



תכנות מונחה עצמים תשע"ט - בחינת סיכום מועד א'

1/07/2019

- את התשובות יש לכתוב אך ורק במחברת הבחינה!!!
- בבחינה 4 שאלות.
- משך הבחינה 3 וחצי שעות.
- יש לפתור את כל השאלות.
- בחינה בחומר פתוח.
- **מרצים:** ד"ר תמיר לוי, ד"ר קרן כהן, מר ירון מזרחי.

הערות כלליות לפותר

- ניתן לקצר את הכתיבה של `System.out.println` ל- `Syso`.
- אין צורך להתחשבולקחת בחשבון\לכתוב משפטי `import` – פשוט תניחו שכל חבילה נחוצה כבר מיובאת (כדי לפשט ולקצר את הקוד שתצטרכו לקרוא).

בהצלחה!!!

1.1 נתונה התוכנית הבאה שעושה שימוש במחלקות Person ו-Teacher שגם מתוארות בהמשך. ציינו את פלט התוכנית שמודפס למסך בזמן ריצת התוכנית.

## Program.java

```
public class Program {
    public static void main(String[] args) {
        List<Person> people = new ArrayList<Person>();
        Teacher t = new Teacher();
        people.add(t);
        people.add(new Person(t));
        people.add(t);
        Person q = new Person( "AVI AVINU" , 999);
        for(Object o : people) System.out.println( o.equals(q) );
    }
}
```

## Teacher.java

```
public class Teacher extends Person {

    protected String course;

    public Teacher() {
        super("Avi avinu", 1);
        System.out.println("Teacher()");
        course = "Programming B";
    }

    public String toString() {
        return "<" + super.toString() + ", course=" + course + ">";
    }

    public boolean equals(Object obj) {
        try {
            return ((Teacher)obj).id == id;
        }
        catch (RuntimeException e) {
            System.out.println("Balagan"); return false;
        }
    }
}
```

## Person.java

```
public class Person
{
    protected String name;
    protected int id;

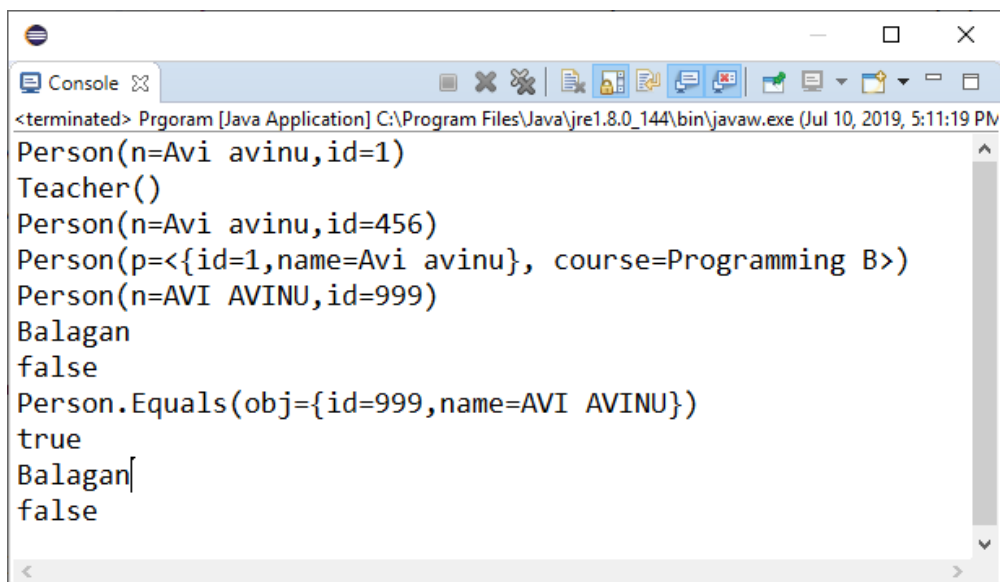
    public Person(String n, int id) {
        System.out.println("Person(n=" + n + ",id=" + id + ")");
        name = n; this.id = id;
    }

    public Person(Person p) {
        this(p.name, 456);
        System.out.println("Person(p=" + p + ")");
    }

    public String toString() {
        return "{id=" + id + ",name=" + name + "}";
    }

    public boolean equals(Object obj) {
        System.out.println("Person.Equals(obj=" + obj + ")");
        if (! obj.getClass().equals(Person.class) ) return false;
        Person p = (Person)obj;
        return p.name.toLowerCase().equals(name.toLowerCase());
    }
    // להזכירכם – השיטה toLowerCase() מחזירה את המחרוזת הנוכחית כאשר כל האותיות
    // הופכות ל-lower case. כך לדוגמא אם String str = "aBcD"
    // אז הקריאה str.toLowerCase() תחזיר את המחרוזת "abcd"
}
}
```

