

## בחינה בתוכנה 1

סמסטר א' תשע"ח, מועד ב', 16 בספטמבר 2019  
לנה דנקין, ברית יונגמן, שי גרשטיין

משך הבחינה שלוש שעות.

סך הניקוד על השאלות בבחינה הוא 105, אך הציון המקסימלי אותו ניתן לקבל הוא 100.

יש להניח, אלא אם צויין אחרת, כי:

- הקוד שמופיע במבחן מתאים לגירסא Java8.
- כל החבילות הדרושות יובאו, ואין צורך לכתוב שורות import בגוף הקוד.
- כל מחלקה שהיא public מופיעה בקובץ Java משלה.
- בכל שאלה, כל המחלקות מופיעות באותה חבילה (package).
- בזמן הבחינה, אתם נדרשים לזהות שגיאות קומפילציה שנוצרות כתוצאה מהפרת עקרונות Java-יים ושימוש לא נכון במחלקות/פונקציות. במידה וישנה טעות הקלדה (סוגר חסר, שימוש באות גדולה שלא לצורך וכו') אין לראות בסיבות אלה גורמים לשגיאות קומפילציה.

בבחינה זו מופיע קוד שבחלקו אינו מתקמפל, אינו רץ או שנוגד את הסטנדרטים של Java כפי שנלמדו בקורס, וזאת מתוך מטרה לבחון ידע והבנה של נושאים מסוימים. אין לראות בקטעי קוד אלה דוגמא לכתיבה נכונה ב Java.

יש לסמן את התשובה הטובה ביותר בתשובון. לא יינתן ניקוד על סימון תשובה בטופס הבחינה, במחברת הבחינה, או בטופס הנימוקים.

יש לנמק בתמצות את כל התשובות על גבי טופס הבחינה במקומות המסומנים. נימוק חסר או לא נכון עלול לגרום לאי קבלת נקודות על שאלה במקרים שבהם יוחלט לקבל יותר מתשובה אחת. נימוק לתשובה אחרת מזו שסומנה על גבי טופס התשובות לא יתקבל.

יש לצרף את טופס המבחן למחברת הבחינה. מחברת ללא טופס עזר תפסל. סימוני תשובות במחברת הבחינה לא יבדקו.

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.

לטופס הבחינה מצורף דף לשאלות הסטודנטים. יש לכתוב שאלות שמתעוררות במהלך הבחינה בדף זה ולהעביר לסגל הקורס. שאלות ענייניות תענינה על ידי סגל הקורס בפני כל הנבחנות/ים.

© כל הזכויות שמורות למחברים. מבלי לפגוע באמור לעיל, אין להעתיק, לצלם, להקליט, לשדר, לאחסן במאגר מידע, בכל דרך שהיא, בין מכונית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה.

בהצלחה!

1. שאלה 1:

מבין הטענות הבאות, איזו טענה לא נכונה?  
 אם כל הטענות נכונות – בחר/י בתשובה "כל הטענות נכונות".  
 נימוק:

- א. הטיפוס הדינאמי של אובייקט נקבע ע"י הבנאי שבו השתמשו ליצירת האובייקט.
- ב. כל אובייקט ב Java נוצר על ה heap.
- ג. **אתחול משתנה ל null משאיר את המשתנה לא מאותחל, והקומפיילר מונע שימוש במשתנה זה עד לאתחול ערך תקין.**
- ד. ב Java יש 8 טיפוסים פרימיטיביים, ושאר הטיפוסים הם אובייקטים.
- ה. לכל טיפוס פרימיטיבי קיים אובייקט מעטפת (Wrapper).
- ו. כל הטענות נכונות.

2. שאלה 2:

להלן מספר טענות על מחלקות אבסטרקטיות. בחר/י בתשובה הטובה ביותר:  
 נימוק:

- א. מחלקה אבסטרקטית היא סוג של ממשק.
- ב. במחלקה אבסטרקטית לא ניתן להגדיר שירותים בניראות private.
- ג. מחלקה אבסטרקטית יכולה לרשת רק ממחלקה אבסטרקטית אחרת.
- ד. במחלקה אבסטרקטית לא ניתן להגדיר שירות שהוא final.
- ה. במחלקה אבסטרקטית לא ניתן להגדיר שדות בניראות private.
- ו. **מלבד תשובה זו, כל התשובות לא נכונות.**

3. שאלה 3:

