

בחינה בתוכנה 1

סמסטר ב' תשע"ט, מועד א', 16 ביולי 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 לכתובה נכונה ב Java.

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

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:

המחלקה Point מוגדרת באופן הבא:

```
public class Point {  
    private int x;  
    private int y;  
  
    public Point(int x, int y){  
        this.x = x;  
        this.y = y;  
    }  
}
```

סטודנטים בקורס תוכנה 2 התבקשו למיין רשימה של נקודות פעמיים, פעם אחת לפי קואורדינטת ה-x ופעם אחת לפי קואורדינטת ה-y. המיון יתבצע ע"י שימוש בפונקציות sort של המחלקה Collections. הניחו כי לא ניתן לשנות את קוד המחלקה Point הנתון, וניתן רק להוסיף מימושים למתודות המופיעות במנשקים Comparable ו-Comparator. סטודנטית א' הציעה שהמחלקה Point מממש את המנשק Comparable, ויהיו שני מימושים שונים למתודה compareTo על ידי שימוש בהעמסה סטודנטית ב' הציעה לכתוב מחלקה חדשה (לא פנימית) שתממש את המנשק Comparator, ובמחלקה זאת לתת שני מימושים שונים למתודה compare על ידי שימוש בהעמסה סטודנטית ג' הציעה שהמחלקה Point תממש את המנשק Comparable וגם ניצור מחלקה חדשה (לא פנימית) אשר תממש את המנשק Comparator

בחר/י את התשובה הנכונה ביותר

נימוק:

- א. כל הסטודנטיות טועות
- ב. יש בדיוק סטודנטית אחת אשר צודקת
- ג. יש בדיוק שתי סטודנטיות אשר צודקות
- ד. כל הסטודנטיות צודקות

2. שאלה 2:

שאלה זו מתייחסת למימוש המחלקה Point הנתון בשאלה 1:
טענה 1: כל אובייקט שמקיים יחס is-a עם המחלקה Point הוא בהכרח immutable.
טענה 2: השדות של המחלקה Point הם פרימיטיביים, ולכן נשמרים על ה stack.
טענה 3: הקריאה `(new Point(1,2)).equals(new Point(1,2))` תחזיר true.
בחר/י בתשובה הטובה ביותר:

נימוק:

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

3. שאלה 3:

אילו מהטענות הבאות אינה נכונה? יש לסמן את התשובה "יש יותר מטענה אחת לא נכונה" אם מבין שאר הטענות, יש יותר מטענה אחת שאינה נכונה.

נימוק:

- א. במנשק ניתן להגדיר מתודות סטטיות ואילו במחלקה אבסטרקטית לא
- ב. גם במנשק וגם במחלקה אבסטרקטית ניתן לממש מתודות שאינן סטטיות.
- ג. גם במנשק וגם במחלקה אבסטרקטית ניתן להגדיר קבועים (public static final).
- ד. הטיפוס הסטטי של אובייקט כלשהו יכול להיות מנשק או מחלקה אבסטרקטית, אך לא הטיפוס ~~הדינמי~~ ~~אך הטיפוס הדינמי לא יכול להיות מנשק או מחלקה אבסטרקטית~~
- ה. מחלקה אנונימית יכולה לרשת ממחלקה אבסטרקטית, יכולה לממש מנשק, ויכולה גם לעשות את ~~את שניהם בו זמנית~~.
- ו. יש יותר מטענה אחת לא נכונה

```
public class MyList<T> {
    private Cell<T> head, lastItem;
    private int listLength;

    /*@pre elements.length > 0 */
    public MyList(T... elements) {
        this.head= null;
        for (int i= elements.length-1; i>= 0; i--) {
            head = new Cell<T>(elements[i], head) ;
            if (lastItem == null) {lastItem = head;}
            listLength++;
        }

        public int getLength(){ return listLength; }

        public List<T> toList(){
            List<T> newList = new ArrayList<T>();
            Cell<T> tmp = head;
            while(tmp != null){
                newList.add(tmp.getCont());
                tmp = tmp.getNext();
            }
            return newList;}

        public void addToLast(T cont){
            lastItem.setNext(new Cell<>(cont, null));
            listLength++;
        }

        public boolean isInList(T cont){
            Cell<T> tmp = head;
            while(tmp.getNext() != null){
                if (tmp.getNext().getCont() == cont){
                    return true;
                }
                tmp = tmp.getNext();
            }
            return false;
        }
    }

    public class Cell<T>{
        private T cont;
        private Cell<T> next;

        public Cell(T content, Cell<T> next){
            this.cont = content;
            this.next = next;
        }

        public void setNext(Cell<T> next){ this.next = next; }

        public T getCont(){ return this.cont; }

        public Cell<T> getNext(){ return this.next; }
    }
}
```

