תשובות לפרק OOP במבחן הבגרות 2007 של מורה מריאנובסקי אלה.

שאלה 17.

א.

1. מחלקה אשר מייצגת סקר אחד:

```
public class OneOpinionPoll{
   private String date;        //  תאריך פרסום של סקר בפורמט: dd/mm/yyyy
   private String question;  // שאלת הסקר
   private String answers=new String[4]; //    תשובות לשאלת סקר
   private int numberOfAnswers=new int[4]; // מערך אשר כל איבר בו מהווה כמות
                                            משתתפים שבחרו בתשובה מסויימת //
   ...
}
```

2.מחלקה אשר מייצגת את 50 הסקרים האחרונים:

```
public  class FiftyOpinionPolls{
   private OneOpinionPoll  opinionPolls[]=new OneOpinionPoll [50];
                                          מערך של 50 הסקרים האחרונים  //
   private int indexOfNewestOpinionPoll  ;
                                          אינדקס הסקר החדש ביותר בין 50 הסקרים //
   ...
}
```

הערה:

בניסוח התרגיל לא ביקשו לטפל במקרה בו עוד לא נצברו 50 סקרים, לכן מניחים שהנתונים של 50 הסקרים האחרונים כבר אוכסנו. לטיפול ב-50 הסקרים חשוב לדעת את המיקומים של הסקר הישן והסקר החדש במערך.אם indexOfNewestOpinionPoll שווה ל-49 אז הסקר הישן שהואכסן במערך נמצא במיקום 0, אחרת מחשבים את מיקומו של הסקר הישן במערך לפי הנוסחה:

indexOfNewestOpinionPoll + 1

ב.

הערה:

ב-OOP קלט מהמקלדת זה עניין של התוכנית הראשית. שיטות במחלקה מקבלות את הנתונים הנדרשים דרך הפרמטרים. לכן התייחסתי לבקשה "לקלוט" בניסוח של התרגיל כאילו זו בקשה "לקבל כפרמטר".

1. במחלקה **OneOpinionPoll:**

```
 /**
   השיטה מקבלת את מספר התשובה של משתתף בסקר (1-4) כפרמטר ומגדילה ב-1 את*
   כמות  המשתתפים אשר בחרו בתשובה הזאת                                    *
   */

public void incNumberOfAnswers( int i);
```

2. במחלקה **FiftyOpinionPolls:**

```
   /**
   השיטה מקבלת את מספר התשובה של משתתף בסקר האחרון כפרמטר ומגדילה ב-1 *
   את כמות  המשתתפים אשר בחרו בתשובה הזאת בסקר האחרון                    *
   */

 public void incNumberOfAnswers( int i);
```

```
/**
 *                    השיטה מקבלת את תאריך פרסום הסקר  date  בפורמט dd/mm/yyyy ,
 *                    שאלת הסקר  q  ומערך של 4 תשובות הסקר m כפרמטרים .
 *   השיטה מחליפה את הסקר הישן ביותר בסקר חדש בעל הנתונים המתקבלים עם 0 בוחרי
 *                    תשובות עבור כל אחת מ-4 התשובות הנתונות.
 *                    בנוסף השיטה מעדכנת את ערך האינדקס של הסקר החדש ביותר.
 */

        public void UpdateOpinionPoll( String date, String q, String m[]);

/**
 *   השיטה מדפיסה את השאלה והתשובות של הסקר החדש
 */
public void printNewestOpinionPoll();

/**
*                    השיטה מקבלת  תאריך  date  בפורמט dd/mm/yyyy
*  ומדפיסה את שאלת הסקר אשר פורסם בתאריך הזה, מספר המשתתפים שבחרו בכל אחת
*          מהתשובות ומספר התשובה שנבחרה על ידי המספר הגדול ביותר של המשתתפים  .
*/
public void  printOpinionPoll ( String date);
```

ג. **במחלקה FiftyOpinionPolls:**

```
public int numOfOpinionPolls(){
   int counter=0;
   for(int i=0; i<50; i++)
    {
         int num= opinionPolls[i].sum();
         if(num>1000)
           counter++;
     }
    return counter;
}
```

השיטה ***numOfOpinionPolls*** *משתמשת בשיטה* ***sum*** *מהמחלקה* ***OneOpinionPoll***

```
/**
*   השיטה מחזירה את כמות המשתתפים אשר בחרו בתשובות של הסקר הנוכחי.
 */
public int sum(){
  int help=0;
  for(int i=0; i<4; i++)
     help+= numberOfAnswers[i];
  return help;
}
```

שאלה 18.

