

מספר סידורי: _____ מספר ת"ז: _____

בחינה בתוכנה 1

סמסטר א' תשע"ט, מועד א', 29 בינואר 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 בטופס הנימוקים המצורף בלבד. 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. תשובות ונימוקים 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 מחשבונים 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, לשרד, לאחסן במאגר מידע, בכל דרך שהיא, בין מכונית ובין אלקטרונית או בכל דרך אחרת כל חלק שהוא מטופס הבחינה. בהצלחה!

1. שאלה 1:

התבונני בקוד הבא ובחרי' בתשובה הטובה ביותר:

```
/** @inv isEmpty() implies top() != null */
public class MyStack{
    private final List<Object> elements = new ArrayList<>();

    /** @pre objs[objs.size()-1] != null
     * @post isEmpty()*
     * @post top() == objs[objs.size()-1] */
    public void pushMany(Object[] objs){
        for (Object o : objs){
            elements.add(o);
        }
    }

    /** @pre !isEmpty()
     * @post @return == @prev(top()) */
    public Object pop(){
        final Object popped = top();
        int lastElementIndex = elements.size()-1;
        elements.remove(lastElementIndex); lastElementIndex--;
        while(lastElementIndex >= 0){
            if (elements.get(lastElementIndex) == null){
                elements.remove(lastElementIndex);
                lastElementIndex--;
            }
            else{
                break;
            }
        }
        return popped;
    }

    /** @pre !isEmpty()
     * @post @return != null */
    public Object top(){
        return elements.get(elements.size()-1);
    }

    /** @post @return == true iff elements.size() == 0 */
    public boolean isEmpty(){
        return elements.size() == 0;
    }
}
```

- א. רק השירות pop מפר את השמורה (invariant) של המחלקה.
- ב. רק השירות top מפר את השמורה (invariant) של המחלקה.
- ג. רק השירות pushMany מפר את השמורה (invariant) של המחלקה.
- ד. קיים יותר משירות אחד אשר מפר את השמורה (invariant) של המחלקה.
- ה. כל השירותים שומרים על השמורה (invariant) של המחלקה.

2. שאלה 2:

להלן מספר טענות בנושא מנשקים:

- טענה 1: מנשק חייב להכיל שירות מופשט (abstract method) אחד לפחות.
- טענה 2: מנשק יכול להכיל מימושים של פונקציות סטטיות.
- טענה 3: מנשק יכול להרחיב (extend) מכמה מנשקים.
- טענה 4: מנשק יכול להכיל שדות מופע.

בחר/י בתשובה הטובה ביותר:

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

3. שאלה 3:

```
public class Test {
    public static void main(String[] args) {
        Stream<Integer> s= Stream.generate(new NaturalNumbers());
        boolean res = s.filter(x-> x %10 == 0 && x > 5)
            .filter(x-> x < 50)
            .anyMatch(x -> {
                System.out.println("!");
                return x%25 == 0;
            });
        System.out.println(res);
    }
}

public class NaturalNumbers implements Supplier<Integer>{
    private int i;

    public Integer get() { return++i;}
}
```

בחר/י בתשובה הטובה ביותר:

- א. התוכנית אינה מבצעת שום הדפסה ואינה עוצרת.
- ב. התוכנית מדפיסה false!!!! ומסתיימת.
- ג. התוכנית מדפיסה true!!!! ומסתיימת.
- ד. התוכנית מדפיסה אינסוף פעמים את הסימן ! ולא עוצרת.
- ה. כל התשובות מלבד זו לא נכונות.
- ו. התוכנית מדפיסה רק false ומסתיימת.
- ז. התוכנית מדפיסה רק true ומסתיימת.