התבונן/י בקוד המצורף ובחרו בתשובה הנכונה:

```
public class Base {
    public Base(int i) { }
    public Base(String s) { }
}

public class Sub extends Base{           //in q4 - public class Sub{
    public Sub() { }                       /**
    public Sub(int i, int j) { super(i+1); } /**
    public Sub(int i) { this(i,i); }       /****
}

public class Test{
    public static void main(String[] args) {
        Base b1 = new Base(123);
        Base b2 = new Sub(1,2);           //****
    }
}
```

נימוק:

- א. הקוד מתמקפל ורץ.
- ב. קיימת שגיאת קומפילציה בשורה המסומנת ב \*
- ג. קיימת שגיאת קומפילציה בשורה המסומנת ב \*\*
- ד. קיימת שגיאת קומפילציה בשורה המסומנת ב \*\*\*
- ה. קיימת שגיאת קומפילציה בשורה המסומנת ב \*\*\*\*
- ו. ישנה שגיאת קומפילציה בשתי שורות שונות.
- ז. ישנה שגיאת קומפילציה ביותר משתי שורות.

4. שאלה 4:

מה תהיה התשובה במידה ונשנה את הקוד של שאלה 3 כך ש Sub לא תירש מ Base?

נימוק:

- א. הקוד מתמקפל ורץ.
- ב. קיימת שגיאת קומפילציה בשורה המסומנת ב \*
- ג. קיימת שגיאת קומפילציה בשורה המסומנת ב \*\*
- ד. קיימת שגיאת קומפילציה בשורה המסומנת ב \*\*\*
- ה. קיימת שגיאת קומפילציה בשורה המסומנת ב \*\*\*\*
- ו. ישנה שגיאת קומפילציה בשתי שורות שונות.
- ז. ישנה שגיאת קומפילציה ביותר משתי שורות.

5. שאלה 5:

להלן מספר טענות כלליות הנוגעות לשפת ה Java. בחר/י בתשובה הטובה ביותר.

נימוק:

- א. קוד שהתקמפל בגירסא 1.6 של Java יוכל לרוץ על גירסא 1.7.
- ב. המהדר (קומפיילר) הוא חלק מההתקנה של ה JRE.
- ג. לאחר תהליך ההידור (קומפילציה) נוצרים קבצי java.
- ד. ניתן להריץ תוכנית java שעברה הידור (קומפילציה) על כל מחשב, ללא התקנות הקשורות ל java.
- ה. ה garbage collector לא מאפשר מצב שבו נגמר הזכרון של תוכנית Java.
- ו. מלבד תשובה זו יש יותר מתשובה נכונה אחת.

## 6. שאלה 6:

להלן מספר טענות כלליות הקשורות למילה השמורה static. בחר/י בתשובה הטובה ביותר:

נימוק:

- א. לא ניתן להגדיר שירות סטטי שהוא בניראות private.
- ב. לא ניתן להגדיר שירות סטטי אבסטרקטי.
- ג. לא ניתן לגשת לשדה סטטי מתוך שירות מופע.
- ד. שדות סטטיים נוצרים בזכרון ברגע שנוצר המופע הראשון של המחלקה בה הם מוגדרים.
- ה. מלבד תשובה זו, כל התשובות לא נכונות.
- ו. מלבד תשובה זו, יש יותר מתשובה נכונה אחת.

7. שאלה 7:

```
public class A {
    public void func(Object o) {
        System.out.println("A's func(Obj)");
    }
}

public class B extends A{
    public void func(String s) {
        System.out.println("B's func(Str)");
    }
    public void func(Object s) {
        System.out.println("B's func(Obj)");
    }
}

public class Test{
    public static void main(String[] args) {
        A a = new B();
        a.func("abc");
    }
}
```

מה יקרה בהרצת התוכנית Test?

נימוק:

- א. הקוד לא ירוץ כיוון שהקוד של Test לא מתקמפל.
- ב. הקוד לא ירוץ כיוון שהקוד של B לא מתקמפל.
- ג. הקוד ירוץ ויודפס A's func(Obj)
- ד. **הקוד ירוץ ויודפס B's func(Obj)**
- ה. הקוד ירוץ ויודפס B's func(Str)
- ו. הקוד יתקמפל ובזמן ריצה נקבל שגיאת ClassCastException.