שמורה 1: `listLength == toList().size()` :`@imp_inv`
שמורה 2: `getLength() != 0` :`@inv`
שמורה 3: `contains(x)` :`isInList(x)` if and only if `toList().contains(x)` :`@inv`
בחר/י בתשובה הטובה ביותר:

נימוק:

- א. כל השמורות מתאימות למימוש.
- ב. רק שמורה 1 מתאימה למימוש.
- ג. רק שמורה 2 מתאימה למימוש.
- ד. רק שמורה 3 מתאימה למימוש.
- ה. רק שמורות 1+2 מתאימות למימוש.
- ו. רק שמורות 1+3 מתאימות למימוש.
- ז. רק שמורות 2+3 מתאימות למימוש.
- ח. כל השמורות לא מתאימות למימוש.

5. שאלה 5:

בשאלה זו, נרצה לאפשר מעבר על MyList באמצעות איטרטור. לצורך כך, נבצע את השינויים הבאים:

```
public class MyList<T> implements Iterable<T>, Iterator<T> {
    //the members defined in q4
    private Cell<T> iter;

    public MyList(T... elements) {
        //the code in q4
        this.iter = head; //outside the for block
    }

    //the rest of the methods implemented in q4

    public boolean hasNext() { return iter.getNext() != null; }

    @Override
    public T next() {
        T content = iter.getCont();
        iter = iter.getNext();
        return content;
    }

    @Override
    public Iterator<T> iterator() { return this; } // *
}
```

כעת, ננסה להריץ את התוכנית הבאה:

```
public class Test{
    public static void main(String[] args){
        MyList<String> mL = new MyList<>("a", "b", "c", "d", "e", "f", "g");
        Iterator<String> it1 = mL.iterator();
        Iterator<String> it2 = mL.iterator();
        for (int i = 0; i < 3; i++){
            System.out.print(it1.next() + it2.next() + "_");
        }
    }
}
```

נימוק:

- א. ישנה שגיאת קומפילציה במחלקה MyList בשורה המסומנת ב *.
- ב. הקוד מתקמפל. בהרצת התוכנית Test תיזרק שגיאת זמן ריצה ולא יודפס כלום.
- ג. הקוד מתקמפל. בהרצת התוכנית Test יודפס aa_bb_cc_.
- ד. הקוד מתקמפל. בהרצת התוכנית Test יודפס ab_cd_ef_.
- ה. הקוד מתקמפל. בהרצת התוכנית Test יודפס aa_cc_ee_.
- ו. הקוד מתקמפל. בהרצת התוכנית Test יודפס aa_aa_aa_.
- ז. הקוד מתקמפל. בהרצת התוכנית Test יודפס aa_cc_ee_.
- ח. הקוד מתקמפל. התוכנית Test תרוץ ללא חריגים ותסתיים ללא הדפסות.

6. שאלה 6:

מהי הטענה הנכונה? בחר/י בתשובה הנכונה ביותר:

נימוק:

- א. מנשקים נועדו לחסוך קוד בצד של הספק.
- ב. בהנתן מנשק כלשהו, למשל המנשק `IPoint` שראינו בכיתה, תבנית העיצוב `Factory` מאפשר למשתמש לממש יותר מבנאי אחד לכל מחלקה המממשת את `IPoint`.
- ג. תנאי `pre` מאפשרים לספק להגדיר הגבלות על הקלט של השירות.
- ד. השמורה (`Invariant`) מאפשרת לספק להגדיר הגבלות על השימוש במחלקה.
- ה. מלבד טענה זו קיימות לפחות שתי טענות נכונות.
- ו. מלבד טענה זו, כל הטענות לא נכונות.