א.

```
public class Painting( String creator,
                       String name,
                       int year,
                       double height,
                       double width,
                       String style){
     super(creator, name, year, height, width);
     this.style=style;
}
```

ב.

דריסה (overriding) – הגדרה מחדש שיטה כלשהי בתוך מחלקה יורשת.

ג .

```
public double exhibitionSpace(){
  return super.exhibitionSpace()*this.depth;
}
```

ד.

```
public class Room{
  private Exhibit items[]=new  Exhibit[25];  // מערך מקומות ל-25 מוצגים
  private int num;                           // כמות המוצגים בחדר
  public int numOfPaints(){
    int counter=0;
    for(int i=0; i<num; i++)
       if( items[i] instanceof Painting)
           counter++;
    return counter;
  }
  public void addItem( Exhibit ex )
  {
    items[num]=ex;
    num++;
  }
   ...
}// Room  סוף מחלקה
```

ה.

```
public class Exhibition{
    private Room rooms[]=new Room[10];  //חדרי התערוכה

    public int numOfPaints (int i) {
        return rooms[i-1].numOfPaints ();
    }

    public void addItem( int i, Exhibit ex){
       rooms[i-1].addItem (ex);
    }
     ...
}// Exhibition  סוף מחלקה
```

שאלה 19.

א.

i

```
/**
  * המחלקה מייצגת טלפון בסיסי
  */
public class BasePhone
```

ii

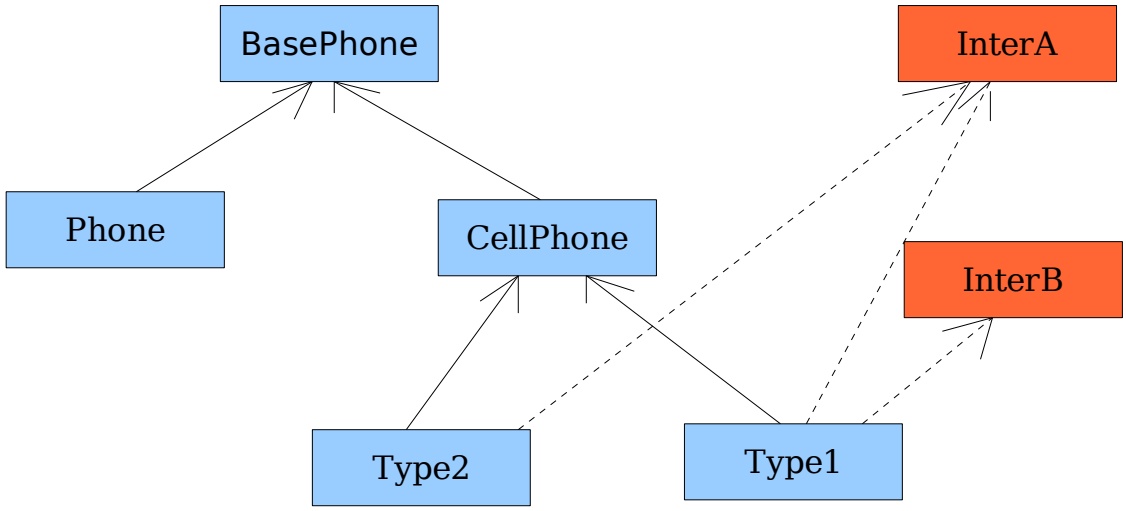
<u>תכונות:</u>

```
String numberOfPhone; // מספר טלפון
int status;  // 1- שיחה נכנסת
             // 2- שיחה יוצאת
             // 3- מצלצל
             // 4- חיוג
             // פנוי
```

<u>פעולות:</u>

- חיוג
- קבלת שיחה

ב.