8. שאלה 8:

לפניכם הקוד של המחלקה Test:

```
public class Test{
    public static void main(String[] args) {
        List<String> strings = new ArrayList<>();
        List<Integer> ints = new ArrayList<>();
        List<Object> objs = new ArrayList<>();
        A.func(strings);
        A.func(ints);
        //A.func(objs); //should not compile
    }
}
```

נרצה לממש המחלקה A כך ש Test (וגם מחלקה A, כמובן) תתקמפל ותרוץ, ואם נוציא את השורה האחרונה של התוכנית מהעברה, הקוד יפסיק להתקמפל? לפניכם מספר מימושים:

```
public class A{ //v1
    public static void func(List<?> lst) {}
}

public class A{ //v2
    public static void func(List<? extends Comparable<?>> lst) {}
}

public class A{ //v3
    public static void func(List<String> lst) {}
    public static void func(List<Integer> lst) {}
}
```

בחר/י בתשובה הטובה ביותר:  
נימוק:

- א. כל הגרסאות של A מקיימות את התנאים.
- ב. כל הגרסאות של A לא מקיימות את התנאים.
- ג. רק גרסא v1 מקיימת את התנאים.
- ד. רק גרסא v2 מקיימת את התנאים.
- ה. רק גרסא v3 מקיימת את התנאים.
- ו. רק גרסאות v1+v2 מקיימות את התנאים.
- ז. רק גרסאות v2+v3 מקיימות את התנאים.
- ח. רק גרסאות v1+v3 מקיימות את התנאים.

9. שאלה 9:

```
public class A{
    public static int[] func(int[] arr) {
        arr = new int[] {4,5,6};
        return arr;
    }
    public static void main(String[] args) {
        int[] arr1 = new int[3];
        int[] arr2 = {1,2,3};
        int[] arr3 = {1,2,3};
        System.out.print(arr1[0] + " ");    /*
        System.out.print((arr2 == arr3) + " ");
        arr3 = func(arr2);
        System.out.print((arr2[0] == arr3[0]) + " ");
    }
}
```

מה יקרה בהרצת התוכנית הבא? בחר/י בתשובה הטובה ביותר:  
נימוק:

- א. התוכנית תזרוק שגיאת זמן ריצה בשורה המסומנת ב \*
- ב. התוכנית תדפיס true false 0
- ג. התוכנית תדפיס null false false
- ד. התוכנית תדפיס null true false
- ה. התוכנית תדפיס 0 false true
- ו. התוכנית תדפיס null false true
- ז. התוכנית תדפיס 0 false false
- ח. התוכנית תדפיס null true true



10. שאלה 10:

```

//@imp_inv: innerMap.keySet().size() == keySet().size()
public class MyMultiMap<K, V> {
    private HashMap<K, List<V>> innerMap = new HashMap<>();

    public void put(K key, V value) {
        List<V> lst = innerMap.get(key);
        if (lst == null) {
            lst = new ArrayList<>();
            innerMap.put(key, lst);
        }
        lst.add(value)
    }

    public Iterator<V> getAll(K key){
        return innerMap.get(key).iterator();
    }

    public V get(K key) {
        if (innerMap.containsKey(key)) {
            return innerMap.get(key).get(0);
        }
        return null;
    }

    public Set<K> keySet(){ return innerMap.keySet(); }

    public void removeKey(K key) { innerMap.remove(key); }
}

```

לפניכם מימוש המחלקה MyMultiMap שאמורה לייצר מפה אשר יכולה להכיל יותר מערך אחד עבור אותו המפתח. מעצבת המחלקה ביקשה לממש את התכונות הבאות:

- תכונה 1: השירות getAll יחיד איטרטור על מנת שלא להפטר למשתמשת שירות זה גישה לערכים השמורים עבור מפתח כלשהו במבנה innerMap. לא מאפשר קריאה של הערכים המתאימים ל key ב innerMap
- תכונה 2: המבנה הפנימי שבו נשמר המידע, ובפרט גם המבנה הפנימי שבו נשמרים הערכים עבור אותו המפתח לא חשוף למשתמשת.
- תכונה 3: במידה והמשתמשת תכניס ערך יחיד עבור כל מפתח, השירותים get, keySet, removeKey יתנהגו כמו השירותים המקבילים ב HashMap.

לנוחותכם, מצורף התיעוד של השירותים get ו remove של המחלקה Map.

|          |  |
|----------|--|
| <u>V</u> | <b>get(Object key)</b><br>Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key. |
| <u>V</u> | <b>remove (Object key)</b><br>Removes the mapping for a key from this map if it is present (optional operation).                       |

בחר/י בתשובה הטובה ביותר:  
נימוק:

- א. רק תכונה 1 נשמרת במימוש הקיים.
- ב. רק תכונה 2 נשמרת במימוש הקיים.
- ג. רק תכונה 3 נשמרת במימוש הקיים.
- ד. רק תכונות 1+2 נשמרות במימוש הקיים.
- ה. רק תכונות 1+3 נשמרות במימוש הקיים.
- ו. רק תכונות 2+3 נשמרות במימוש הקיים.
- ז. כל התכונות נשמרות במימוש הקיים.
- ח. כל התכונות לא נשמרות במימוש הקיים.