4. שאלה 4:

נתונות המחלקות A,B,C,D המוגדרות באופן הבא:

```
public class A{ /* some code here */ }  
public class B extends A /* some code here */ }  
public class C extends A /* some code here */ }  
public class D extends B /* some code here */ }  
public class E { /* some code here */ }
```

לפניכם שירות המבצע מספר פעולות casting :

```
public static void func(B b) {  
    C c = (C)b;    // line 1  
    E e = (E)b;    // line 2  
    A a = (A)b;    // line 3  
}
```

טענה 1: שורה 1 תמיד תתקמפל אך עלולה לזרוק חריג בזמן ריצה.
טענה 2: שורה 2 תמיד תתקמפל אך עלולה לזרוק חריג בזמן ריצה.
טענה 3: שורה 3 תמיד תתקמפל אך עלולה לזרוק חריג בזמן ריצה.

בחר/י בתשובה הטובה ביותר:

- 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.

5. שאלה 5:

```
public class Box<V> {  
    public <T> void func1(List<T> lst){/*some code here */}  
  
    public void func2(List<?> lst) {/*some code here */}  
  
    public void func3(List<?> lst1, List<?> lst2) {/*some code here */}  
  
    public void func4(List<V> lst1, List<V> lst2) {/*some code here */}  
  
    public <T> void func5(List<T> lst1, List<T> lst2) {/*some code here */}  
}
```

לפניכם מספר טענות לגבי הפונקציות func1-5:
טענה 1: ניתן להחליף כל קריאה ל func2 בקריאה ל func1 ללא שינוי נוסף בקוד, והקוד עדין ימשיך להתקמפל.
טענה 2: ניתן להחליף כל קריאה ל func5 בקריאה ל func4 ללא שינוי נוסף בקוד, והקוד עדין ימשיך להתקמפל.
טענה 3: ניתן להחליף כל קריאה ל func5 בקריאה ל func3 ללא שינוי נוסף בקוד, והקוד עדין ימשיך להתקמפל.

הערה: החלפת כל קריאה ל X בקריאה ל Y פירושו החלפה חד כיוונית של X ב Y. זה לא אומר שניתן להחליף גם את Y ב X.

בחר/י בתשובה הטובה ביותר:

- 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.

6. שאלה 6:

```
public class Test {  
    public static void main(String[] args) {  
        List<String> strs = Arrays.asList("a", "b", "c"); //This compiles  
        List<Object> objs = Arrays.asList("a", "b", "c"); //This compiles  
        Test t = new Test();  
        t.func(strs);  
        t.func(objs);  
    }  
  
    public void func(List<String> lst){ System.out.println("String"); }  
    public void func(List<Object> lst){ System.out.println("Object"); }  
}
```

בחר/י בתשובה הטובה ביותר:

- א. הקוד מתקמפל והתוכנית תדפיס String Object
- ב. הקוד מתקמפל והתוכנית תדפיס String String
- ג. הקוד מתקמפל והתוכנית תדפיס Object Object
- ד. הקוד לא מתקמפל. אם נסיר את המימוש של main, הקוד כן יתקמפל.
- ה. הקוד לא מתקמפל. אם נסיר את המימוש של main, הקוד ימשיך לא להתקמפל.

7. שאלה 7:

נתונה מחלקה A כלשהי שידוע כי היא אבסטרקטית.

טענה 1: במחלקה A אין בנאים.

טענה 2: במחלקה A אין מתודות עם ניראות `private`.

טענה 3: במחלקה A אין שירותים שדורסים (override) שירותים שנורשו (inherited) ממחלקה אחרת.

טענה 4: מחלקה A אינה מממשת מנשקים.

אילו מבין הטענות נכונות לכל מחלקה A אבסטרקטית? בחר'י בתשובה הטובה ביותר:

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

8. שאלה 8:

```
public class Item{
    private int i;

    private Item(int i){ this.i = i; }

    public int getNum() { return i; }

    public static final Item I1 = new Item(1);
    public static final Item I2 = new Item(2);
    public static final Item I3 = new Item(3);
    public static Item I4 = new Item(4);
}

public class Test{
    public static void main(String[] args){
        /* **** you can add your code here **** */
        System.out.println(Item.I4.getNum()); // line 1, remains unchanged
        System.out.println(Item.I4 == Item.I3); // line 2, remains unchanged
    }
}
```

לפניכם מספר טענות המתייחסות למימושים אפשריים של השירות `main` במחלקה `Test`. שימו לב, השורות המסומנות ב 1 ו 2 ישארו ללא שינוי. כמו כן, אין לשנות את מימוש המחלקה `Item`.

