מבחן בתכנות מונחה עצמים - מועד נוסף

נושאים: מחלקות ואובייקטים, הורשה, פולימורפיזם ודפוסי עיצוב
משך הבחינה: 90 דקות | **סה״כ:** 100 נקודות

שם: _____ תאריך: _____ ציון: _____

הוראות: קראו היטב כל שאלה. בשאלות אמריקאיות יש לבחור תשובה אחת נכונה בלבד, אלא אם צוין אחרת. בשאלות קוד יש להתייחס לפלט, לשגיאות קומפילציה אם קיימות, ולנימוק קצר. הקוד כתוב בסגנון Java.

חלק א׳ - שאלות אמריקאיות (30 נקודות)

1. איזו פעולה מתרחשת בזמן יצירת אובייקט חדש ממחלקה? (3 נק׳)
א. נוצר מופע בזיכרון ונקרא הבנאי המתאים
ב. נוצרת מחלקה חדשה בזמן ריצה
ג. כל המתודות הופכות לסטטיות
ד. הקובץ של המחלקה נמחק ונבנה מחדש

2. מה ההבדל המרכזי בין משתנה מופע לבין משתנה סטטי? (3 נק׳)
א. משתנה מופע שייך לכל אובייקט בנפרד, ומשתנה סטטי משותף למחלקה
ב. משתנה סטטי תמיד פרטי ומשתנה מופע תמיד ציבורי
ג. משתנה מופע לא יכול להשתנות אחרי יצירת האובייקט
ד. אין הבדל, אלו שני שמות לאותו דבר

3. מהי הורשה בתכנות מונחה עצמים? (3 נק׳)
א. יכולת של מחלקה לקבל תכונות והתנהגויות ממחלקה אחרת
ב. יכולת של אובייקט להימחק אוטומטית
ג. יכולת של מתודה להחזיר כמה ערכים
ד. יכולת של מחלקה להכיל רק משתנים ללא מתודות

4. מה נכון לגבי Override? (3 נק׳)
א. מחלקת בן מממשת מחדש מתודה שירשה ממחלקת אב
ב. שתי מתודות באותה מחלקה עם אותו שם אך פרמטרים שונים
ג. יצירת אובייקט ללא בנאי
ד. המרת טיפוס פרימיטיבי לאובייקט

5. איזה עיקרון מאפשר לקרוא לאותה מתודה על אובייקטים מטיפוסים שונים ולקבל התנהגות שונה? (3 נק׳)

א. פולימורפיזם

ב. קומפילציה

ג. אינקפסולציה בלבד

ד. סידור מערכים

6. מה תפקידו של Interface בדרך כלל? (3 נק׳)

א. להגדיר חוזה של פעולות שמחלקות צריכות לממש

ב. לשמור נתונים במסד נתונים

ג. להחליף את כל הבנאים במחלקה

ד. למנוע הורשה בין מחלקות

7. באיזה מצב סביר להשתמש ב-Singleton? (3 נק׳)

א. כאשר רוצים מופע יחיד ונקודת גישה מרכזית אליו

ב. כאשר רוצים ליצור המון אובייקטים אקראיים בלי שליטה

ג. כאשר אין צורך בבנאים בכלל

ד. כאשר כל מחלקה חייבת לרשת משתי מחלקות

8. מה הרעיון המרכזי של Factory Method? (3 נק׳)

א. להפריד את תהליך יצירת האובייקטים מהקוד שמשתמש בהם

ב. להעתיק אובייקט קיים בדיוק

ג. לעדכן מאזינים כאשר מצב משתנה

ד. למנוע שימוש בפולימורפיזם

9. מה מאפיין את Observer? (3 נק׳)

א. אובייקט אחד מודיע לכמה אובייקטים אחרים כאשר יש שינוי

ב. מחלקה אחת יוצרת לעצמה מופע יחיד בלבד

ג. אובייקט מחליף אלגוריתם בזמן ריצה

ד. מחלקת בן חוסמת את כל המתודות של האב

10. איזה דפוס מתאים כאשר רוצים להחליף אלגוריתם חישוב בלי לשנות את המחלקה שמשתמשת בו? (3 נק׳)

א. Strategy

ב. Singleton

ג. Iterator

ד. Finalizer

חלק ב׳ - נכון / לא נכון (16 נקודות)