7. שאלה 7:

השאלה מתייחסת למחלקה `BufferedWriter` אשר עובדת בדומה למחלקה `MyBufferedWriter` אותה מימשתם בתרגיל בית 7. הניחו כי `bw` מצביע לאובייקט מטיפוס `BufferedWriter`, וכן, שגודל ה `buffer` לא ידוע.

- 1. טענה 1. יתכן מצב בו מספר הפעמים ש `bw` כתב לקובץ בפועל קטן ממספר הקריאות ל `write`
- 2. טענה 2. יתכן מצב בו מספר הפעמים ש `bw` כתב לקובץ בפועל גדול ממספר הקריאות ל `write`
- 3. טענה 3. במידה והמשתמש לא קרא למתודה `close`, יתכן והקובץ הנוצר לא מכיל את כל המילים אותם ביקש המשתמש לכתוב
- 4. טענה 4. מספר הפעמים ש `bw` כתב לקובץ שווה בדיוק למספר הקריאות ל `write`.

בחר/י בהתשובה הנכונה ביותר

נימוק:

- א. ישנה טענה אחת נכונה בלבד.
- ב. ישנן בדיוק שתי טענות נכונות.
- ג. ישנן בדיוק שלוש טענות נכונות.
- ד. כל הטענות נכונות.
- ה. כל הטענות לא נכונות.

8. שאלה 8:

מהי הטענה הנכונה? בחר/י בתשובה הטובה ביותר:

נימוק:

- א. כל מתודה שזורקת כל סוג של חריג צריכה להצהיר עליו בחתימה שלה.
- ב. מתודה private יכולה לזרוק כל סוג של חריג מבלי להצהיר עליו, כי לא ניתן לקרוא לה מחוץ למחלקה.
- ג. בלוק catch יכול לתפוס רק חריגים שהם checked.
- ד. מתודה שדורסת מתודה אשר אינה מצהירה על חריג שהוא checked, יכולה בעצמה להצהיר על חריג שהוא checked.
- ה. מלבד טענה זו, כל הטענות לא נכונות.
- ו. מלבד טענה זו, יש יותר מטענה אחת נכונה.

9. שאלה 9:

משרד ממשלתי ביקש לממש מחלקה אשר תייצג רשימת של פריטי מידע מטיפוס String. בנוסף ישנה דרישה כי כל קוד אשר משתמש במחלקה לא יכול לשנות את תוכן רשימת המידע, אך כן יוכל לקרוא אותו על ידי שירות getData.

```
public class DataItems {  
    private List<String> data;  
    public List<String> getData() {return data;}  
}
```

הערה בזמן בחינה - קוד הבנאי לא מופיע, אך ניתן להניח כי הוא בטוח ודרכו לא ניתן לשנות את תוכן רשימת המידע

טענה 1. כדי להשיג התנהגות זאת מספיק להגדיר את השדה data כ-final ואת ה-getter כ-private

טענה 2. כדי להשיג התנהגות זאת מספיק להגדיר את המחלקה DataItems כ-final

טענה 3. כדי להשיג התנהגות זאת מספיק לממש את getData כך שיחזיר עותק של data.

בחר/י את הטענה הנכונה ביותר.