11. שאלה 11:

לפניכם שלוש טענות על המחלקה MyMultiMap שהוצגה בשאלה 9.

טענה 1: לפחות אחד משירותי המחלקה יכול לזרוק `NullPointerException`.

טענה 2: אינווריאנטת היצוג (implementation invariant) היא תקינה ונכונה.

טענה 3: אם נוסיף לשירות `get` את תנאי ה `pre` הבא: `pre: key != null`, אופן השימוש במחלקה ע"י הלקוחה לא ישתנה.

בחר/י בתשובה הטובה ביותר:

נימוק:

- א. כל הטענות נכונות.
- ב. כל הטענות לא נכונות.
- ג. רק טענה 1 נכונה.
- ד. רק טענה 2 נכונה.
- ה. רק טענה 3 נכונה.
- ו. רק טענות 1+2 נכונות.
- ז. רק טענות 1+3 נכונות.
- ח. רק טענות 2+3 נכונות.

12. שאלה 12:

נרצה לממש את המחלקה MyIntegersMultiMap – שלה אותה ההתנהגות כמו ל MyMultiMap, אך ידוע שכל הערכים שלה הם מטיפוס Integer, ולכן יש לה רק פרמטר גנרי אחד – הטיפוס של המפתחות. לפניכם מספר מימושים אפשריים:

```
//v1
public class MyIntegersMultiMap<K, Integer>
    extends MyMultiMap<K, Integer> { }

//v2
public class MyIntegersMultiMap<K>
    extends MyMultiMap <K, Integer>{ }

//v3
public class MyIntegersMultiMap<K, V>
    extends MyMultiMap <K, V>{ }

//v4
public class MyIntegersMultiMap<K, V>
    extends MyMultiMap <K, Integer>{ }
```

מבין המימושים המוצעים, איזה מימוש מתאים? בחר/י בתשובה הטובה ביותר:  
נימוק:

- א. רק מימוש 1.
- ב. רק מימוש 2.
- ג. רק מימוש 3.
- ד. רק מימוש 4.
- ה. כל המימושים לא מתאימים.
- ו. יש יותר ממימוש מתאים אחד.

13. שאלה 13:

```
public class A{
    private int i;
    private int j;

    public A(int i, int j) {
        this.i = i;
        this.j = j;
    }

    public int hashCode() {
        return new Random().nextInt(3); /* random int values from
{0,1,2}*/
    }

    public boolean equals(Object obj) {
        A other = (A)obj;
        return this.i == other.i || this.j == other.j
            || this.i == other.j || this.j == other.i;
    }

    public String toString() {
        return "<" + this.i + "," + this.j + ">";
    }

    public static void main(String[] args) {
        A a1 = new A(1,2);
        A a2 = new A(2,1);
        A a3 = new A(1,3);
        A a4 = new A(5,2);
        List<A> aList = Arrays.asList(a1, a2, a3, a4);
        aList.sort((x,y) -> {
            int a = Integer.compare(x.i, y.i);
            if (a != 0) {return a;}
            return Integer.compare(y.j, x.j);
        });
        System.out.println(aList);
    }
}
```

מה יודפס בהרצת התוכנית?  
נימוק:

- א. [**<1,3>, <1,2>, <2,1>, <5,2>**]
- ב. [**<1,2>, <1,3>, <2,1>, <5,2>**]
- ג. [**<5,2>, <2,1>, <1,2>, <1,3>**]
- ד. [**<5,2>, <2,1>, <1,3>, <1,2>**]

- ה. יודפסו בוודאות ארבעה איברים, אך סדר ההדפסה יכול להשתנות מריצה לריצה.
- ו. יודפסו בוודאות שלושה איברים, אך בחירת האיברים והסדר יכולים להשתנות מריצה לריצה.
- ז. יודפסו בוודאות שני איברים, אך בחירת האיברים והסדר יכולים להשתנות מריצה לריצה.