- טענה 1: ניתן לממש את `main` כך שבשורה 1 יודפס 5.
- טענה 2: ניתן לממש את `main` כך בשורה 1 יודפס 1.
- טענה 3: ניתן לממש את `main` כך שבשורה 2 יודפס `true`.

בחר'י בתשובה הטובה ביותר:

- 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.

9. שאלה 9:

נרצה לממש את המחלקה MyOptional הדומה למחלקה Optional אותה ראינו בשיעור על Java8. נשתמש במחלקה זו בפונקציות שצריכות להחזיר ערך מטיפוס כלשהו, אבל בפועל לא תמיד יש להן ערך להחזיר.

דוגמת שימוש:

```
public class Test{
    public static void main(String[] args){
        List<String> lst = Arrays.asList("abc", "de", "fg");
        MyOptional<Integer> opRes = func(lst);
        if (opRes.hasValue()){
            System.out.println(opRes.getValue());
        }
        else{
            System.out.println("no results");
        }
    }

    public static MyOptional<Integer> func(List<String> strings){
        /* some code here */
    }
}
```

להלן המימוש של MyOptional:

```
public class MyOptional<T>{
    private boolean isInitialized = false;
    private T value;

    public MyOptional(T value) {
        this.value = value;
        this.isInitialized = true;
    }

    public boolean hasValue(){ return this.isInitialized; }

    /** @pre hasValue() == true; */
    public T getValue(){ return this.value; }
}
```

לפניכם מספר טענות הקשורות למחלקה MyOptional:

טענה 1: המתודה main משתמשת במחלקה MyOptional נכון על פי מה שמוגדר בחוזה של

MyOptional.

טענה 2: ניתן לממש את func (מבלי לשנות את שאר הקוד) כך שבהרצת התוכנית Test, הקוד של

main ידפיס no results.

טענה 3: ניתן לממש את func (מבלי לשנות את שאר הקוד) כך שבהרצת התוכנית Test, הקוד של

main ידפיס מספר שלם (integer).

בחר/י בתשובה הטובה ביותר:

- 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.

10. שאלה 10:

```
public class Container<S> {  
  
    public void f1(List<? extends Float> l1, List<? super Number> l2) {  
        l2.add(l1.get(0));  
    }  
  
    public <E> void f2(List<Object> l1, List<?> l2) {  
        l1 = l2;  
    }  
  
    public void f3(Collection<?> c, List<S> l ) {  
        c = l;  
    }  
  
}
```

אילו מבין הפונקציות f1-3 המופיעות במחלקה Container מתקמפלות? בחר/י את התשובה הטובה ביותר. תזכורת: Float יורש מ Number.

- א. רק f1 מתקמפלת.
- ב. רק f2 מתקמפלת.
- ג. רק f3 מתקמפלת.
- ד. רק f1+f2 מתקמפלות.
- ה. רק f1+f3 מתקמפלות.
- ו. רק f2+f3 מתקמפלות.
- ז. כל הפונקציות מתקמפלות.
- ח. כל הפונקציות לא מתקמפלות.

11. שאלה 11:

```
public class A {  
    A f(int x, int y) throws IOException { return new A(); }  
}  
  
public class B extends A {  
  
    /* B f(int x, int y) { return new B(); } */  
  
    /* protected A f(int x, int y) throws Exception { return new A(); } */  
  
    /* A f(double x, double y) throws Exception { return new B(); } */  
  
    /* Object f(int x, int y) throws IOException { return new A(); } */  
  
    /* private A f(int x, int y) throws EOFException { return new A(); } */  
}
```

תזכורת: EOFException יורשת מ IOException. ידוע כי המחלקות A ו-B נמצאות באותה חבילה. הניחו כי בכל פעם נוסף מתודה אחת בלבד למחלקה B. בחרו בתשובה הנכונה ביותר.

