x$  בכדי לסיים את התכנית.

תשובה:



**שאלה 3 (25%) (יש לכתוב את התשובה על גבי טופס הבחינה)**

נתון הממשק הג'נרי `Codeable` הבא:

```
public interface Codeable <T>{  
    public T code() throws NonCodeable;  
}
```

הממשק משמש ל"קידוד" אובייקטים בצורה כלשהי המוגדרת ע"י המחלקה המממשת. דוגמאות אפשריות לקידוד מחרוזת או מערך הן: הזזה מעגלית של סדר האותיות במחרוזת, היפוך סדר האיברים במערך וכו'. המחלקה המממשת את הממשק תממש אותו בצורה כזו שהמתודה `code` כך שתחזיר גרסה מקודדת של האובייקט עליו היא הופעלה. המתודה לא תשנה את האובייקט המקורי.

המצב החריג `NonCodeable` המוגדר כבן של `Exception` מתעורר כאשר האובייקט אינו ניתן לקידוד.

א. (8%) נתונה המחלקה `Text` המייצגת טקסט כלשהו באמצעות מחרוזת:

```
public class Text{  
    private String st=""; // טקסט כלשהו  
    public Text(String st){  
        this.st = st;  
    }  
    ...  
}
```

ממשו במחלקה `Text` את הממשק הג'נרי `Codeable<T>` כך שהמתודה `code` תחזיר אובייקט `Text` שהמחרוזת שלו הוזזה תו אחד ימינה בצורה מעגלית (כך שהתו האחרון יופיע במקום הראשון). למשל אם הטקסט כלל את המחרוזת "abcdefgh" אז אובייקט הטקסט שיוחזר יכלול את המחרוזת "habcdefgh". אם אורך המחרוזת קטן מ-2, יש לעורר ולייצא את המצב החריג `NonCodeable`.

**שימו לב** שהמתודה `code` לא תשנה את המחרוזת המקורית באובייקט הטקסט, אלא תחזיר אובייקט טקסט חדש עם מחרוזת מקודדת.

**הערה:** אין צורך לכתוב את המחלקה `NonCodeable`. ניתן להניח שהיא קיימת.



ב. (11%) כתבו מתודה ג'נרית בשם `coder` המקבלת `ArrayList` מסוג כלשהו המממש את הממשק `Codeable<T>`. המתודה תחזיר `ArrayList` חדש שבו כל האיברים מקודדים. איברים שלא ניתן לקודד יועברו כפי שהם.  
הערה: האיברים ב- `ArrayList` שהתקבל כפרמטר ישארו ללא שינוי.

תשובה:

ג. (6%) נתונות ההצהרות הבאות המסתמכות על הממשק הגנרי `Codeable<T>` שהוגדר לעיל, והמחלקה `Text` כפי שכתבתם בסעיף א:

```
Codeable<?> data1=null;
Codeable<Object> data2=null;
Codeable<Text> data3=null;
public void codeTest(Codeable<Object> data){...}
```

ציינו לגבי כל אחת מהקריאות הבאות למתודה `codeTest` אם היא תקינה או לא תקינה:

1. `codeTest (data1);` תקין / לא תקין

תשובה:

2. `codeTest (data2);` תקין / לא תקין

תשובה:

3. `codeTest (data3);` תקין / לא תקין

תשובה:

שאלה 4 (24%) (יש לכתוב את התשובה על גבי טופס הבחינה)

נתונות המחלקות `WorkThread` ו-`Test` הבאות (כל אחד בקובץ נפרד):

```
1. public class WorkThread extends Thread{
2.     private int[] vec;
3.     private int id;
4.     private int result;
5.     public WorkThread (int[] vec, int id){
6.         this.vec = vec;
7.         this.id = id;
8.     }
9.     public static synchronized int process(int[] vec, int id){
10.         int result = 0;
11.         for (int i = 0; i < vec.length; i++){
12.             vec[i] = vec[i] + 1;
13.             result = result + vec[i];
14.         }
15.         return result;
16.     }
17.     public void run (){
18.         result = process(vec, id);
19.         System.out.println("task" + id + " result=" + result);
20.     }
21. }
```

```

1. public class Test{
2.     public static void main(String [] args){
3.         int[] vec = {1,2,3,4};
4.         WorkThread[] workers = new WorkThread[3];
5.         for (int i=0; i<3; i++)
6.             workers[i] = new WorkThread(vec,i);
7.         for (int i=0; i<3; i++)
8.             workers[i].start();
9.         System.out.println("**main**");
10.    }
11. }

```

המתודה main במחלקה Test יוצרת 3 תהליכים מסוג WorkThread. כל תהליך מפעיל את המתודה process המוסיפה 1 לכל אחד מאיברי המערך ומחזירה את סכום האיברים.

