

דוח משוב לסטודנט

מזהה סטודנט

0336540331

מזהה מבחן

000000092972

תאריך בחינה

יום שלישי, 24 ביוני 2025

מזהה קורס

7284-50988

שם קורס

תכנות מונחה עצמים

מרצה

כהן איינהורן ענבל

ניקוד שאלות פתוחות	ציון מבחן מקורי	ציון מבחן סופי
49.50	49.50	49.50

סיכום

מספר שאלה	הערות	ניקוד	ניקוד מירבי
1.1	חסר תיקון!	2.50	3.00
1.2		12.50	15.00
1.3		1.00	3.00
1.4.1		3.00	4.00
1.4.2		1.50	4.00
1.4.3		0	1.00
2.1		7.00	10.00
2.2		9.00	15.00
2.3		6.50	10.00
3.1	זה לא מעקב אובייקטים כמו שלמדנו - חסר טיפוס הפניות ואובייקטים	3.00	5.00
3.2	חסרים הסברים והפלט שגוי	3.50	16.00
3.3		0	6.00
3.4		0	2.00
3.5		0	3.00
3.6		0	3.00



הטכניון
מכון-טכנולוגי לישראל

משרד העבודה
זרוע העבודה

מה"ט המכון הממשלתי להכשרה בטכנולוגיה ובמדע

בית הספר הארצי להנדסאים (ע"ר)

מחברת בחינה

הוראות לנבחן

(22)

ID 0336540331

Exam 000000092972

1. לפני תחילת הבחינה אנא קרא בעיון את ההוראות ומלא את הפרטים בכתב יד ברור.
שים לב: מחברות הבחינה נסרקות למאגר נתונים.
יש להקפיד: לא לקפל, לא לתלוש ולא לכתוב בצבעים או בעפרון.
2. כתוב את פרטיך האישיים על גבי מחברת הבחינה מיד עם תחילת הבחינה.
3. כתוב את תשובותיך בכתב יד ברור **משני צידי הדף.**
4. ציין בראש כל דף את מספר השאלה עליה אתה עונה.
5. כתוב את כל עבודתך (כולל הטייטה) במחברת זו. אל תכתוב דבר על דף אחר.
6. מתח קו אלכסוני ורשום "מבוטל" על אותם חלקים שאינך רוצה שייבדקו.
7. אם מחברת זו לא תספיק לך, בקש מהמשיגי מחברת נוספת. כתוב על כל מחברת את פרטיך האישיים.
8. הוצאת מחברת בחינה (או חלק ממנה) מחוץ לחדר הבחינה הינה עבירה משמעתית חמורה.
9. על הנבחן לציית להוראות המשיגים. כל פנייה למשיגי תיעשה על ידי הרמת יד בלבד.
10. נבחן שיפריע למהלך הבחינה, יועמד לדין משמעת של ביה"ס. ניתן להפסיק את בחינתו ולתבעו בוועדת משמעת של ביה"ס.
11. נבחן לא יהיה רשאי לעזוב את חדר הבחינה, אלא על פי נהלי בית הספר ובאישור המשיגי.
12. מותר להשתמש בחומר עזר רק אם הדבר הותר במפורש בכתב בגוף שאלון הבחינה. הימצאות חומר אסור בידי הנבחן בשעת הבחינה על שולחנו, או בסמוך אליו, מהווה עבירה משמעת חמורה.

הערה: ההוראות מתייחסות לזכר ולנקבה כאחד.

תאריך הבדיקה: _____

חתימת המרצה: _____

חתימת הסטודנט

ציין:

כאן ס' / 19:30

בהצלחה!

לתשומת לבך:

יש לכתוב את תשובותיך משני צידי הדף
אין לכתוב מעבר לקו האדום
יש לכתוב את הבחינה בעט (כחול/שחור) בלבד.

שם משפחה: NE'IC'ן שם פרטי: דן

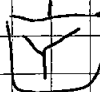
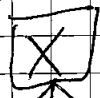
מספר תעודת זהות: 1 3 3 0 4 5 6 3

תאריך הבחינה: 06.07.2025 מגמה: תכנה מחזור: _____

שם הקורס: תכנות מונחה שם המרצה: דן

סמסטר: אביב / קיץ מועד: אמצע / ב / מיוחד

מחברת מספר 1 מתוך סה"כ מחברות 1



$\checkmark X \times 1 = \text{new } Y() - \text{upCast} \checkmark$

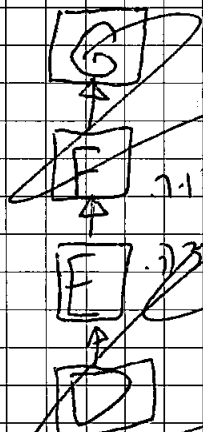
$\text{Object } o = x1; - \text{upCast. (i)}$
 אין עז'אב' X יורש Object, sic, Object N
 לכתוב הורכוק ✓

