תוכנה 1

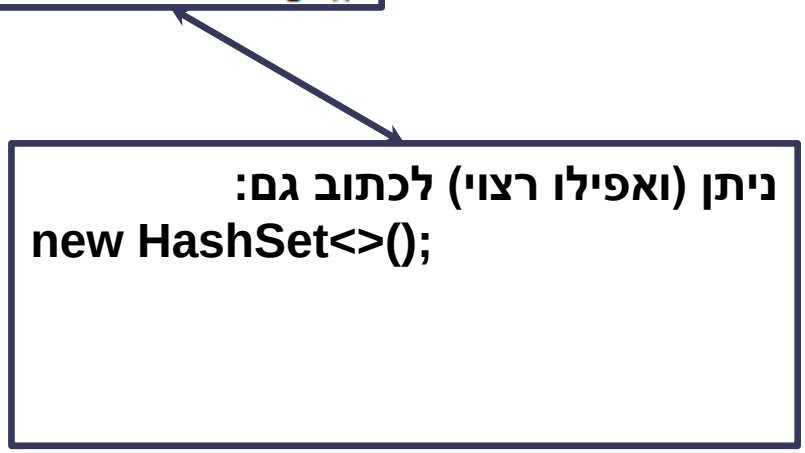
תרגול 13 – הכנה למבחן

אוספים גנריים

```
public static void func(HashSet<String> set){
    for (String s : set){
        System.out.println(s);
    }
}

public static void main(String[] args){
    HashSet<String> mySet = new HashSet<String>();
    mySet.add("abc");
    mySet.add("dce");
    func(mySet);

}
```



אוספים גנריים

```
public static void func(HashSet<String> set){
    for (String s : set){
        System.out.println(s);
    }
}

public static void main(String[] args){
    HashSet<String> mySet = new HashSet<String>();
    mySet.add("abc");
    mySet.add("dce");
    func(mySet);

}
```

האם אנחנו חייבים להצהיר על טיפוס סטטי שהוא HashSet? בד"כ נשתמש בטיפוס הכללי יותר Set אלא אם כן אנחנו נדרשים ספציפית ל HashSet. למשל במקרים הבאים:

1. **אנחנו רוצים להפעיל מתודה שיש ל HashSet אך לא ל Set (יש כזו בכלל?)**
2. **אנחנו משתמשים בשירות שדורש לקבל רק HashSet ולא Set.**

אוספים גנריים

```
public static void func(HashSet<String> set){
    for (String s : set){
        System.out.println(s);
    }
}

public static void main(String[] args){
    HashSet<String> mySet = new HashSet<String>();
    mySet.add("abc");
    mySet.add("dce");
    func(mySet);

}
```

מדוע הפונקציה דורשת לקבל HashSet? בד"כ נשתמש בטיפוס כמה שיותר כללי. האם נוכל לשלוח לפה כל Set? ע"מ המימוש שלה, אין סיבה שלא. למען האמת, נוכל לשלוח אפילו Collection.

אוספים גנריים

```
public static void func(Collection<String> set){
    for (String s : set){
        System.out.println(s);
    }
}

public static void main(String[] args){
    Set<String> mySet = new HashSet<>();
    mySet.add("abc");
    mySet.add("dce");
    func(mySet);

}
```

האם יש עוד משהו שנוכל לשפר בקוד?
נשים לב כי המימוש של func לא מחייב אותנו לקבל אוסף של מחרוזות. הדרישה היחידה היא שאברי האוסף יממשו את toString, מה שמובטח לכל אובייקט ב Java.

אוספים גנריים

```
public static void func(Collection<?> set){
    for (Object s : set){
        System.out.println(s);
    }
}

public static void main(String[] args){
    Set<String> mySet = new HashSet<>();
    mySet.add("abc");
    mySet.add("dce");
    func(mySet);

}
```

נשתמש ב <?> ע"מ לאפשר שימוש באוספים של כל טיפוס אפשרי.
שימו לב, להגדיר את set מטיפוס Collection<Object> לא ישיג את אותה המטרה, כיוון שאז נוכל להפעיל את הפונקציה הזו רק עם אובייקט מטיפוס סטטי Collection<Object>