א. (4%) בהינתן התכנית שהוצגה לעיל, בחרו את התשובה הנכונה ונמקו:

1. שלושת התהליכים מעדכנים בו זמנית את אותו המערך.
2. שלושת התהליכים מעדכנים את אותו המערך אך לא בו זמנית.
3. שלושת התהליכים מעדכנים בו זמנית אך כל תהליך מעדכן מערך משלו.
4. כל אחד מהתהליכים מעדכן מערך משלו אך העדכונים אינם מתבצעים בו זמנית.

תשובה:

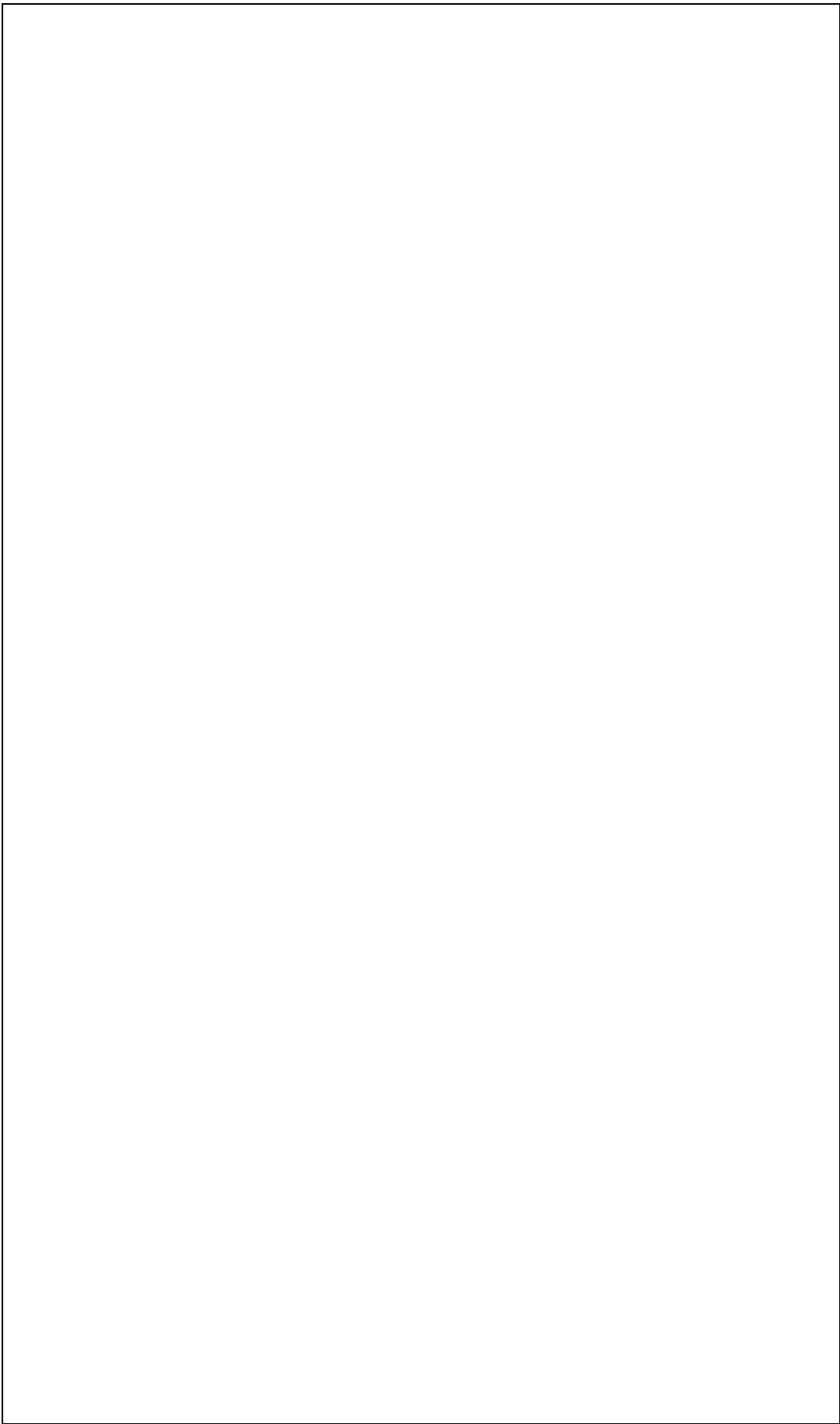
ב. (4%) התייחסו למשפט הבא וציינו (נכון / לא נכון) ונמקו את תשובתכם:

הורדת המאפיין static (בלבד) מהמתודה process תגרום לכך ששלושת התהליכים יוכלו לבצע את המתודה בו זמנית.

תשובה:

ג. (16%) שנו את המחלקה WorkThread כך שההדפסות המבוצעת ע"י התהליך WorkThread יתבצעו לפי סדר ה-id, כלומר הדפסה ראשונה תהייה של תהליך בעל id=0, ההדפסה השנייה תהייה של תהליך בעל id=1 וכן הלאה. השינוי צריך להתבצע בתוך המחלקה WorkThread מבלי לשנות או להוסיף מחלקות נוספות ומבלי לבצע המתנה פעילה. ניתן להניח שמספר ה-id הראשון הוא 0.

תשובה:



**בהצלחה !**