$X \times 2 = x1; - \text{אין עז'אב'}$
 כי X הוא פרמטר מט'אב' X ✓

$+ (iii) Y \times 1 = x1; - \text{אין עז'אב' ח'צור}$
 Y יורש X ולא לכתוב לכתוב לכתוב לכתוב
 מבט של Y (בן) לכתוב לכתוב X (אב) 2.9 (1.1) חסר תיקון! ✓

ג'ו'ו

(2)



$G = \text{new } (E)$ - אין עז'אב' G
 $F = (F) \text{new } (E)$ - אין עז'אב' F
 אין עז'אב' D

(2)

E יורש G

F יורש E

~~אין עז'אב' G~~

לכתוב לכתוב לכתוב לכתוב

G (אב)

E יורש G

(3)

בית הספר הארצי להנדסאים
מבחן סמסטר ב' מועד א' 2025 – תכנות מונחה עצמים

שם משפחה	שם פרטי	מס' ת. זהות	כיתה
עמ' אקין	דניאל	336540331	48-5

קורס: מחשבים 48/5+6

משך הבחינה: 150 דקות



מרצה: ענבל כהן-איינהורן.

תאריך: 06/07/2025

הוראות כלליות

1. במבחן זה 3 שאלות יש לענות על כל השאלות.
2. יש להשתמש אך ורק בחומר העזר שהינו קלסר אחד עם חומר הרצאות.
3. אין להוציא דפים מהקלסר.
4. אין להעביר חומר עזר מנבחן אחד למשנהו.
5. אין להעתיק תשובות מחומר עזר שהותר לשימוש.
6. יש לכתוב במחברת הבחינה בלבד! (גם טיוטה לסמן בקו ולרשום ליד טיוטה).
7. בדיקת המבחן תבצע בצירוף טופס הבחינה.
8. חל איסור מוחלט להוציא שאלון מחדר הבחינה!

בהצלחה!



שאלה 1: (30 נקודות)

בשאלה זו ארבעה סעיפים, א-ד. אין קשר בין הסעיפים. עליכם לענות על כל הסעיפים.

(3 נקודות)

א. נתונות: המחלקות X ו Y כך ש Y יורשת מ X.

ידוע שההוראה הבאה עוברת קומפילציה ורצה תקין:

X x1 = new Y();

עבור כל אחד מן המשפטים i-iii שלפניכם, ציינו אם הוא נכון או שגוי.

נמקו את תשובתכם. אם המשפט שגוי תקנו אם ניתן.

- | | | |
|------|--------------------------------------------------------------------|---|
| i. | לא ניתן לכתוב את ההוראה: Object o = x1; מכיוון שדרושה המרה מפורשת. | - |
| ii. | לא ניתן לכתוב את ההוראה: X x2 = x1; מכיוון שדרושה המרה מפורשת. | - |
| iii. | לא ניתן לכתוב את ההוראה: Y y1 = x1; מכיוון שדרושה המרה מפורשת. | + |

(15 נקודות)

ב. נתונה הפעולה main במחלקה Test ובה 4 הוראות ביצוע:

```
public static void main(String[] args) {
    (1) G g = new E();
    (2) F f1 = (F)(new E());
    (3) F f2 = new D();
    (4) E e = new D();
}
```

לפניכם חמש אפשרויות לעץ ירושה בין המחלקות.

עבור כל אחת מהאפשרויות i-v:

(1) ציירו תרשים UML, אם לא ניתן הסבירו מדוע

(2) במידה ו UML תקין, עבור כל אחת מהשורות (1-4):

○ ציינו האם מתרחשת שגיאה – כן / לא

אם כן: הסבירו מדוע וציינו את סוג השגיאה – קומפילציה או זמן ריצה

- | | |
|-----------------|-----------------|
| (1) G יורשת מ-E | (4) E יורשת מ-G |
| F יורשת מ-E | F יורשת מ-E |
| D יורשת מ-E | D יורשת מ-F |
| | G יורשת מ-F |
| (2) E יורשת מ-G | |
| G יורשת מ-F | (5) E יורשת מ-G |
| D יורשת מ-E | F יורשת מ-E |
| | D יורשת מ-E |
| (3) G יורשת מ-E | |
| F יורשת מ-E | |
| D יורשת מ-F | |