מס׳	טענה	נכון	לא נכון
1		☐	☐

	כל אובייקט שנוצר מאותה מחלקה חייב להחזיק בדיוק אותם ערכים בכל השדות שלו.		
2	Encapsulation עוזר להסתיר את המימוש הפנימי ולחשוף גישה מבוקרת דרך מתודות.	☐	☐
3	מחלקת בן יכולה להוסיף שדות ומתודות מעבר למה שקיבלה ממחלקת האב.	☐	☐
4	Overloading ו-Overriding הם אותו מנגנון בדיוק.	☐	☐
5	אפשר להשתמש בהפניה מטיפוס אב כדי להחזיק אובייקט ממחלקת בן.	☐	☐
6	אם מתודה מוגדרת כ-private במחלקת אב, מחלקת בן יכולה לבצע לה Override רגיל.	☐	☐
7	Factory יכול לעזור כאשר סוג האובייקט שנוצר תלוי בקלט או בתנאי מסוים.	☐	☐
8	ב-Observer, האובייקטים המאזינים חייבים לדעת את כל המימוש הפנימי של האובייקט שעליו הם מאזינים.	☐	☐

חלק ג׳ - שאלות קוד וניתוח (32 נקודות)

1. שאלת קוד 1 - מחלקות ואובייקטים (8 נק׳)

```
class Student {
    private String name;
    private int grade;

    public Student(String name, int grade) {
        this.name = name;
        this.grade = grade;
  }

    public void bonus() {
        grade += 5;
  }

    public String toString() {
        return name + ":" + grade;
  }
}

public class Main {
    public static void main(String[] args) {
        Student s1 = new Student("Noa", 80);
        Student s2 = new Student("Noa", 80);
        s1.bonus();
        System.out.println(s1);
        System.out.println(s2);
  }
}
```

מה יודפס? הסבירו בקצרה מדוע שני האובייקטים אינם משתנים יחד.

_____

_____

_____

2. שאלת קוד 2 - הורשה ו-Override (8 נק׳)

```
class Animal {
    public void speak() {
        System.out.println("Animal sound");
  }
}

class Cat extends Animal {
    public void speak() {
        System.out.println("Meow");
  }

    public void scratch() {
        System.out.println("Scratch");
  }
}

public class Main {
    public static void main(String[] args) {
        Animal a = new Cat();
        a.speak();
        // a.scratch();
  }
}
```

א. מה יודפס? ב. האם השורה שמסומנת כהערה יכולה לעבוד אם מסירים את ההערה? נמקו.

_____

_____

_____

3. שאלת קוד 3 - Overloading מול Overriding (8 נק׳)

```
class Printer {
    public void print(int x) {
        System.out.println("int " + x);
  }

    public void print(String x) {
        System.out.println("String " + x);
  }
}

public class Main {
    public static void main(String[] args) {
        Printer p = new Printer();
        p.print(7);
        p.print("7");
  }
}
```

איזה מנגנון מופיע כאן? מה יודפס?

_____

_____

_____

4. שאלת קוד 4 - פולימורפיזם במערך (8 נק׳)

```
abstract class Shape {
    public abstract double area();
}

class Square extends Shape {
    private double side;
    public Square(double side) { this.side = side; }
    public double area() { return side * side; }
}

class Circle extends Shape {
    private double r;
    public Circle(double r) { this.r = r; }
    public double area() { return 3.14 * r * r; }
}

public class Main {
    public static void main(String[] args) {
        Shape[] shapes = { new Square(4), new Circle(2) };
        for (Shape s : shapes) {
            System.out.println(s.area());
    }
  }
}
```

מה יודפס? איזה עיקרון OOP מאפשר את הקריאה ל-area על שני סוגי אובייקטים?

