NETANYA ACADEMIC COLLEGE
16 Kibutz-Galuyot St.
Kiryat Yitzhak Rabin
NETANYA 42365, ISRAEL

המכללה האקדמית נתניה
רח' קיבוץ גלויות 16
קרית יצחק רבין
נתניה 42365

מבוא לתכנות מונחה עצמים תשע"ז - בחינת סיכום מועד ב'

05/04/2017

- **שימו לב:** את התשובות יש לכתוב אך ורק **במחברת הבחינה!**
- **שימו לב:** יש להקפיד על תיכון (design) בסגנון מונחה עצמים. אין להסתפק ב"תוכנה שעובדת" בלבד!
- משך הבחינה 3 וחצי שעות.
- יש לפתור את כל השאלות.
- בחינה בחומר פתוח.
- בטרם תיגשו לפתרון ודאו כי הטופס מכיל 7 עמודים ו- 4 שאלות.
- מרצים: ד"ר אחמד ג'בארה, גב' קרן כהן, ד"ר תמיר לוי.
- מתרגלים: מר קובי דאבוש

בהצלחה!!!

Fax: (972-9) 8844439 **:פקס** **Tel:** (972-9) 8607777 **:טל**

שאלה מס׳ 1 (20 נקודות)

כתבו שיטה רקורסיבית סטטית בשם findDiff. השיטה תקבל כפרמטרים מערך a של מספרים שלמים ממוינים בסדר לא יורד, גודלו n, ומספר שלם k . השיטה תחזיר אמת אם במערך a קיימים בדיוק k פעמים מספרים שונים זה מזה, אחרת השיטה תחזיר שקר.

הערה: ניתן להניח שאיברי המערך אכן ממוינים בסדר לא יורד.

לדוגמה:

```
public static  boolean  findDiff (int[]  a , int n , int k) {
…
}
public static void main( String[] args ) {
int[]  a = { 1 , 1 , 1 , 2 , 2 , 5 , 5 , 5 } ;
System.out.println ( findDiff ( a , a.length , 3 ) ) ;

יודפס  true  משום שבמערך a קיימים בדיוק 3 מספרים שונים זה מזה//
int[]  b = { 3 , 7 , 7 , 8 , 9 , 9 } ;
System.out.println ( findDiff ( b , b.length ,  3 ) ) ;
יודפס  false  משום שלא קיימים בדיוק 3 מספרים שונים זה מזה //
System.out.println ( findDiff ( b , b.length , 4  ) ) ;
 }
יודפס true  משום שבמערך b קיימים בדיוק 4 מספרים שונים זה מזה//
```

שאלה מס׳ 2 (20 נקודות)

כתבו שיטה סטטית בשם searchChars שמקבלת שלושה פרמטרים: מחרוזת s, מילה word ומספר שלם k.

השיטה תחזיר מחרוזת שמכילה את כל המילים ב s שיש להן בדיוק k תווים משותפים עם המחרוזת word.

כותרת השיטה:

```
public static  String  searchChars  (String s, String word, int k) {…}
```

הערה: ניתן להניח שהמחרוזת s מכילה אותיות קטנות בלבד ובין כל שתי מילים מפריד רווח בודד. המחרוזת word מכילה אותיות קטנות בלבד (ללא רווחים) וללא כפילויות (לא קיים תו במילה שמופיע יותר מפעם אחת).

לדוגמה: אם s="abcd badf eac dba" , word = "ced" , k =2 אז תוחזר המחרוזת "abcd eac" משום שרק למילים אלו יש בדיוק שני תווים משותפים עם המחרוזת word.

שאלה מס' 3 (30 נקודות)

התוכנית הבאה מממשת משחק קלפים שמשוחק בין מספר שחקנים לבין הקזינו. (בכדי לקצר מובאת כאן גרסה מאד חסרה של המשחק) בתוכנית זו הוגדרו המחלקות הבאות: Card, Deck, Game. ענו על הסעיפים הבאים על סמך תוכנית זו (מובאת בהמשך).

3.1 תארו מה מבצעת השיטה unknown_1() שבמחלקה Game. בפרט תארו האם היא שיטת מחלקה או שיטת אוביקט ?, מה היא מבצעת ?, מתי היא מסתיימת ? ואיזה ערך היא מחזירה ?

3.2 תארו מה מבצעת השיטה unknown_2() שבמחלקה Game. בפרט תארו האם היא שיטת מחלקה או שיטת אוביקט ?, מה היא מבצעת ?, מתי היא מסתיימת ? ואיזה ערך היא מחזירה ?

3.3 נסו לתאר את המשחק. בפרט ענו על השאלות הבאות:

א) תארו את התפקיד של המשתנה player_balance ואיך הוא קובע את המשך המשחק.

ב) תארו איך מתנהל כל סיבוב במשחק. (מי מושך קלפים? באיזה סדר? מתי מפסיקים למשוך קלפים ?)

ג) איך מוודאים שחבילת הקלפים לא תסתיים באמצע הסיבוב?

ד) כמה שחקנים משחקים במשחק הנ"ל (חוץ מהקזינו) ומתי מסתיים המשחק.