- א. עבור כל מתודה שנוסף הקוד יתקמפל.
- ב. עבור כל מתודה שנוסף תיוצר שגיאת קומפילציה.
- ג. ישנה בדיוק אחת שהוספתה תייצר שגיאת קומפילציה.
- ד. ישנן בדיוק שתי מתודות שהוספת כל אחת מהן תייצר שגיאת קומפילציה.
- ה. ישנן בדיוק שלוש מתודות שהוספת כל אחת מהן תייצר שגיאת קומפילציה.
- ו. ישנן בדיוק ארבע מתודות שהוספת כל אחת מהן תייצר שגיאת קומפילציה.

12. שאלה 12:

```
public class E {
    public static void main(String[] args) {
        int x = 0;
        int[] a = { 1, 2, 3, 4 };
        for (int i : a) {
            if (i % 2 == 0)
                x++;
            else {
                try {
                    throw new RuntimeException();
                }
                catch (RuntimeException e) {
                    x++;
                }
                catch (Exception e) {
                    x++;
                }
                finally {
                    x++;
                }
            }
        }
        System.out.println(x);
    }
}
```

מה יקרה בהרצת הקוד הבא?

- א. התוכנית תעוף על חריג (Exception) ולא יודפס שום פלט.
- 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.

13. שאלה 13:

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

    public A(int i, int j) {
        this.i = i;
        this.j = j;
    }
    // the rest of the code is not provided

    public static void main(String[] args) {
        Set<A> s = new LinkedHashSet<>();
        s.add(new A(3, 1));
        s.add(new A(1, 3));
        s.add(new A(3, 1));
        s.add(new A(3, 1));
        s.add(new A(3, 1));
        s.add(new A(2, 1));
        System.out.println(s.size());
    }
}
```

להלן מספר טענות המתייחסות למימושים השונים של המחלקה A. הניחו כי הקוד הקיים של A לא ישתנה, וניתן רק להוסיף קוד. כמו כן, הניחו כי הקוד של A הוא דטרמיניסטי, כלומר, אין אקראיות (randomness) וכל פונקציה תחזיר את אותה התשובה בכל קריאה עבור אותו הקלט־מצב.

- טענה 1: קיים מימוש של A עבורו יודפס 5.
- טענה 2: קיים מימוש של A עבורו יודפס 4.
- טענה 3: קיים מימוש של A עבורו יודפס 3.
- טענה 4: קיים מימוש של A עבורו יודפס 2.
- טענה 5: קיים מימוש של A עבורו יודפס 1.
- טענה 6: קיים מימוש של A עבורו יודפס 0.

בחר/י בתשובה הטובה ביותר:

- א. רק טענה 1 לא נכונה.
- ב. רק טענה 2 לא נכונה.
- ג. רק טענה 3 לא נכונה.
- ד. רק טענה 4 לא נכונה.
- ה. רק טענה 5 לא נכונה.
- ו. רק טענה 6 לא נכונה.
- ז. קיימות שתי טענות לא נכונות.

14. שאלה 14:

```
public class A {
    public void f(Object o) { System.out.println("ao"); }
    public void f(String s) { System.out.println("as"); }
}

public class B extends A {
    public void f(Object o) { System.out.println("bo"); }
}

public class C extends B {
    public void f(Object o) { System.out.println("co"); }
    public void f(String s) { System.out.println("cs"); }

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

מה יודפס בסיום ריצת התוכנית? בחר/י בתשובה הטובה ביותר:

- א. ao
- ב. as
- ג. bo
- ד. co
- ה. cs

15. שאלה 15:

להלן קוד שידוע שהוא מתקמפל ורץ ללא שגיאות. המימוש של מחלקות A ו B לא נתון.

```
public class Test{
    public static void main(String[] args) {
        A<String> a = new A<>();
        Iterator<String> it = a.iterator();
        //it = new B(a);    /**
        while(it.hasNext()) {
            System.out.println(it.next());
        }
    }
}
```

בחר/י בתשובה הטובה ביותר:

- א. המחלקה A בהכרח מממשת את הממשק Iterable.
- ב. המחלקה A בהכרח מממשת את הממשק Iterator.
- ג. קיים מימוש של A ו B שעבורו, אם נוציא את השורה המסומנת ב *** מהערה, הקוד יתקמפל וירוך ללא שגיאות.
- ד. מלבד תשובה זו יש יותר מתשובה אחת נכונה.
- ה. מלבד תשובה זו, כל התשובות לא נכונות.

16. שאלה 16:

```
public class A {
    private int b = 2;

    public void func1(int c) {
        int d = 2;
        A a = new A();

        class Inner {
            private void f(int e) {
                // System.out.println(a.b); // *
                // System.out.println(b); // $
                // System.out.println(c); // #
                // System.out.println(d); // &
            }
        }
        d = 3;
        a.b = 4;
    }
}
```

הניחו כי בכל פעם מוסיפים שורה אחת בלבד למתודה f. באילו מקרים הקוד שלפניכם יתקמפל?

- א. הקוד יתקמפל בהוספת כל שורה מלבד השורה *.
- ב. הקוד יתקמפל בהוספת כל שורה מלבד השורה \$.
- ג. הקוד יתקמפל רק אם נוסיף את שורות # או &.
- ד. הקוד יתקמפל בהוספת כל שורה מלבד השורה &.
- ה. הקוד יתקמפל רק אם נוסיף את שורות \$ או &.
- ו. הקוד יתקמפל בכל הוספה של שורה.
- ז. יש רק שורה אחת שניתן להוסיף כך שהקוד יתקמפל.

17. שאלה 17:

```
public class A<T,S> {  
    private T t;  
    private S s;  
  
    public A(T t, S s) {this.t= t; this.s = s;}  
  
    public /*static*/ <K> void foo(K t) {  
        System.out.println(t+" "+s); /**  
    }  
  
    public static void main(String[] args) {  
        A a = new A("Hello", 0);  
        a.foo(0); /**  
    }  
}
```

בחר/י בתשובה הטובה ביותר:

- א. הקוד מתקמפל ומדפיס 00. אם המתודה foo תוגדר כמתודה סטטית, נקבל שגיאת קומפילציה בשורה המסומנת ב-*
- ב. הקוד מתקמפל ומדפיס Hello 0. אם המתודה foo תוגדר כמתודה סטטית, נקבל שגיאת קומפילציה בשורה המסומנת ב-*
- ג. הקוד מתקמפל ומדפיס Hello 0. אם המתודה foo תוגדר כמתודה סטטית, יודפס 00
- ד. הקוד אינו מתקמפל, שגיאת קומפילציה בשורה המסומנת ב-*. והוא גם לא יתקמפל אם המתודה foo תוגדר כמתודה סטטית
- ה. הקוד אינו מתקמפל, שגיאת קומפילציה בשורה המסומנת ב-**. והוא גם לא יתקמפל אם המתודה foo תוגדר כמתודה סטטית
- ו. הקוד אינו מתקמפל - שגיאת קומפילציה בשורה המסומנת ב-*, אך אם המתודה foo תוגדר כמתודה סטטית הקוד יתקמפל ויודפס 00
- ז. הקוד אינו מתקמפל - שגיאת קומפילציה בשורה המסומנת ב-**. אך אם המתודה foo תוגדר כמתודה סטטית הוא יתקמפל ויודפס 00

18. שאלה 18:

איזו בעיה מיועדת לפתור תבנית העיצוב Bridge (Bridge design pattern)?