נימוק:

- א. רק טענה 1 נכונה
- ב. רק טענה 2 נכונה
- ג. רק טענה 3 נכונה
- ד. כל הטענות לא נכונות.
- ה. כל הטענות נכונות
- ו. אף טענה בפני עצמה נכונה, אבל אם נשלב את הדרישות שמופיעות בשתי טענות שונות, נשיג את ההתנהגות הרצויה.
- ז. אף טענה בפני עצמה אינה נכונה, ורק שילוב הדרישות שמופיעות בשלושת הטענות יחדיו ישיג את ההתנהגות הרצויה

10. שאלה 10:
התבונני/ בקטע הקוד הבא.

```
public class A {           // class A compiles
    public A(int i) {
        this(i,i);
        System.out.print(" A "+i);
    }
    public A(double i, double j) {
        System.out.print(" A "+i+" "+j);
    }
}
public class B extends A {
    public B(int i) {
        super(i);
        System.out.print(" B "+i);}

    public static void main(String[] args) {
        List<A> lst= new ArrayList<A>();
        lst.add(*****); /*
    }
}
```

בחרי את התשובה הנכונה
נימוק:

- א. אם נחליף את ***** ב- `new B()` הקוד יתקמפל ולא ידפיס כלום.
- ב. אם נחליף את ***** ב- `new B(1)` הקוד יתקמפל וידפיס: `A 1.0 1.0 A 1 B 1`
- ג. אם נחליף את ***** ב- `new B(1)` הקוד יתקמפל וידפיס: `A 1 A 1.0 1.0 B 1`
- ד. אם נחליף את ***** ב- `new B(1)` הקוד יתקמפל וידפיס: `B 1 A 1.0 1.0 A 1`
- ה. אם נחליף את ***** ב- `new B(1)` הקוד יתקמפל וידפיס: `B 1 A 1 A 1.0 1.0`
- ו. אם נחליף את ***** ב- `new B(1)` הקוד לא יתקמפל

11. שאלה 11.

```
public class A{
    private int a;
    public static void func() {}
    //some code here
    public class B{
        //some code here
    }
}

public class C extends A{
    //some code here
}
```

בחר/י בתשובה הטובה ביותר

נימוק:

- א. גם למחלקה B וגם למחלקה C יש גישה לשדה a של המחלקה A בהנתן אובייקט מטיפוס A.
- ב. המחלקות B ו C יכולות לקרוא לשירות func של A.
- ג. גם מחלקה B וגם מחלקה C מקיימות יחס is-a עם מחלקה A
- ד. מספר המופעים של מחלקה B בתכנית כלשהי בהכרח קטן או שווה למספר המופעים של מחלקה A באותה התכנית
- ה. מלבד תשובה ז, יש יותר מתשובה אחת נכונה
- ו. מלבד תשובה ז, כל התשובות לא נכונות.

12. שאלה 12:
התבונני בקטע קוד הבא.

```
public class A {
    public String str = "A";
    public void foo() {
        goo();
        A.moo();    /*
    }
    private void goo() {System.out.print(str+" ");}
    public static void moo() {System.out.print("A ");}
}

public class B extends A {
    public String str = "B";
    public void goo() throws Exception {System.out.print(str+" ");} /**
    public static void moo() {System.out.print("B ");}

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

נימוק:

- א. הקוד לא מתקמפל, שגיאת קומפילציה בשורה המסומנת ב-*
- ב. הקוד לא מתקמפל, שגיאת קומפילציה בשורה המסומנת ב-**
- ג. הקוד מתקמפל ורץ, יודפס AAAA
- ד. הקוד מתקמפל ורץ, יודפס AABB
- ה. הקוד מתקמפל ורץ, יודפס ABAB
- ו. הקוד מתקמפל ורץ, יודפס BABA
- ז. הקוד מתקמפל ורץ, יודפס BBBB

13. שאלה 13:

להלן קוד המתייחס למימושים של A, B ו I. הניחו כי יתכן וקיימות מחלקות נוספות שמימושן לא נתון.