אוספים גנריים

```
public static void func(Collection<?> set){
    for (Object s : set){
        System.out.println(s);
    }
}

public static void main(String[] args){
    Set<String> mySet = new HashSet<>();
    mySet.add("abc");
    mySet.add("dce");
    func(mySet);

}
```

נשתמש ב <?> ע"מ לאפשר שימוש באוספים של כל טיפוס אפשרי.
שימו לב, להגדיר את set מטיפוס Collection<Object> לא ישיג את אותה המטרה, כיוון שאז נוכל להפעיל את הפונקציה הזו רק עם אובייקט מטיפוס סטטי Collection<Object>

אוספים גנריים

```
public static void func(Collection<? extends Rectangle> set){
    for (Rectangle s : set){
        System.out.println(s.getArea());
    }
}

public static void main(String[] args){
    Set<Rectangle> mySet = new HashSet<>();
    mySet.add(new Rectangle(5, 6));
    mySet.add(new Rectangle(1,2));
    func(mySet);

}
```

שאלה 9

הסטודנטית ברית מעוניינת לממש מחלקה A כלשהי כך שתהיה immutable. A יורשת מ-Object, מכילה שדות מופע בלבד, ולא מוגדרות בתוכה מחלקות פנימיות. בנוסף, המחלקה A מוגדרת להיות final.

לפניכם מספר טענות על אופן המימוש של המחלקה A.

<u>טענה 1</u>: אם כל שדות המופע של A הם final, אז A היא immutable.

<u>טענה 2</u>: אם A היא immutable, כל שדות המופע שלה הם בהכרח final.

<u>טענה 3</u>: אם כל מתודות המופע ושדות המופע שמוגדרים ב-A הם פרטיים, אז A היא בהכרח immutable.

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

דוגמא למחלקה שהיא mutable

```
public final class A{
        private final int[] arr;
        public A(int[] arr){
                this.arr = arr;
        }
}
```

שאלה 10

לפניכם 3 טענות הקשורות לקוד גנרי:

<u>טענה 1</u>: אם T הוא פרמטר גנרי של מחלקה כלשהי, למחלקה זו יכול להיות שדה סטטי מטיפוס T.

<u>טענה 2</u>: מחלקה גנרית יכולה לרשת ממחלקה לא גנרית, ומחלקה לא גנרית יכולה לרשת ממחלקה גנרית.

<u>טענה 3</u>: ההשמה בשורה השניה בקוד המצורף תתקמפל תמיד, ללא תלות במה שיכתב במקום הכוכביות.

```
Set<******> genSet = ********;
Set<?> jokerSet = genSet;
```

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

שאלה 11

```
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

    @Override
    public boolean equals(Object obj) {
         return true;
    }

    public static void main(String[] args) {
        Set<A> s = new HashSet<>();
        s.add(new A(3, 1));
        s.add(new A(1, 3));
        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:

1. הקוד הקיים של A לא ישתנה, וניתן רק להוסיף קוד.
2. הקוד של A הוא דטרמיניסטי. כלומר, עבור אותו הקלט מחזירות את אותה התשובה בכל הרצה.
3. גודל טבלת ה hash גדול מ 4.

טענה 1: קיים מימוש של A עבורו יודפס 4.
טענה 2: קיים מימוש של A עבורו יודפס 3.
טענה 3: קיים מימוש של A עבורו יודפס 2.
טענה 4: קיים מימוש של A עבורו יודפס 1.

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- קיימות שתי טענות לא נכונות.

ו- קיימות שלוש טענות לא נכונות.

ז- כל הטענות נכונות.

שאלה 12

```
public class Car {
        private int year;
        private Engine engine  = new Engine(this);

        public Car(int year) {  this.year = year;  }

        public class Engine{
                public Car car;

                public Engine(Car car) {  this.car = car;}

                public int f() {  return year;  }
        }

        public static void main(String[] args) {
                Car c1 = new Car(1960);
                Car c2 = new Car(1970);
                c2.engine.car = c1.engine.car; // *
                System.out.println(c2.engine.f()); // **
        }
    }
```

א- יש שגיאת קומפילציה בשורה *.

ב- יש שגיאת קומפילציה בשורה **.

ג- תיזרק שגיאה בשורה ** ולא יודפס כלום.

ד- ריצת התוכנית תסתיים בהצלחה ויודפס 1960

ה- ריצת התוכנית תסתיים בהצלחה ויודפס 1970

שאלה 13

```
public class A {
        public int i = 1;
        public void foo() {   System.out.println(i);  }
}

public class B extends A {
        private int i = 3;    //*

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

א- התוכנית מדפיסה 11

ב- התוכנית מדפיסה 13

ג- התוכנית מדפיסה 31

ד- התוכנית מדפיסה 33

ה- קיימת שגיאת קומפילציה בשורה המסומנת ב *

ו- התוכנית מדפיסה 1 ועפה על חריג

ז- התוכנית מדפיסה 3 ועפה על חריג.