Company

(3 נקודות)

ג. נתונות הגדרות של המחלקות School ו-City

```
public class City {
    ____ (*) ____ int employees = 20;
}
```

```
public class Company extends City {
    public void poll() {
        System.out.println("There are " + employees + " employees in the company.");
    }
}
```

1) עליכם להשלים את השורה המסומנת (*) כך שלא תתרחש שגיאת קומפילציה. בחרו באחת מהאפשרויות הבאות, וציינו מהי האפשרות המתאימה ביותר לפי עקרונות תכנות מונחה עצמים (OOP). נמקו תשובתכם.

- i. לכתוב public , private או protected
- ii. לכתוב רק public או private
- iii. לכתוב רק public או protected
- iv. לכתוב רק protected
- v. לכתוב רק public
- vi. לכתוב רק private

(9 נקודות)

ד. לפניכם הגדרה של המחלקה Stam :

```
public class Stam {
    private int val;
    private static int count = 0;

    public Stam() {
        count++;
        this.val = count;
    }

    public Stam(boolean over) {
        if (!over) {
            count++;
            this.val = count;
        } else {
            this.val = -999;
        }
    }

    public void display() {
        System.out.println("val=" + val + ", count=" + count);
    }
}
```




1) כתבו פעולה ראשית main כך שבסופה תוצג שורת הפלט:

```
val=3, count=5
```

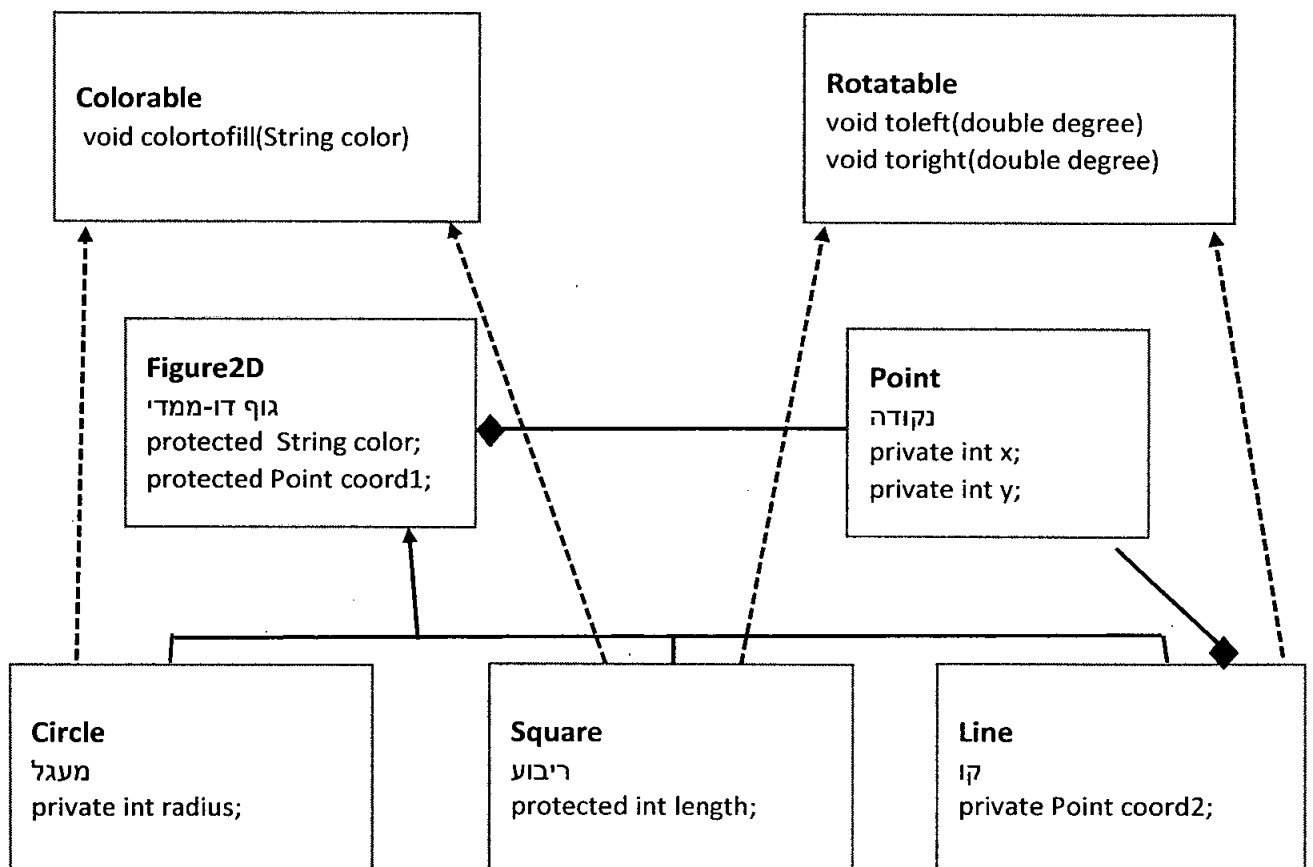
2) כתבו פעולה ראשית main כך שבסופה יוצגו 3 שורות הפלט:

```
val=2, count=5  
val=-999, count=5  
val=5, count=5
```