### פתרון שאלה 1.1:



```
<terminated> Prgoram [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Jul 10, 2019, 5:11:19 PM)
Person(n=Avi avinu,id=1)
Teacher()
Person(n=Avi avinu,id=456)
Person(p=<{id=1,name=Avi avinu}, course=Programming B>)
Person(n=AVI AVINU,id=999)
Balagan
false
Person.Equals(obj={id=999,name=AVI AVINU})
true
Balagan[
false
```

1.2) האם התוכנית הבאה תתקמפל? אם כן כתוב מה הפלט שלה? אם לא, הסבר מדוע.

**Program.java (סעיף 1.2)**

```
import java.util.*;
public class Program {
    public static void main(String[] args) {
        HashMap<int , Person > mymap = new HashMap<int , Person >();
        mymap.put(111,null);
        System.out.println(mymap.containsKey(111));
    }
}
```

**פתרון שאלה 1.2:**

השיטה הראשית Main הזו לא תתקמפל כיון שבהגדרת טיפוסים גנריים ניתן להשתמש רק בעצמים ולא בטיפוסים פרמיטיביים (כמו int בדוגמא הזו).

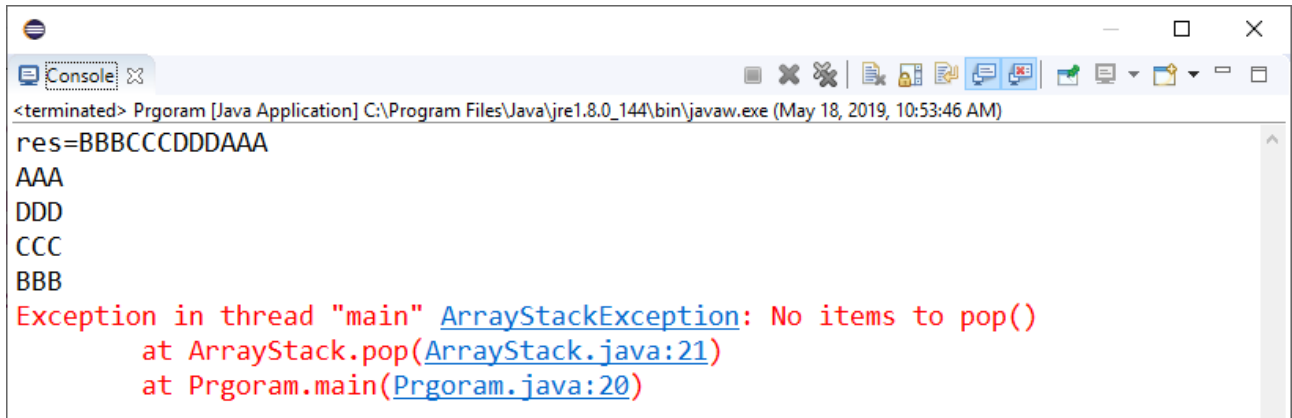
---

## שאלה מס' 2 (20 נקודות)

להלן תוכנית הבנויה מ-4 המחלקות הבאות:

ArrayStack	– מממש מחסנית גנרית בעזרת מערך
ArrayStackException	– מתאר סוג חדש של חריגה
AS_Iterator	– מתאר איטרטור עבור המחלקה ArrayStack
Program	– מכיל את השיטה הראשית main

בהמשך מוצג הקוד של המחלקות הנ"ל. חלק משורות הקוד הוחלפו במלבנים ריקים ממוספרים. עליכם להשלים את הקוד החסר בכל מלבן כך שהפלט של התוכנית יהיה:



```
<terminated> Prgoram [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (May 18, 2019, 10:53:46 AM)
res=BBBCCDDDDAAA
AAA
DDD
CCC
BBB
Exception in thread "main" ArrayStackException: No items to pop()
    at ArrayStack.pop(ArrayStack.java:21)
    at Prgoram.main(Program.java:20)
```