(<u>נא לא לתרגם לי java לעברית – אלא להסביר זאת במילים שלכם</u>)

```
public class Program {

      public static void main(String[] args) {
            Game g = new Game();
            g.play();
      }

}
```

```
class Card {
  private int value;   // מתאר את הערך על הקלף
  private char suit;   // מתאר את ה"צורה" של הקלף

  public Card(int value, char suit){
    this.value = value;
    this.suit = suit;
  }
  public int getValue(){return value;}

  // פונקציה שמחזירה את הייצוג של העצם כמחרוזת – לצורכי הדפסה
  public String toString(){
      // לקיצור לא פרטתי לכם את גוף הפונקציה
  }
  // מערך סטטי שמכיל את כל ה"צורות" האפשריות
  // 'D' - Diamond - יהלום
  // 'S' - Spade - פיק
  // 'H' - Hart - לב
  // 'C' - Club - תלתן
  // 'J' - Joker – ג'וקר
  public static char[] suits = {'D','S','H','C', 'J'};
```

```java
class Deck {
  private Card[] cards;
  private int deckStart;
  private int deckEnd;
  private int deckSize;

  public int getDeckSize() {
    return deckSize ;
  }

  public Deck (int size){
    cards = new Card[size];
    deckSize=deckStart=deckEnd=0;
  }

  public boolean add(Card c){
    if (isFull()) return false;
    cards[deckEnd] = c;
    deckEnd++;
    deckEnd %= cards.length;
    deckSize++;
    return true;
  }

  public Card draw(){
    if (deckSize == 0) return null;
    Card c = cards[deckStart];
    deckStart++;
    deckStart %= cards.length;
    deckSize--;
    return c;
  }
}
```

```java
public boolean isFull(){
  return deckSize == cards.length;
}

public Deck shuffle (){
  //  – מערבבת את חבילת הקלפים
  //  למען קיצור לא  פירטתי את גוף הפונקציה
}

public static Deck GetFullDeck (){
  Deck result = new Deck(52);
  for (int v = 1 ; v <= 13 ; v++){
    for (int s = 0 ; s < 4 ;s++)
    {
      result.Add(new Card(v,Card.suits[s]));
    }
  }
  return result;
}
```

```java
import java.util.*;
public class Game {

   public final int round_cost = 10;

  private Deck deck;
  int player_balance;

   public Game(){
      player_balance = 100;
   }

   public void play(){
     while(player_balance > 0){
        deck = Deck.GetFullDeck().shuffle();
        System.out.println("balance="+player_balance);
        play_1_round();
     System.out.println("Game finished - player bust");
  }

  private int unknown_1(){
    int sum = 0;
    while (true)
    {
       Card c = deck.Draw();
       System.out.println("draw card :" + c.toString());
       sum +=  c.getValue();
       System.out.println("sum=" + sum);
       if (sum > 21) break;
       System.out.println("Take more cards ? (y/n)");
       String str = new Scanner(System.in).next(); // קליטת מחרוזת מהמשתמש
       if (!str.equals("y")) break;
      }
      return sum;
  }

  // המחלקה Game ממשיכה גם בעמוד הבא
```

```java
private int unknown_2(){
     int sum = 0;
     while (sum < 17)
     {
           Card c = deck.Draw();
           System.out.println("draw card :" + c.toString());
           sum +=  c.getValue();
           System.out.println("sum=" + sum);
     }
     return sum;
}

private void play_1_round(){
     player_balance -= round_cost;

     int playerSum = unknown_1();
     if ( playerSum > 21){
           System.out.println("Player looses, Casino wins");
           return;
     }
     int bankSum = unknown_2();
     if (bankSum > 21 || playerSum > bankSum){
           System.out.println("Player wins, Casino looses");
           player_balance += 2 * round_cost;
     }
     else if (playerSum < bankSum) {
           System.out.println("player looses, Casino wins");
     }
     else {
           System.out.println("Draw (YAANI תיקו)");
           player_balance += round_cost;
     }

}
```

שאלה מס' 4 (30 נקודות)

המחלקה Player מייצגת שחקן בקבוצת כדורגל אשר מאופיין על ידי שם השחקן, רמת השחקן (מספר בין 1-10) וגיל השחקן.

המחלקה Team מייצגת קבוצת שחקנים אשר מאופיינת על ידי שם הקבוצה ואוסף שחקניה. מספר השחקנים המקסימלי בקבוצה אינו יודע מראש ונקבע בעת הקמת הקבוצה.

4.1 הגדירו את המחלקות Player, Team. בסעיף זה עליכם להתייחס לתכונות ובנאים ולהגדירם במידת הצורך.

4.2 כתבו שיטה בשם addPlayer המקבלת כפרמטרים פרטי שחקן ומוסיפה אותו לרשימת השחקנים בקבוצה. השחקן לא יתווסף אם קיים שחקן בקבוצה עם אותו שם. בנוסף, אם אין מקום לעוד שחקן בקבוצה יש להחליף את השחקן החדש בשחקן בקבוצה שהרמה שלו היא הנמוכה ביותר בקבוצה (אם יש כמה כאלה בוחרים אחד מהם) בתנאי שרמתו של השחקן החדש גבוהה מהרמה הנמוכה ביותר בקבוצה.

4.3 מתוך שחקני הקבוצה מעוניינים לבחור **קבוצה** של n שחקנים שישמשו כהרכב למשחק הבא. קריטריון הבחירה הוא שחקנים שרמתם הגבוהה ביותר. אם קיימים מספר שחקנים עם אותה רמה ואין מספיק מקום לכולם בקבוצה הנבחרת, בוחרים את הצעירים יותר. במידה וקיימים כמה שחקנים באותו הגיל אין חשיבות במי בוחרים. כתבו שיטה שמקבלת n כפרמטר ומחזירה קבוצה שתכיל n שחקנים שנבחרו על פי הקריטריונים שתוארו בשאלה.