14. שאלה 14:

לפניכם התוכנית Test המשתמשת במחלקה A אשר מומשה בשאלה קודמת.

```
public class Test{
    public static void main(String[] args) {
        A a1 = new A(1,2);
        A a2 = new A(2,1);
        A a3 = new A(1,3);
        A a4 = new A(5,2);
        A a5 = new A(3,1);
        Set<A> aSet = new HashSet<>();
        aSet.add(a1); aSet.add(a2); aSet.add(a3);
        aSet.add(a4); aSet.add(a5);
        System.out.println(aSet);
    }
}
```

מה ניתן לומר בוודאות על פלט התוכנית?  
נימוק:

- א. לא יתכן שיודפסו רק שני איברים.
- ב. לא יתכן שיודפסו שלושה איברים.
- ג. לא יתכן שיודפסו ארבעה איברים.
- ד. לא יתכן שיודפסו חמישה איברים.
- ה. כל התשובות לא נכונות.
- ו. מלבד תשובה זו יש יותר מתשובה נכונה אחת.

15. שאלה 15:

```
public class E {
    public static void main(String[] args) {
        String[] a = { "A", "B", null, "C" };
        try {
            for (String s : a) {
                System.out.print(s.toString());
            }
        } catch (Exception e) {
            System.out.print("$");
        } finally {
            System.out.print("#");
        }
    }
}
```

מה יקרה בהרצת הקוד הבא?  
נימוק:

- א. התוכנית תדפיס A##B ותעוף על שגיאה בזמן ריצה.
- ב. התוכנית תדפיס A#B# ותעוף על שגיאה בזמן ריצה.
- ג. התוכנית תדפיס AB\$# ותסתיים ללא שגיאה.
- ד. התוכנית תדפיס AB# ותעוף על שגיאה בזמן ריצה.
- ה. התוכנית תדפיס AB\$#C ותסתיים ללא שגיאה.
- ו. התוכנית תדפיס AB\$#C# ותסתיים ללא שגיאה.
- ז. התוכנית תדפיס A##B ותסתיים ללא שגיאה.
- ח. התוכנית תדפיס AB# ותסתיים ללא שגיאה.

## 16. שאלה 16:

שאלה זו מתייחסת למימושן של מחלקות A, B שמימושן אינו נתון. ניתן להניח את קיומן של מחלקות נוספות ללא שום מגבלות.

```
public void func(A a, B b) {
    B b2 = (B) a; // line 1
    A a2 = (A) b; // line 2
    A a3 = b;     // line 3
}
```

להלן מספר טענות לגבי קוד זה:

טענה 1: ניתן לממש את A ו-B כך שרק אחת משלוש השורות תתקמפל.

טענה 2: ניתן לממש את A ו-B כך שרק שורות 1 ו-2 יתקמפלו.

טענה 3: ניתן לממש את A ו-B כך ששלוש השורות יתקמפלו.

נימוק:

בחרו את התשובה הנכונה ביותר:

- א. אף טענה אינה נכונה
- ב. רק טענה 1 נכונה
- ג. רק טענה 2 נכונה
- ד. רק טענה 3 נכונה
- ה. רק טענות 1+2 נכונות
- ו. רק טענות 1+3 נכונות
- ז. רק טענות 2+3 נכונות
- ח. כל הטענות נכונות

17. שאלה 17:

```
public class Outer {
    public void f() { new NI().n(); }
    private static void s() { new NS().m(); }
    public static class NS {
        private static void m() {
            f();
            s();
        }
    }
    public class NI {
        private void n() {
            f();
            s();
        }
    }
}
```

קבעו אילו מהמתודות n,m,s,f מתקמפלות. שימו לב, בבואכם לבדוק את הקומפיליציה של מתודה x המשתמשת במתודה y, הניחו כי y היא תקינה ומתקמפלת, גם אם זה לא נכון.  
נימוק:

- א. רק מתודות n,m מתקמפלות.
- ב. רק מתודות f,s מתקמפלות.
- ג. רק מתודה f מתקמפלת.
- ד. רק מתודות m,f מתקמפלות.
- ה. רק מתודות m,s מתקמפלות.
- ו. רק מתודות n,f מתקמפלות.
- ז. כל המתודות מתקמפלות.

ח. מלבד תשובה זו, כל התשובות לא נכונות.