- אין לשנות את שורות הקוד הקיימות
- יכולים להיות מלבנים (0 או יותר) שאמורים להישאר ריקים.
- לדוגמא, עבור ריבוע מספר -0-, התוכן שלו צריך להיות void main.
- לכן עליכם לכתוב במחברת הבחינה את השורה:

void main = 0 ריבוע

עליכם לכתוב במחברת הבחינה באופן דומה את התוכן של יתר המלבנים הממוספרים.

```
public class Program {
    public static (String[] args){
        -0- (פתור בדוגמא למעלה)
        -1- ArrayStack<String> strStack = new ArrayStack<String>();
        strStack.push("BBB");
        strStack.push("CCC");
        strStack.push("DDD");
        strStack.push("AAA");

        String res = "";
        for (String s : strStack){
            res += s;
        }

        System.out.println("res="+res);
        while (true){
            System.out.println(strStack.pop());
        }
    }
}
```

```
public class ArrayStack<T> implements
```

```
Iterable<T> { -2-
```

```
    private T[] items;  
    private int ctr;
```

```
    public ArrayStack(){  
        items = (T[])new Object[5];  
        ctr = 0;  
    }
```

```
    public void push(T t){ -3-  
        if (ctr == items.length){  
            System.out.println("No more room for item:"+t);  
            return;  
        }  
        items[ctr++] = t; -4-  
    }
```

```
    public T pop(){  
        if (0 == ctr){  
            throw new ArrayStackException("No items to pop()"); -5-  
        }  
        return items[--ctr];  
    }
```

```
    public Iterator<T> iterator(){  
        return new AS_Iterator<T>(items,ctr);  
    }
```

```
    public int size() {  
        return ctr;  
    }
```

```
}
```

```
public class ArrayStackException extends RuntimeException { -6-
```

```
    public ArrayStackException(String str){  
        super(str); -7-  
    }
```

```
}
```

```
public class AS_Iterator<T> implements Iterator<T> -8- {  
  
    private int current, top;  
    private T[] items;  
  
    public AS_Iterator(T[] items, int top) {  
        this.items = items; this.top = top;  
    }  
  
    public boolean hasNext() {  
        return current < top; -9-  
    }  
  
    public T next() { -10-  
        return items[current++];  
    }  
  
}
```

---

### שאלה מס' 3 (35 נקודות)

חברת המשלוחים "קח את זה" מבצעת משלוחים של חבילות באמצעות כלי תחבורה אוטונומיים. נשכרת כדי לפתח תוכנית ניהול עבור החברה. בתוכנית זו אובייקט מסוג "משלוח" צריך לשמור את הנתונים הבאים:

- כתובת מקור (מחרוזת) – הכתובת ממנה נאסף המשלוח.
- כתובת יעד (מחרוזת) – הכתובת אליה נשלח המשלוח.
- מספר מזהה של כלי הרכב איתו מבצעים את המשלוח. – יש לציין כי לחברה מספר סוגים של כלי רכב לביצוע המשלוחים אבל לכל אחד מכלי הרכב בחברה ולא משנה מאיזה סוג יש מספר שלם ייחודי שייקרא "המספר המזהה" של כלי הרכב.

החברה מחזיקה את סוגי הכלים הבאים לביצוע המשלוחים, לקוח יכול לבחור באיזה סוג משלוח הוא מעוניין ולשלם בהתאם:

- א. רחפן - מבצע משלוח אווירי. זמן הטיסה נמדד בדקות. עלות משלוח הינה 10 שח לדקת טיסה.
- ב. רובוט - הרובוט הינו כלי תחבורה קטן בעל ארבעה גלגלים וארגז לנשיאת מטען. הרובוט נוסע על המדרכה בדרכו אל היעד. עבור הרובוט שומרים את המרחק בקילומטרים. בנוסף לכתובת המקור וכתובת היעד ניתן להוסיף לרובוט נקודות עצירה ללא מגבלה. כל נקודת עצירה הינה כתובת בה על הרובוט לעבור מטיפוס מחרוזת בדומה לכתובת המקור וכתובת היעד. עלות המשלוח הינה 15 שח לקילומטר ועוד 5 שח לכל נקודת עצירה.
- ג. רכב אוטונומי - הרכב האוטונומי נוסע על הכביש. לרכב האוטונומי שומרים מרחק נסיעה בקילומטרים וזמן נסיעה בדקות. החיוב עבור משלוח ברכב אוטונומי הוא 8 שח לקילומטר בתוספת 5 שח לכל דקת נסיעה.

ברגע שכלי תחבורה מסיים את משימתו מערכת הניהול יוצרת אובייקט המתאים לכלי התחבורה וטוענת את תכונותיו עם פרטי המשלוח לצורך חישוב עלות המשלוח.