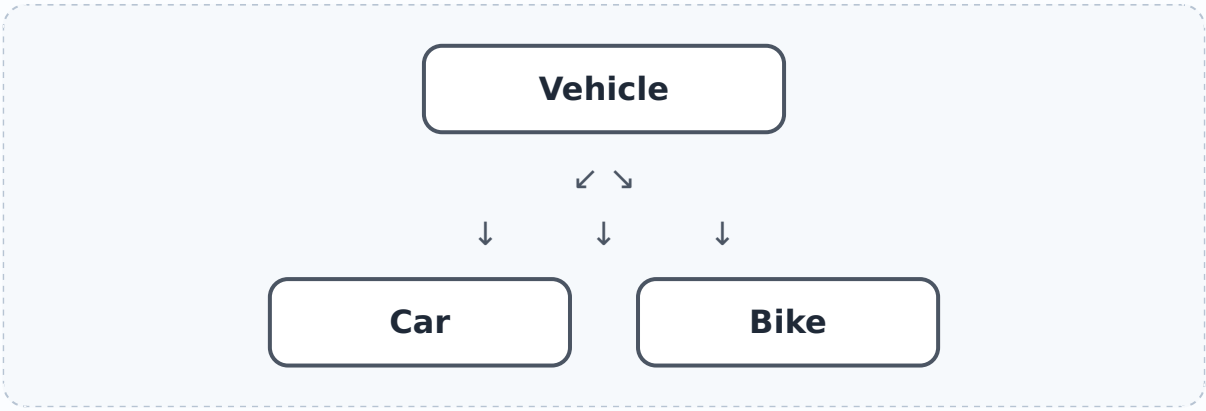
_____

_____

_____

חלק ד׳ - שאלות ויזואליות / איורים (22 נקודות)

1. שאלה ויזואלית 1 - זיהוי קשר בין מחלקות (5 נק׳)



בתרשים מוצגת מחלקה כללית שממנה יוצאות שתי מחלקות ספציפיות. איזה קשר OOP מתואר כאן? כתבו גם דוגמה קצרה מהעולם האמיתי.

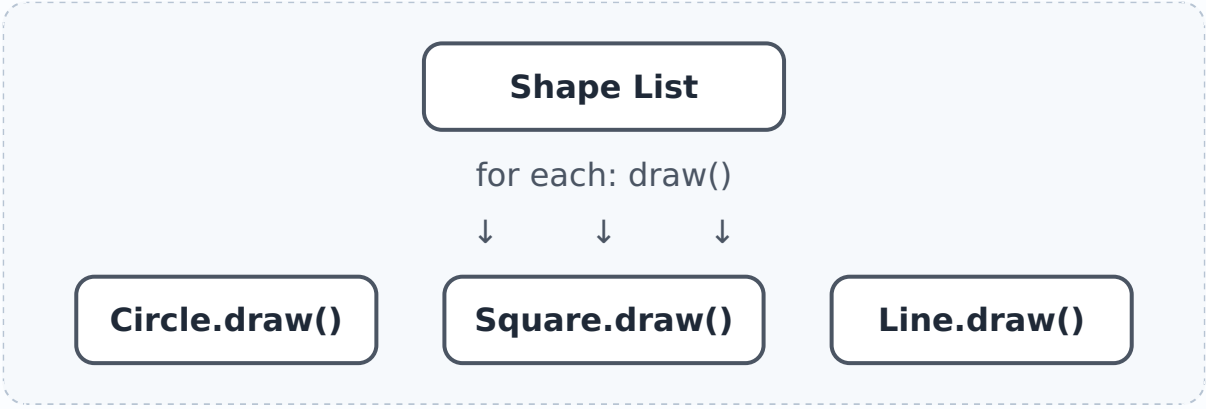
---

2. שאלה ויזואלית 2 - זיהוי בעיית תכנון (5 נק׳)

הלקוח יוצר ישירות כמה סוגי אובייקטים בעזרת new בתוך אותו קטע קוד. איזה דפוס עיצוב עשוי לשפר את התכנון? הסבירו למה.

---

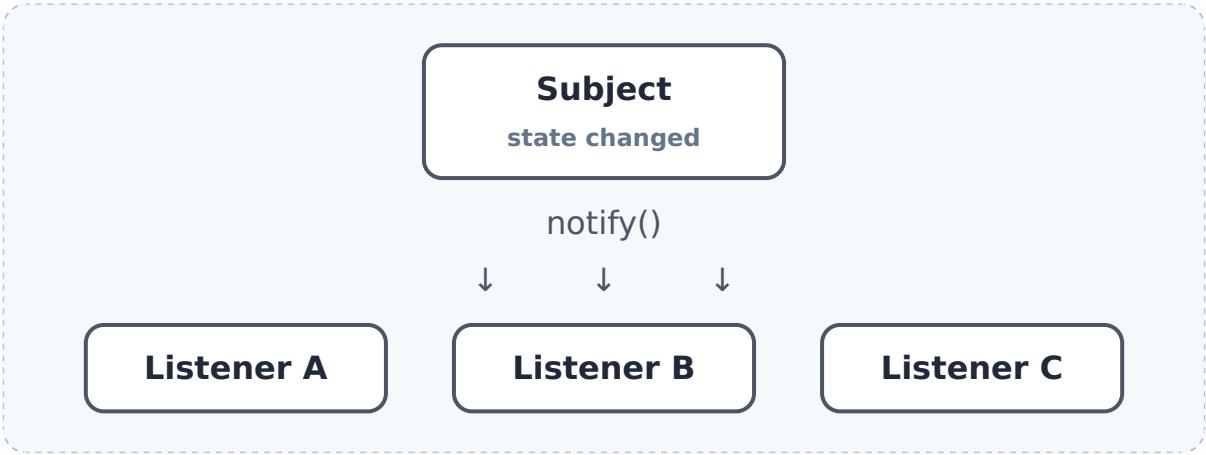
3. שאלה ויזואלית 3 - פולימורפיזם (5 נק׳)