ח- קיימת שגיאת קומפילציה באחת מהבין השורות המסומנות ב ** וב ***

שאלה 14

```
public class A {
        protected String s = "A";

        public void f() {  System.out.print(s); }

        public A() { f(); }
}

public class B extends A {
        private String s = "B";

        public void f() { System.out.print(s); }

        public B() { f();  }

        public static void main(String[] args) {
                A a = new B(); // ***
                System.out.print(a.s);
        }
    }
```

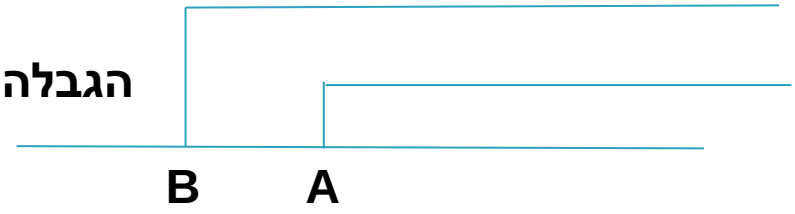
א- קיימת שגיאת קומפילציה במחלקה B.
ב- התוכנית עפה על חריג בשורה ***.
ג- התוכנית מדפיסה BBA
ד- התוכנית מדפיסה nullBA
ה- התוכנית מדפיסה ABA
ו- התוכנית מדפיסה BBB
ז- התוכנית מדפיסה nullBB
ח- התוכנית מדפיסה ABB

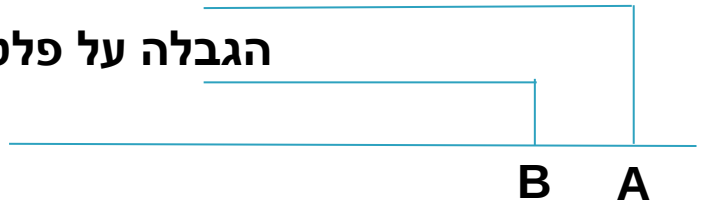
שאלה 15

```
class A{
   /**  missing contract */
    public int func(int i) { /* some implementation here */}
}

public class B extends A{
    /**
    * @pre i > 3
    * @post $ret < 10
    */
    @Override
    public int func(int i) { /* some implementation here  */}
}
```

החוזה של השירות func של המחלקה A אינו נתון.
מבין האופציות המוצעות, איזה חוזה הוא חוקי
ותקין על פי עקרונות הירושה? בחר/י בתשובה
הטובה ביותר:





א- @pre i>0, @post $ret < 20

ב- @pre i> 5, @post $ret < 20

ג- @pre i>0, @post $ret < 2

ד- @pre i>5, @post $ret<2

ה- מלבד תשובה זו, יש יותר מתשובה נכונה אחת.

ו- מלבד תשובה זו, כל התשובות לא נכונות.

שאלה 16 - gui

- לא נפתור אותה כי לא בחומר

שאלה 17

```
public class B{
        public static void func1(List<?> lst) {}
        public static <T> void func2(List<T> lst) {}
        public static void func3(List lst) {}
        public static void func4(List<Object> lst) {}
}
```

נרצה להוריד את השירות func1 מהמחלקה B.
לפניכם שלוש טענות:

טענה 1: ניתן להחליף את כל הקריאות ל func1 בקריאות ל func2, והקוד ימשיך להתקמפל.

טענה 2: ניתן להחליף את כל הקריאות ל func1 בקריאות ל func3, והקוד ימשיך להתקמפל.

טענה 3: ניתן להחליף את כל הקריאות ל func1 בקריאות ל func4, והקוד ימשיך להתקמפל.

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.