3) הסבירו מה ההבדל בין שני הבנאים של המחלקה Stam ומה תפקידו של כל בנאי?

שאלה 2: (35 נקודות)

חברת 'צורות' מפתחת תוכנה המאפשרת להגדיר גופים דו-ממדיים, ולבצע עליהן פעולות שונות. להלן תרשים UML חלקי של המערכת:





(10 נקודות)

- א. עבור כל אחת מחמש המחלקות שלפניכם יש להגדיר:
- a. כותרת המחלקה
 - b. תכונות המחלקה
 - c. כותרות של כל הפעולות שחייבות להיות במחלקה

(15 נקודות)

- ב. להלן קטע קוד. עבור כל אחת מהשורות (1–15), יש לציין:
- האם השורה תקינה או לא תקינה
 - אם אינה תקינה – יש לנמק ואם אפשר להציע תיקון
 - לציין את סוג השגיאה: שגיאת קומפילציה / שגיאת זמן ריצה

הניחו שכל תכונה במחלקות קיימות פעולות set ו-get מתאימות.

1.	Figure2D c = new Circle("red",new Point(50,50),8);
2.	Line line = new Line("blue",new Point(100,100),new Point(200,200));
3.	Square d2 = new Square("orange",new Point(45,45),4);
4.	Colorable col1 = new Colorable();
5.	Rotatable rot1;
6.	Colorable col2 = d2;
7.	rot1 = line;
8.	col2 = (Colorable)c;
9.	rot1.setColor("green");
10.	col2.colortofill("red");
11.	col2.toleft(Math.PI/2);
12.	((Square)d2).toleft(Math.PI/4);
13.	Figure2D d=c;
14.	((Line)d).setColor("red");
15.	d.colortofill("Yellow");

(10 נקודות)

- ג. כתבו פעולה חיצונית בשם handleFigures שמקבלת מערך של Figure2D ומבצעת את הפעולות הבאות:

- מדפיסה את נקודת cood1 של כל הצורות בצבע "blue"
- אם הצורה ניתנת למילוי תמלא אותה בצבע "red"
- אם הצורה ניתנת לסיבוב תסובב אותה 30 מעלות שמאלה
- אם הצורה היא Circle, הדפיסו את הרדיוס המעגל
- אם היא Line, הדפס את הנקודה השנייה שלה
- הפעולה תחזיר את מספר הצורות שניתנות לסיבוב אך אינן ניתנות למילוי

הניחו שכל תכונה במחלקות קיימות פעולות set ו-get מתאימות.



שאלה 3: (35 נקודות)

נתונת 2 המחלקות A ו B:

<pre> public class A { private int val; public A() { this.val = 1; } public A(int val) { this.val = val; } public int getVal() { return this.val; } public boolean equals(Object other) { System.out.println("A-Object"); if (other instanceof A) return this.val == ((A) other).val; return false; } } </pre>	<pre> public class B extends A { private String st; public B() { this.st = "B"; } public B(String st, int val) { super(val); this.st = st; } public String getSt() { return this.st; } public boolean equals(Object other) { System.out.println("B-Object"); if (other instanceof B) return this.st.equals(((B) other).st) && this.getVal() == ((B) other).getVal(); return false; } public boolean equals(A other) { System.out.println("B-A"); if (other instanceof B) return this.st.equals(((B) other).st) && this.getVal() == ((B) other).getVal(); return false; } public boolean equals(B other) { System.out.println("B-B"); return this.st.equals(other.st) && this.getVal() == other.getVal(); } } </pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



(5 נקודות)

א. נתונה המחלקה Test:

```
public class Test {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = new A(3);
        A ab = new B();
        B b1 = new B("B", 1);
        B b2 = new B("B", 3);
        // *** //
    }
}
```

עקבו אחרי יצירת האובייקטים בפעולה הראשית main.
עבור כל אחד מהם עליכם לציין במעקב את ערכי כל התכונות של כל אובייקט שנוצר.