- א. מחלקה C רוצה להיות מסוגלת לייצר מופעים של מחלקות המממשות את המנשק I, מבלי להכיר את המימושים הקונקרטיים שלהם.
- ב. מחלקה C רוצה להשתמש בשירות func של המחלקה A, אלא ש func מצריך פרמטרים שאין ל C אפשרות לייצר, ולכן היא צריכה לפנות לשירות נוסף שיבצע את ההמרה.
- ג. מחלקה C רוצה לעשות שימוש חוזר בקוד של המחלקות A ו B. שתיהן מתאימות לכך ש C תירש (Inherit) מהן, אבל בגלל מגבלות השפה, זה לא אפשרי.
- ד. מחלקות C ו B הן מחלקות שלא חולקות קוד משותף. למרות זאת, נרצה לייצר להן מחלקת בסיס משותפת ע"י מימוש מנשק ריק I כך שנוכל לשלוח גם את B וגם את C לפונקציות שמקבלות כפרמטר את המנשק I.

19. שאלה 19:

```
public class A {
    public String str = "A ";

    public static void foo() {System.out.print("A.foo ");}
    public void goo() {System.out.print("A.goo" + " " + str); foo(); }
}

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

    public static void foo() {System.out.print("B.foo");}
    public void goo() {System.out.print("B.goo" + " " + str); foo(); }

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

מה יודפס בהרצת התוכנית? בחר/י בתשובה הטובה ביותר:

- א. יודפס B.foo B.goo
- ב. יודפס A.foo A.goo
- ג. יודפס B.foo A.goo
- ד. יודפס A.foo A.goo
- ה. יודפס A.foo B.goo
- ו. יודפס B.foo A.goo
- ז. יודפס B.foo B.goo

20. שאלה 20:

המונח מחלקה עוטפת מתייחס למחלקה בה מוגדרת המחלקה הפנימית (הסטטית או הלא סטטית).
טענה 1: ניתן ליצור מופע של מחלקה פנימית סטטית גם אם לא קיים מופע של המחלקה העוטפת.
טענה 2: מחלקה מקומית של שירות יכולה לגשת לכל משתנה מקומי של השירות שבו היא מוגדרת.
טענה 3: ניתן להוציא כל מחלקה פנימית סטטית לקובץ נפרד ללא כל שינוי נוסף בקוד (למעט אולי שינוי בשורות ה import), כך שהתכנית תעבוד בדיוק באותו האופן.

- א. רק טענה 1 נכונה
- ב. רק טענה 2 נכונה
- ג. רק טענה 3 נכונה
- ד. קיימות בדיוק שתי טענות נכונות
- ה. כל הטענות נכונות
- ו. כל הטענות אינן נכונות

21. שאלה 21:

נרצה לממש מחלקה שהיא Immutable – כלומר, מרגע שנוצר אובייקט כלשהו מטיפוס זה, לא ניתן לשנות אותו.

```
public final class A{
    private int i;
    private int[] arr;

    public A(int i, int[] arr) {
        this.i = i;
        this.arr = arr;
    }

    public int getI() {
        return this.i;
    }

    public int[] getArr() {
        return this.arr;
    }
}

public class B{
    public static void main(String[] args) {
        //some code here
        A a = ???;
        //some code here
        //a hasn't changed since its creation
    }
}
```

האם מחלקה A היא immutable? בחר/י בתשובה הטובה ביותר:

- א. מחלקה A היא immutable.
- ב. מחלקה A אינה immutable. מספיק להסיר את המתודה getArr כדי ש A תהפוך להיות immutable.
- ג. מחלקה A אינה immutable. על מנת ש A תהפוך להיות immutable צריך להסיר גם את ~~getI~~ וגם את getI.
- ד. מחלקה A אינה immutable. גם אם נסיר את המתודות getI ו getArr המחלקה A תמשיך להיות לא immutable.