18. שאלה 18:

מה יקרה בהרצת התוכנית B?

```
public class A {
    public String str = "A";
    public boolean bool = false;

    public void foo() {
        try { moo(); }
        catch (Exception exp){}
    }

    public void flipBool() { bool = !bool; }

    public void moo() throws Exception {
        System.out.print("A.moo " + bool + " ");}
    public static void doo(A a) { a.foo(); }
}

public class B extends A {
    protected String str = "B";

    public void moo() {System.out.print("B.moo " + bool + " ");}

    public void foo() {
        flipBool();
        super.foo();
    }
    public static void doo(A a) {
        a.flipBool();
        a.doo(a);
    }

}

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

נימוק:

- א. ישנה שגיאת קומפילציה במתודה main.
- ב. ישנה שגיאת קומפילציה במחלקה B מחוץ למתודה main.
- ג. התוכנית תרוץ ותדפיס B.moo true B.moo false
- ד. התוכנית תרוץ ותדפיס B.moo false B.moo false
- ה. התוכנית תרוץ ותדפיס A.moo true B.moo false
- ו. התוכנית תרוץ ותדפיס A.moo false B.moo false
- ז. התוכנית תרוץ ותדפיס A.moo false B.moo true
- ח. התוכנית תרוץ ותדפיס A.moo false B.moo true

19. שאלה 19:

```
public interface I<T>{
    public T foo(T t, I<T> it) ;
}

public class Test{

    public static <T> void func(I<T> i, I<T> ii, T t) {
        System.out.println(i.foo(t,ii));    /**
    }

    public static void main(String[] args) {
        I<Integer> i1 = (x,y)->x*2; /**
        I<String> i2 = (x,y)->x+x; /****
        func(i1, i2, 1); /*****
    }
}
```

מה יקרה בהרצת התוכנית הבאה?  
נימוק:

- א. יש שגיאת קומפילציה בשורה המסומנת ב \*
- ב. יש שגיאת קומפילציה בשורה המסומנת ב \*\*
- ג. יש שגיאת קומפילציה בשורה המסומנת ב \*\*\*
- ד. יש שגיאת קומפילציה בשורה המסומנת ב \*\*\*\*
- ה. התוכנית מתקמפלת, רצה ומדפיסה 3.
- ו. התוכנית מתקמפלת, רצה ומדפיסה 2.
- ז. התוכנית מתקמפלת, רצה, וזורקת ClassCastException בזמן ריצה.

20. שאלה 20:

נתון מימוש המחלקה Sub.

```
public class Sub extends Base{
    public String func(String str) {
        return (String)super.func(str);
    }
}
```

אילו מבין המימושים הבאים למחלקה Base יאפשרו ל Sub להתקמפל? (כמובן ש Base צריכה גם כן להתקמפל).

```
//v1
public class Base{
    public Object func(String str) { return null;}
}

//v2
public class Base{
    public String func(String str)
        throws RuntimeException{ return null;}
}

//v3
public class Base{
    public final String func(String str){ return null;}
}
```

אילו מבין המימושים מתאים? בחר/י בתשובה הטובה ביותר:  
נימוק:

- א. כל המימושים מתאימים.
- ב. כל המימושים לא מתאימים.
- ג. רק מימוש 1 מתאים.
- ד. רק מימוש 2 מתאים.
- ה. רק מימוש 3 מתאים.
- ו. רק מימושים 1+2 מתאימים.
- ז. רק מימושים 1+3 מתאימים.
- ח. רק מימושים 2+3 מתאימים.

21. שאלה 21:

```
public class Test {
    public static void main(String[] args) {
        List<String> strs = Arrays.asList("abc", "d", "ef", "aaa");
        strs.stream()
            .filter(x -> {
                System.out.print("!_");
                return !x.contains("d");
            })
            .peek(x -> {
                System.out.print("?_");
            })
            .sorted((x, y) -> {
                int res = Integer.compare(x.length(), y.length());
                if (res == 0) {
                    return x.compareTo(y);
                }
                return res;
            })
            .forEach(x -> System.out.print(x + "_"));
    }
}
```

מה יודפס בהרצת התוכנית הבאה?

- א. !\_!!\_!\_?\_?\_?\_ef\_abc\_aaa\_
- ב. !\_?\_!\_!\_?\_!\_?\_ef\_abc\_aaa\_
- ג. !\_!!\_!\_!\_?\_?\_?\_ef\_aaa\_abc\_
- ד. !\_?\_!\_!\_?\_!\_?\_ef\_aaa\_abc\_
- ה. !\_!\_?\_ef\_!\_?\_abc\_!\_?\_aaa\_
- ו. !\_!\_?\_ef\_!\_?\_aaa\_!\_?\_abc\_
- ז. !\_?\_ef\_!\_?\_abc\_!\_?\_aaa\_
- ח. !\_?\_ef\_!\_?\_aaa\_!\_?\_abc\_