(16 נקודות)

ב. לפניכם 8 הוראות ביצוע.

הציבו כל אחת מההוראות במקום // *** // בפעולה הראשית,
כתבו מה יהיה הפלט בעבור כל אחת מההוראות והסבירו בקצרה את הסיבה.

- 1) if (a1.equals(b1)) System.out.println(11);
- 2) if (b1.equals(a1)) System.out.println(12);
- 3) if (a1.equals(ab)) System.out.println(13);
- 4) if (ab.equals(a1)) System.out.println(14);
- 5) if (b1.equals(ab)) System.out.println(15);
- 6) if (ab.equals(b1)) System.out.println(16);
- 7) if (a1.equals(a2)) System.out.println(17);
- 8) if (b1.equals(b2)) System.out.println(18);

(6 נקודות)

ג. כתבו פעולה חיצונית בשם matchCount שמקבלת מערך של הפניות מסוג Object[].
הפעולה תחשב ותחזיר כמה מתוך העצמים במערך הם מופעים של המחלקה B,
וגם עונים על שני תנאים:
ערך התכונה val הוא גדול מ 2
ערך התכונה st שונה מהמחרוזת "B"

(2 נקודות)

ד. שינו את הפרמטר של הפעולה matchCount מהסעיף הקודם למערך מסוג A[],
האם יש צורך לשנות את מימוש הפעולה שהגדרתם בסעיף ג? נמקו תשובתכם.

(3 נקודות)

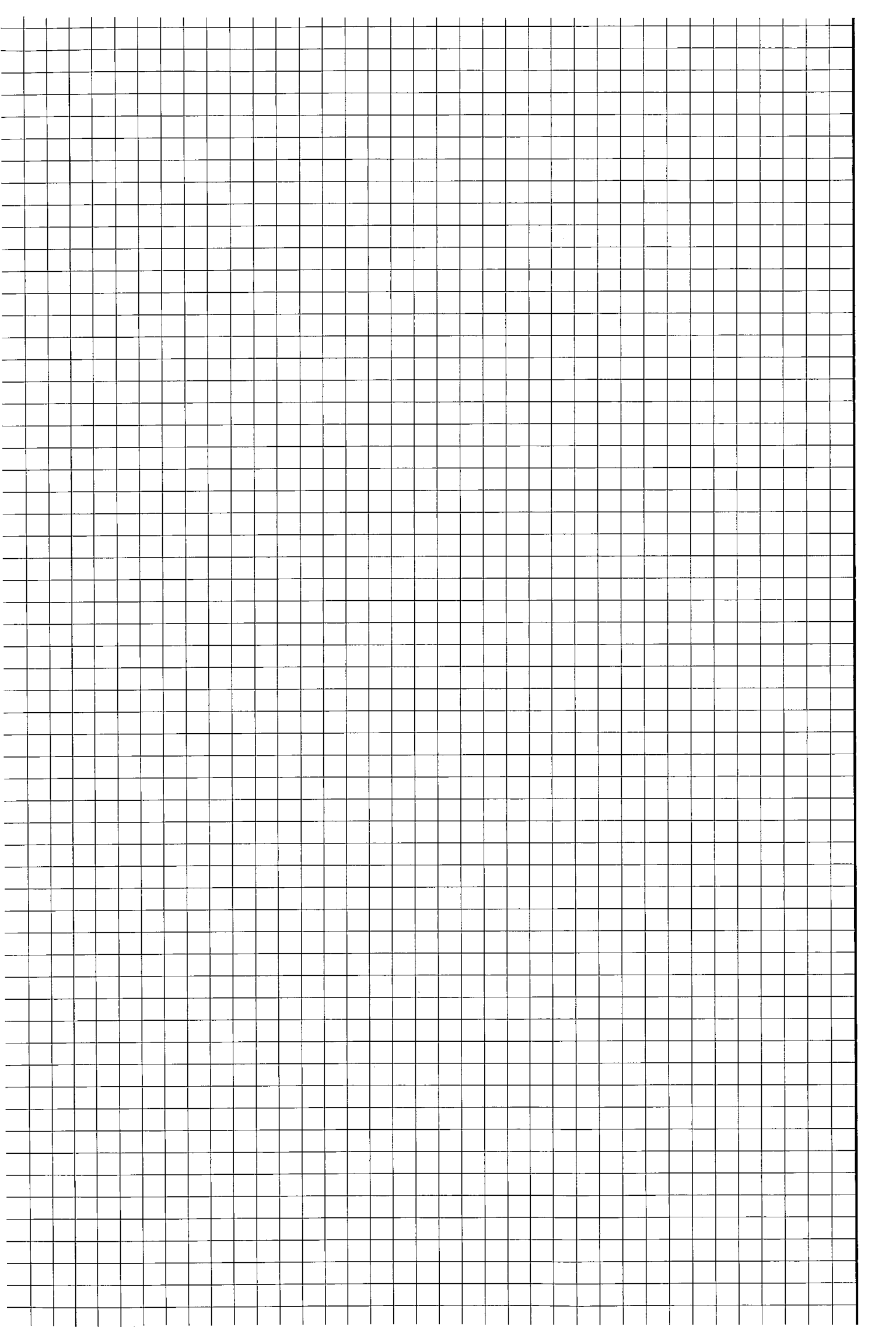
ה. האם קיים שימוש בעקרון העמסה Overloading במחלקות A ו B?
אם כן ציינו היכן והסבירו תשובתכם.