בתרשים יש רשימת אובייקטים שונים שמקבלים את אותה הודעה draw(). מהו העיקרון שמודגם?

---

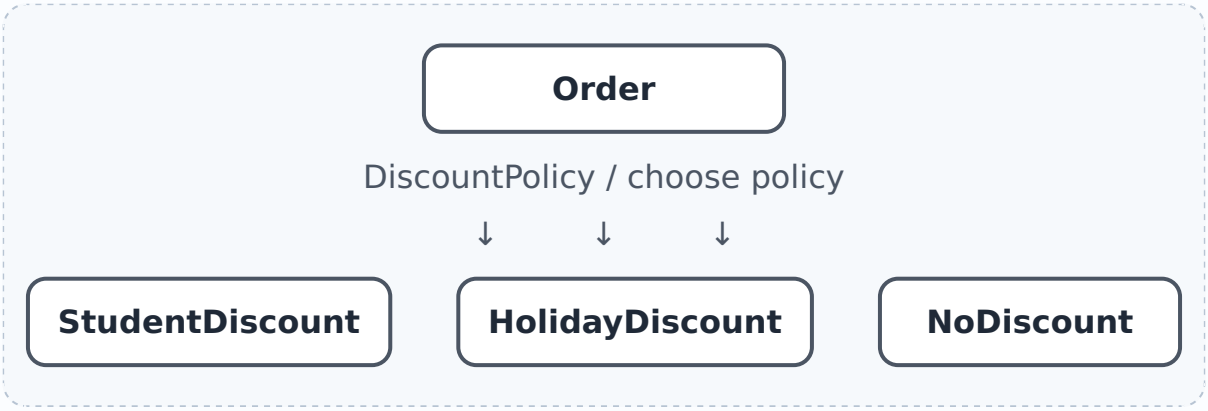
4. שאלה ויזואלית 4 - Observer (3.5 נק׳)



בתרשים יש אובייקט מרכזי שמחזיק רשימת מאזינים ומודיע להם על שינוי. מה קורה כאשר המצב משתנה?

---

5. שאלה ויזואלית 5 - Strategy (3.5 נק׳)



בתרשים אובייקט Order משתמש באחד מתוך כמה סוגי חישוב הנחה. איזה רעיון תכנוני מוצג כאן?

נספח תשובות מקוצר - לבדיקה עצמית

הנספח מיועד לבדיקה אחרי פתרון המבחן. בשאלות פתוחות ייתכנו ניסוחים נוספים שיקבלו ניקוד מלא אם הנימוק נכון.

1. א
2. א
3. א
4. א
5. א
6. א
7. א
8. א
9. א
10. א

נכון/לא נכון 1. לא נכון

נכון/לא נכון 2. נכון

נכון/לא נכון 3. נכון

נכון/לא נכון 4. לא נכון

נכון/לא נכון 5. נכון

נכון/לא נכון 6. לא נכון

נכון/לא נכון 7. נכון

נכון/לא נכון 8. לא נכון

קוד 1. יודפס Noa:85 ואז Noa:80. אלו שני אובייקטים שונים בזיכרון.

קוד 2. יודפס Meow. הקריאה scratch לא תתקמפל דרך הפניה מטיפוס Animal ללא המרה מתאימה.

קוד 3. Overloading. יודפס int 7 ואז String 7.

קוד 4. יודפס 16.0 ואז 12.56. העיקרון הוא פולימורפיזם עם מחלקה אבסטרקטית.

ויזואלי 1. הורשה / is-a.

ויזואלי 2. Factory Method או Simple Factory, כי הוא מרכז את יצירת האובייקטים.

ויזואלי 3. פולימורפיזם.

ויזואלי 4. כל המאזינים הרשומים מקבלים עדכון.

ויזואלי 5. Strategy - החלפת אלגוריתם ללא שינוי המחלקה הראשית.

בהצלחה ❤️