

۱. در مورد اصطلاحات و مفاهیم زیر توضیحات کافی ارائه دهید:

الف) RIPR Model	د) Metamorphic Testing
ب) Testability	ه) Class Integration Test Order (CITO)
ج) Test Case	و) Mutation Testing

۲. کد<sup>۱</sup> زیر را در نظر بگیرید و به سوالات پاسخ دهید. (توضیح مختصر فراموش نشود!)

```
class Point {
    private int x; private int y;

    public Point (int x, int y) { this.x=x; this.y=y; }
    @Override public boolean equals (Object o)
    {
        // Location A
        if (!(o instanceof Point)) return false;
        Point p = (Point) o;
        return (p.x == this.x) && (p.y == this.y);
    }
}

class ColorPoint extends Point
{
    private Color color;
    // Fault: Superclass instantiable; subclass state extended
    public ColorPoint (int x, int y, Color color)
    {
        super (x,y);
        this.color=color;
    }
}

@Override public boolean equals (Object o)
{
    // Location B
    if (!(o instanceof ColorPoint)) return false;
    ColorPoint cp = (ColorPoint) o;
    return (super.equals(cp) && (cp.color == this.color));
}

// Tests:
Point p = new Point (1,2);
ColorPoint cp1
ColorPoint cp2
p.equals (cp1);
cp1.equals (p);
cp1.equals (cp2); // Test 3: Result = false;
// Expected: p.equals (cp1) = true; cp1.equals (p) = true,
//             cp1.equals (cp2) = false
```

الف) برنامه داده شده چه مشکلی دارد؟ عیب برنامه را دقیقاً و با پیشنهاد یک تغییر در کد مشخص کنید.  
 ب) اگر امکان دارد، آزمونی تعریف کنید که عیب را اجرا نکند. اگر ممکن نیست، توضیح دهید.

<sup>۱</sup> یکی از اشتباهات OO، از کتاب Effective Java, Second Edition

- ج) اگر امکان دارد، آزمونی تعریف کنید که عیب را اجرا کند، اما منجر به خطا نشود. اگر ممکن نیست، توضیح دهید.
- د) اگر امکان دارد، آزمونی تعریف کنید که منجر به خطا شود، اما شکست نه. اگر ممکن نیست، توضیح دهید.
- ه) در کد بالا، اولین حالت خطا را مشخص کنید. حالت برنامه را به صورت کامل مشخص کنید.

۳. تفاوت میان طراحی آزمون مدل رانه<sup>۲</sup> و آزمون مبتنی بر مدل<sup>۳</sup> در چیست؟

۴. برای مقایسه معیارهای پوشش (با یکدیگر!) چه روش ها و معیارهایی را پیشنهاد می کنید؟

۵. متد `search()` را در نظر بگیرید.

```
public static int search (List list, Object element)
// Effects: if list or element is null throw NullPointerException
//     else if element is in the list, return an index
//     of element in the list; else return -1
//     for example, search ([3,3,1], 3) = either 0 or 1
//     search ([1,7,5], 2) = -1
```

- الف) یک مدل دامنه ورودی<sup>۴</sup>، شامل متغیرهای ورودی و مشخصات آنها، ویژگی (های) مورد نظر برای افراز، و بلاک، ارائه کنید.
- ب) ویژگی های یک افراز مناسب را برای افراز ارائه شده، بررسی کنید.
- ج) یکی از روش های ترکیب بلاک را انتخاب کنید و یک مجموعه آزمون (شامل داده آزمون) متناسب با آن بنویسید.

۶. متد `equals()` از رده `java.util.AbstractList<E>` را در نظر بگیرید.

```
public boolean equals (Object o)
{
    if (o == this) // A
        return true;
    if (!(o instanceof List)) // B
        return false;
    ListIterator<E> e1 = listIterator();
    ListIterator<?> e2 = ((List) o).listIterator();
    while (e1.hasNext() && e2.hasNext()) // C
    {
        E o1 = e1.next();
        Object o2 = e2.next();
        if (!(o1 == null ? o2 == null : o1.equals (o2))) // D
            return false;
    }

    return !(e1.hasNext() || e2.hasNext()); // E
}
```

- الف) گراف جریان کنترلی<sup>۵</sup> متد را بکشید. گره ها و یال ها را با مقادیر مناسب برچسب گذاری کنید.
- ب) حداقل به چند آزمون برای پوشش گره<sup>۶</sup> نیاز هست؟ چرا؟
- ج) نیازمندی های آزمون و مقادیر آزمون برای پوشش جفت-یال<sup>۷</sup> را مشخص کنید. (آزمون ها را به صورت فراخوانی متد بنویسید)

۷. رده `GoodFastCheap`، در آدرس زیر را در نظر بگیرید.

<https://cs.gmu.edu/~offutt/softwaretest/java/GoodFastCheap.java>

الف) مجموعه آزمون هایی که `GACC` و `RACC` را برای گزاره متد `isSatisfactory()`، برآورده می کند، مشخص کنید.

<sup>۲</sup> Model-Driven Test Design  
<sup>۳</sup> Model-based Testing  
<sup>۴</sup> Input Domain Model (IDM)  
<sup>۵</sup> Control Flow Graph  
<sup>۶</sup> Node Coverage  
<sup>۷</sup> Edge-pair Coverage

ب) فرض کنید متد `issatisfactory()` به صورت زیر بازآرایی شده باشد.

```
public boolean issatisfactory() { // Refactoring here
    if (good && fast) return true;
    if (good && cheap) return true;
    if (fast && cheap) return true;

    return false;
}
```

آیا آزمون های قسمت قبل، RACC را بر روی متد بازآرایی شده برآورده می کند؟ چه مواردی کم هستند؟

۸. (امتیازی) TDD و BDD چیست؟ چه نسبتی با یکدیگر دارند؟

موفق باشید