(3 נקודות)

ו. האם קיים שימוש בעקרון דריסה Overriding במחלקות A ו B?
אם כן ציינו היכן והסבירו תשובתכם.

בהצלחה (:





```
public interface Colorable {
    public void colorToFill(String color);
}
```

(19) (2)

```
public interface Rotatable {
    public void toLeft(double degree);
    public void toRight(double degree);
}
```

```
public class Figure2D {
```

```
    protected String color;
    protected Point coord;
    public Figure2D(Point p, String String color) {
```

-0.5
(2.1)

בנאים לא חובה

```
    {
        public Figure2D() {
            this.p = new P(0, 0);
            this.s = "white";
        }
    }
```

```
public class Circle extends Figure2D
    implements Colorable {
```

-0.5
(2.1)

```
    private int radius;
    private String color;
    private Point coord;
    public Circle(String color, Point p,
        int radius) { super(color, p); this.radius = radius; }
    //
```

```
    public void colorToFill(
        String color) {
        //
    }
```

לא צריך ממש חבל על הזמן שבובות

✓ public class Square extends
Figure 2D implements Colorable,
Rotatable {

~~protected int length;~~

~~private String color;~~

✓ ~~private Point coord1;~~

~~public Square(String color, Point P,
int length) {~~

~~super (color, P);~~

~~this.length = length~~

~~... }~~

~~}~~

-1
(2.1)

חסרות פעולות חובה

public class Line extends Figure 2D
implements Rotatable {

~~private Point coord2;~~

~~priv String color;~~

✓ ~~priv Point coord1;~~

~~public Line(String color, Point
coord1, Point coord2) {~~

~~super (color, coord1);~~

~~this.coord2 = coord2;~~

~~}~~

public class Point {

~~public Point~~ ~~private~~ int x;

~~priv~~ int y;

✓ public Point (int x, int y) {

this.x = x;

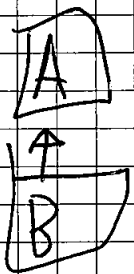
this.y = y;
}

7
(2.1)


```

2) if (ARR[i] instanceof Colorable)
    ARR[i].colorToFill("red"); -1
3) if (ARR[i] instanceof Rotatable) -1
    ARR[i].toLeft(30.0); (2.3)
4) if (ARR[i] instanceof Circle)
    svso((Circle)ARR[i]).getRadius();
5) if (ARR[i] instanceof Line)
    svso((Line)ARR[i]).coord2; -1
6) if (ARR[i] instanceof Rotatable) &&
    (ARR[i] instanceof Colorable)
    count++;
return count;
}
}

```



```

a1 (1)
a2 (3)
ab
b1 ("B", 1)
b2 ("B", 3)

```

זה לא מעקב אובייקטים כמו שלמדנו - חסר טיפוס הפניות ואובייקטים

```

1) A-object, //false.
2) "B-A", 12
3) A-object
4) 14
5) AB
6) A obj
7) "B obj"
8)

```

6.5
(2.3)

3
(3.1)

חסרים הסברים והפלט שגוי

0
(3.3)

0
(3.4)

0
(3.5)

0
(3.6)