3.1 עליך להגדיר את 4 המחלקות הבאות המייצגות משלוחים מסוגים שונים:

- א. Delivery – משלוח כללי
- ב. DroneDelivery – משלוח אווירי באמצעות רחפן.
- ג. RobotDelivery – משלוח באמצעות הרובוט.
- ד. AutonomousDelivery – משלוח באמצעות רכב אוטונומי.

בהגדרת המחלקות יש להתייחס לתכונות המחלקה, יחסים בין מחלקות, מחלקות מופשטות. כמו כן, עליכם להוסיף לכל מחלקה לפחות בנאי אחד שייאפשר לאתחל את כל תכונות המחלקה.

3.2 לכל מחלקה הגדירו שיטה בשם cost המחשבת ומחזירה את עלות המשלוח.

3.3 עליך להגדיר את השיטה equals להשוואה בין שני משלוחים. לפני ההשוואה על השיטה עצמה לבדוק את תקינות הפראמטר שהית מקבלת. שני משלוחים נחשבים זהים אם הם בוצעו ע"י אותו כלי תחבורה (הכוונה לאותו "מספר מזהה של כלי הרכב" בשני המשלוחים) וכתובות המקור והיעד שלהן זהות.



```
public abstract class Delivery {
    private int vehicleID;
    private String source, destination;

    public Delivery(int vehicleID, String source, String destination)
    {
        this.vehicleID = vehicleID;
        this.source = source;
        this.destination = destination;
    }

    public abstract int cost();

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null) return false;
        if (getClass() != obj.getClass()) return false;

        Delivery other = (Delivery) obj;
        if(this.vehicleID == other.vehicleID &&
            this.source.equals(other.source) &&
            this.destination.equals(other.destination)) return true;

        return false;
    }
}
```

### פתרון – המחלקה DroneDelivery :

```
public class DroneDelivery extends Delivery {
    private int time;

    public DroneDelivery(int vehicleID ,String source,
                        String destination,int time) {
        super(vehicleID, source, destination);
        this.time = time;
    }

    @Override
    public int cost() {
        return 10 * time;
    }
}
```

### פתרון – המחלקה RobotDelivery :

```
public class RobotDelivery extends Delivery {
    private int distance;
    private List<String> stopPoints;

    // C'tor option 1
    public RobotDelivery(int vehicleID , String source,
                        String destination, int distance,
                        List<String> stopPoints) {
        super(vehicleID, source, destination);
        this.distance = distance;
        this.stopPoints = stopPoints;
    }

    // C'tor option 2
    public RobotDelivery(int vehicleID, String source,
                        String destination, int distance,
                        String ... stopPoints) {
        super(vehicleID, source, destination);
        this.distance = distance;
        this.stopPoints = Arrays.asList(stopPoints);
    }

    @Override
    public int cost() {
        return 15 * distance + 5 * stopPoints.size();
    }
}
```

### פתרון – המחלקה AutonomousDelivery :

```
public class AutonomousDelivery extends Delivery {
    private int distance;
    private int time;

    public AutonomousDelivery(int vehicleID, String source,
                               String dest, int distance, int time)
    {
        super(vehicleID, source, dest);
        this.distance = distance;
        this.time = time;
    }

    @Override
    public int cost() {
        return 8 * distance + 5 * time;
    }
}
```

3.4 כתוב מחלקה בשם DailyDeliveries. המחלקה תשמור כתכונה את רשימת של כל המשלוחים שנעשו ביום מסוים. הוסף למחלקה את השיטה findTheMostExpensiveDelivery. השיטה תסרוק את המשלוחים השונים שבוצעו באותו יום ותחזיר את המשלוח שעלותו הייתה הגבוהה ביותר.

### פתרון – המחלקה DailyDeliveries :

```
public class DailyDeliveries {
    private List<Delivery> deliveries;

    public Delivery findTheMostExpensiveDelivery()
    {
        Delivery maxDelivery = null;
        int maxCost = 0;
        int cost;

        for(Delivery delivery : deliveries) {
            cost = delivery.cost();
            if(cost < maxCost) {
                cost = maxCost;
                maxDelivery = delivery;
            }
        }

        return maxDelivery;
    }
}
```

---

## שאלה מס' 4 (20 נקודות)

לעם מסוים במדינה קטנה במזרח התיכון נמאס מבחירות חוזרות ונשנות ולכן הוחלט להפוך את הבחירות לאלקטרוניות (שלא ייצטרכו לצאת מהבית כל שלושה חודשים).

עליכם לכתוב שיטה סטטית בשם `results` :

שמקבלת כפרמטר רשימה (`List`) של מחרוזות ומספר שלם `hasima`.