```
public class Test{  
    public void func(A a, B b, I i){  
        //a = (A)b; //1  
        //b = (B)a; //2  
        //a = (A)i; //3  
    }  
}  
  
class A{ }  
  
class B extends A{ }  
  
interface I{ }
```

הערה: הניחו כי כל הקוד נמצא באותה החבילה

- טענה 1: אם נוציא את שורה 1 מההערה, הקוד יתקמפל ובוודאות לא יזרק חריג בזמן ריצה.
טענה 2: אם נוציא את שורה 2 מההערה, הקוד לא יתקמפל.
טענה 3: אם נוציא את שורה 3 מההערה, הקוד יתקמפל ובוודאות לא יזרק חריג בזמן ריצה.

בחר/י בתשובה הטובה ביותר:

נימוק:

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

14. שאלה 14:

להלן שלוש טענות לקשורות לפרמטרים גנריים ((תזכורת, המחלקה Integer מקיימת יחס is-a עם Number)).

טענה 1: מחלקה פנימית סטטית יכולה לעשות שימוש בפרמטר הגנרי של המחלקה העוטפת.

טענה 2: `Set<? extends Integer>` מקיים יחס is-a עם `Set<? extends Number>`.

טענה 3: `Set<? super Integer>` מקיים יחס is-a עם `Set<? extends Number>`.

בחרו את התשובה הנכונה ביותר:

נימוק:

- א. כל הטענות לא נכונות.
- 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

15. שאלה 15:

לפניכם 3 טענות לקשורות למחלקות אבסטרקטיות:

טענה 1: למחלקה אבסטרקטית תמיד קיים בנאי (אם הבנאי לא מוגדר, קיים בנאי דיפולטי).

טענה 2: מחלקה אבסטרקטית יכולה לרשת ממחלקה שאינה אבסטרקטית.

טענה 3: במחלקה אבסטרקטית יכולים להיות שירותים אבסטרקטיים בנראות private.

בחרו את התשובה הנכונה ביותר:

נימוק:

- 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

16. שאלה 16:

להלן מספר טענות:

- 1: הטיפוס String הוא למעשה שם אחר למערך של char-ים.
- 2: ניתן להצהיר על משתנה מקומי בתוך מתודה מבלי לתת לו ערך התחלתי.
- 3: קובץ class הוא למעשה קובץ טקסט אשר מכיל פקודות ב Java וניתן לפתוח אותו באמצעות כל עורך טקסט.
- 4: האיבר הראשון במערך args שמקבלת מתודת ה main הוא תמיד שם התוכנית.
- 5: המפרש (interpreter) של Java מייצר קבצי exe מקבצי ה class.

בחר/י בתשובה הטובה ביותר:

נימוק:

- א. כל הטענות לא נכונות.
- ב. יש רק טענה נכונה אחת.
- ג. יש רק שתי טענות נכונות.
- ד. יש רק שלוש טענות נכונות.
- ה. יש רק ארבע טענות נכונות.
- ו. כל הטענות נכונות.

```
public class E {  
    public static void main(String[] args) {  
        int x = 0;  
        int[] a = { 2, 1, 0 };  
        for (int i : a) {  
            try {  
                if ((6 / i) < 3) throw new IOException();  
            }  
            catch (IOException e) { x++; }  
            catch (Exception e) { x += 2; }  
            finally {x++;}  
        }  
        System.out.println(x);  
    }  
}
```

מה יקרה בהרצת הקוד הבא?

נימוק:

- א. תיזרק שגיאה לפני שהתוכנית תדפיס פלט כלשהו.
- ב. ריצת התוכנית תסיים בהצלחה ויודפס 8.
- ג. ריצת התוכנית תסתיים בהצלחה ויודפס 7.
- ד. ריצת התוכנית תסיים בהצלחה ויודפס 6.
- ה. ריצת התוכנית תסיים בהצלחה ויודפס 5.
- ו. ריצת התוכנית תסיים בהצלחה ויודפס 4.
- ז. ריצת התוכנית תסיים בהצלחה ויודפס 3.
- ח. ריצת התוכנית תסתיים בהצלחה ויודפס 2.