שאלה 18

1. על מנת שנוכל לפתח תוכניות Java (כלומר, לקמפל אותן), עלינו להתקין JRE על המחשב.

   חייב להיות JDK

2. תוכנית Java מקומפלת ניתן להריץ על כל מחשב שעליו מותקנת מערכת הפעלה עליה ביצענו את הקומפילציה.

   בפועל הטענה הזו שקולה ל-3 שגם היא לא נכונה. אם מניחים שמותקן JRE אז זה יכול לרוץ על כל מערכת הפעלה שהיא בלי קשר לאיפה זה קומפל

3. בהינתן קבצי class, ניתן להריץ אותם ללא התקנת java על המחשב.

   חייב להיות או JDK או JRE

4. מלבד תשובה זו, כל התשובות לא נכונות.

5. מלבד תשובה זו יש יותר מתשובה נכונה אחת.

שאלה 20

1. שדות סטטיים נשמרים על ה heap.
2. שדות סטטיים לא מקבלים ערכים דיפולטיים. כן מקבלים
3. שדה סטטי אינו יכול להיות final. יכול
4. שדה סטטי אינו יכול להיות מטיפוס פרימיטיבי. יכול
5. ניתן לגשת לשדה סטטי רק מתוך שירות סטטי. לא נכון, גם מלא סטטי
6. מלבד תשובה זו כל התשובות לא נכונות.
7. מלבד תשובה יש לפחות שתי תשובות נכונות.

שאלה 21

```
public class Bar {
  static class FirstException extends Exception {
         public FirstException(String message) { super("1" + message);}
  }
  static class SecondException extends FirstException {
         public SecondException(String message) { super("2" + message);}
  }
  static class ThirdException extends Exception {
         public ThirdException(String message) { super("3" + message);}
  }
  public static void main(String[] args) throws Exception{
         Exception e1 = new FirstException("");
         Exception e2 = new SecondException("");
         Exception e3 = new ThirdException("");
         List<Exception> l = Arrays.asList(e1,e2,e3);
         foo(l); //*
  }
  public static void foo(List<? extends Exception> lst) throws Exception{
      for (Exception ex : lst) {
         try {
            throw ex; // **
         }
         catch (FirstException e) {
            System.out.print(ex.getMessage());
          }
      }
  }
}
```

א- יש שגיאת קומפילציה בשורה המסומנת ב *

ב- יש שגיאת קומפילציה בשורה המסומנת ב **

ג- התוכנית תעוף על שגיאת זמן ריצה ולא תדפיס כלום.

ד- התוכנית תדפיס 123 ולאחר מכן תעוף על שגיאת זמן ריצה.

ה- התוכנית תדפיס 12 ולאחר מכן תעוף על שגיאת זמן ריצה.

ו- התוכנית תדפיס 112 ולאחר מכן תעוף על שגיאת זמן ריצה.

ז- התוכנית תדפיס 1123 ולאחר מכן תעוף על שגיאת זמן ריצה.

ח- התוכנית תסיים בהצלחה ללא שום הדפסה.

streams

התבוננ/י בקוד הבא:

```
public class Test{
    public static void main(String[] args){
        Stream<Integer> s = Stream.generate(new NaturalNumbers());
        Optional<Integer> mV =
                    s.filter(x-> {System.out.println(x);
                                  return x <= 10;})
                    .max((x,y)-> Integer.compare(x, y));
        System.out.println(mV.isPresent() ? mV.get() : "no max value");
    }
}
```

```
Optional<T> max(Comparator<? super T> comparator)

Return the maximum element of this stream according to
the provided Comparator.
```

```
class NaturalNumbers implements Supplier<Integer> {
    private int i;
    @Override
    public Integer get() {
        return ++i;
    }
}
```

בחר/י את התשובה הטובה ביותר:

א- התוכנית תסתיים ובסיומה יודפס 10

ב- התוכנית תסתיים ובסיומה יודפס הערך של Integer.MAX_VALUE (הערך המקסימלי ל int).

ג- התוכנית תסתיים ובסיומה יודפס no max value

ד- התוכנית תיכנס ללולאה אינסופית ולא יודפס שום פלט במהלך ריצתה.

ה- התוכנית תסתיים ולא יודפס שום פלט במהלך ריצתה.

ו- מלבד תשובה זו, כל התשובות לא נכונות.

בהצלחה!