- כל מחרוזת ברשימה מייצגת הצבעה של אזרח. (רצף אותיות כלשהו שמייצג את אותה המפלגה. שימו לב כי האזרחים יכולים לכתוב את שם המפלגה באותיות קטנות (lower case) או גדולות (upper case) או גם וגם וזה תקין ויש לקחת זאת בחשבון).
- המספר `hasima` מייצג את מספר הקולות הנדרש כדי להיות זכאי למושב בكنסת.

על השיטה לבצע את הסעיפים הבאים לפי הסדר:

- להדפיס את רצפי האותיות של כל המפלגות שמספר ההצבעות עבורן לא עבר את המספר `hasima`. אין דרישות לגבי סדר ההדפסה \ מבנה ההדפסה – רק שכל שם יופיע פעם אחת בלבד.
- אם אף מפלגה לא קיבלה לפחות `hasima` קולות – על השיטה להחזיר ערך 0 ולא לבצע את ההמשך. אחרת (ישנה לפחות מפלגה אחת שקיבלה לפחות `hasima` קולות) :

(a) להדפיס לכל מפלגה שכן עברה את `hasima` – את מספר ההצבעות עבורה. יש לבצע

את ההדפסות לפי סדר אלפביתי של אותיות המפלגה.

(b) להדפיס את רצף האותיות של המפלגה הגדולה ביותר.

(c) על השיטה להחזיר כערך החזרה שלה את מספר ההצבעות שקיבלה המפלגה

הגדולה ביותר.

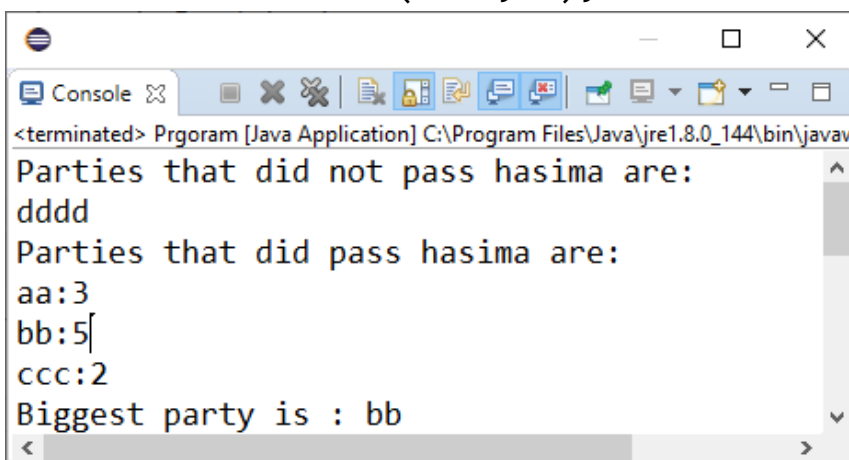
לדוגמא: אם רשימת ההצבעות היא :

```
List<String> votes = Arrays.asList("BB", "BB", "Bb", "aA", "CCC", "aa", "cCc", "aa", "dddd", "bb", "bb");
```

אז הקריאה:

```
int x = results(votes, 2);
```

תדפיס:



```
<terminated> Prgoram [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw
Parties that did not pass hasima are:
dddd
Parties that did pass hasima are:
aa:3
bb:5
ccc:2
Biggest party is : bb
```

ותחזיר לתוך המשתנה `x` את הערך 5. (מספר הקולות של המפלגה שאותיותיה הן `bb` שהיא הגדולה ביותר)

```

public static int results (List<String> votes, int hasima){

    Map<String,Integer> map = new HashMap<String, Integer>();
    for (String v : votes){
        String k = v.toLowerCase();
        if (!map.containsKey(k)) map.put(k, 0);
        map.put(k, map.get(k)+1);
    }

    List<String> toRemove = new ArrayList<String>();
    for (String k : map.keySet()){
        if (map.get(k) < hasima) toRemove.add(k);
    }
    System.out.println("Parties that did not pass hasima are:");
    for (String k : toRemove){
        System.out.println(k);
        map.remove(k);
    }

    if (map.size() == 0) return 0;

    int maxVotes = Collections.max(map.values());
    String biggestParty = null;

    System.out.println("Parties that did pass hasima are:");
    List<String> keys = new ArrayList<String>();
    keys.addAll(map.keySet());
    Collections.sort(keys);
    for (String k : keys){
        System.out.println(k + ":" + map.get(k));
        if (map.get(k) == maxVotes) biggestParty = k;
    }

    System.out.println("Biggest party is : " + biggestParty );

    return map.get(biggestParty);

}

```