```
public class Box<T extends Comparable> {  
  
    public static <T extends Comparable> int comp(T v1, T v2) {  
        return v1.compareTo(v2);  
    }  
  
    public int f1(List<? extends T> l) {  
        return comp(l.get(0), l.get(1));  
    }  
  
    public int f2(List<? extends Comparable> l) {  
        return comp(l.get(0), l.get(1));  
    }  
  
    public int f3(List<String> l) {  
        return comp(l.get(0), l.get(1));  
    }  
  
    public static void main(String[] args) {  
        Box<String> b = new Box<>();  
        List<Integer> ints = Arrays.asList(1,2);  
        List<String> strings = Arrays.asList("1","2");  
        b.f1(ints); // *  
        b.f1(strings); // $  
        b.f2(ints); // #  
        b.f2(strings); // @  
        b.f3(strings); // &  
    }  
}
```

אילו מהקריאות שנמצאות בתוך מתודת ה-main מתקמפלות? בחר/י בתשובה הטובה ביותר
נימוק:

- א. כולן מלבד *
- ב. כולן מלבד * ו-#
- ג. כולן מלבד &
- ד. כולן מלבד \$
- ה. כולן מלבד \$ ו-#
- ו. כולן מלבד * ו-\$
- ז. רק @ ו-& מתקמפלות.
- ח. רק & מתקמפלת.

```
public class A {  
    public int val;  
    public A(int val) { this.val = val; }  
  
    public static void main(String[] args) {  
        Collection<A> s;  
        //s = new LinkedHashSet<>(); //op1  
        //s = new TreeSet<>(); //op2  
        //s = new HashSet<>(); //op3  
        s.add(new A(3));  
        s.add(new A(1));  
        s.add(new A(2));  
        for (A a : s) {  
            System.out.print(a.val);  
        }  
    }  
}
```

לפניכם שלוש אופציות לאתחל את המשתנה s. נתון כי ריצה מסויימת של התוכנית הדפיסה את הפלט 312. ע"י איזו מבין האופציות s יכול היה להיות מאותחל בריצה זו ?

בחר/י בתשובה הטובה ביותר

נימוק:

- א. רק אופציה 1.
- ב. רק אופציות 1 או 2.
- ג. רק אופציות 1 או 3.
- ד. רק אופציות 2 או 3.
- ה. כל אחת מהאופציות.

20. שאלה 20:

```
public class A{
    public void func(String a, Object b) {System.out.println("A");}
}

public class B extends A{
    public void func(String a, Object b) {System.out.println("B1");}
    public void func(Object a, String b) {System.out.println("B2");}
    public void func(String a, String b) {System.out.println("B3");}
}

public class Test{
    public static void main(String[] args){
        String s1 = "abc";
        String s2 = "def";
        A a = new B();
        a.func(s1, s2);
    }
}
```

בחר/י בתשובה הטובה ביותר:

נימוק:

- א. המחלקה B לא מתקמפלת.
- ב. המחלקה Test לא מתקמפלת.
- ג. המחלקה Test מתקמפלת. במהלך הרצת התוכנית תיזרק שגיאת זמן ריצה.
- ד. המחלקה Test מתקמפלת. בהרצת התוכנית יודפס A.
- ה. המחלקה Test מתקמפלת. בהרצת התוכנית יודפס B1.
- ו. המחלקה Test מתקמפלת. בהרצת התוכנית יודפס B2.
- ז. המחלקה Test מתקמפלת. בהרצת התוכנית יודפס B3.

```
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("f");
            })
            .sorted((x, y) -> {
                int res = Integer.compare(x.length(), y.length());
                if (res == 0) {
                    return y.compareTo(x);
                }
                return res;
            })
            .map(x -> {
                System.out.print("?_");
                return x + x;
            })
            .forEach(x -> System.out.print(x + "_"));
    }
}
```

מה יודפס בהרצת התוכנית הבאה?

נימוק:

- א. !_!?!_?_dd_?_abcabc_?_aaaaaa_
- ב. !_!?!_?_dd_?_aaaaaa_?_abcabc_
- ג. !_!?!_?_?_?_dd_abcabc_aaaaaa_
- ד. !_!?!_?_?_?_dd_aaaaaa_abcabc_
- ה. i_?_dd_i_?_abcabc_i_i_?_aaaaaa_
- ו. i_?_dd_i_?_aaaaaa_i_i_?_abcabc_