המחלקות **Phone** ו-**CellPhone** יורשים מהמחלקה **BasePhone**.

ג.

i

```
public class Phone extends BasePhone
public class CellPhone extends BasePhone
public class Type1 extends CellPhone implements InterA, InterB
public class Type2 extends CellPhone implements InterA
```

מחלקה **Phone**:

איננו רואה שום תכונה ושום פעולה שצריך להוסיף למחלקה **Phone**. כל האיברים אשר היא צריכה להכיל כבר נמצאים במחלקה **BasePhone**.

מחלקה **CellPhone**:

<u>תכונה:</u>

- ווקטור מספרי טלפון

<u>פעולות:</u>

- הוספת מספר
- מחיקת מספר

מחלקה **Type1**:

<u>פעולות:</u>

- הצגת שעון
- כיוון שעון
- צילום תמונה
- הצגת תמונה

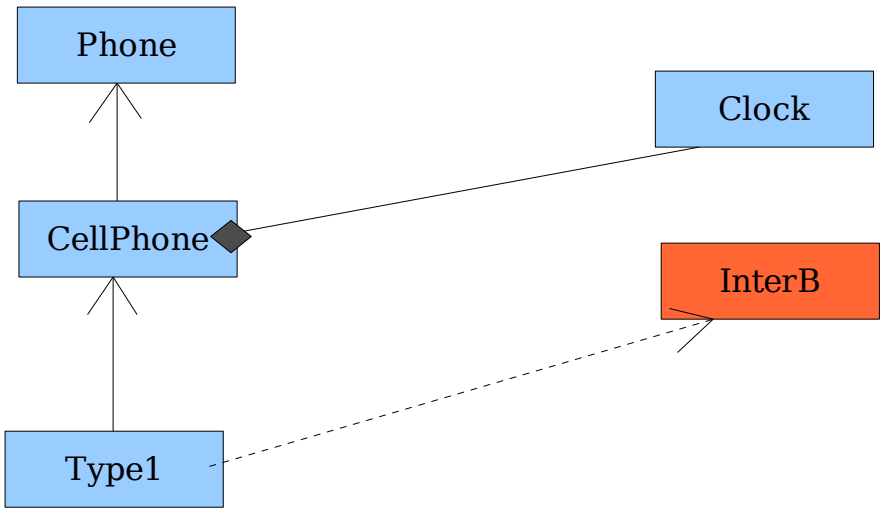
מחלקה **Type2**:

<u>פעולות:</u>

- הצגת שעון
- כיוון שעון

<u>הערה:</u>

בסעיף א' דרשו להוסיף מחלקה חדשה. אני לא רואה צורך להוסיף מחלקה חדשה. אני גם חושבת שהממשק InterA חייב להיות מחלקה. הנימוק הוא: אם יש פעולות לכיוון שעון, אזי יש צורך לשמור את נתוני השעון בתכונות . בהתאם לסיפור בתרגיל צריך לסדר את המחלקות לפי ה-**UML** הבא:



המחלקה **CellPhone** מכילה reference של עצם מטיפוס **Clock** ויורשת מהמחלקה **Phone**.

המחלקה **Type1** מממשת את הממשק **InterB**.

מחלקה **Phone**:

תכונות:

- *String* numberOfPhone; // מספר טלפון
- *int* status; // 1- שיחה נכנסת
  // 2- שיחה יוצאת
  // 3- מצלצל
  // 4- חיוג

פעולות:

- חיוג
- קבלת שיחה

מחלקה **CellPhone**:

תכונות:

- ווקטור מספרי טלפון
- שעון

פעולות:

- הוספת מספר
- מחיקת מספר

מחלקה **Clock**:

תכונות:

- שעות
- דקות
- שניות

פעולות:

- הצגת שעון
- כיוון שעון

מחלקה **Type1**:

פעולות:

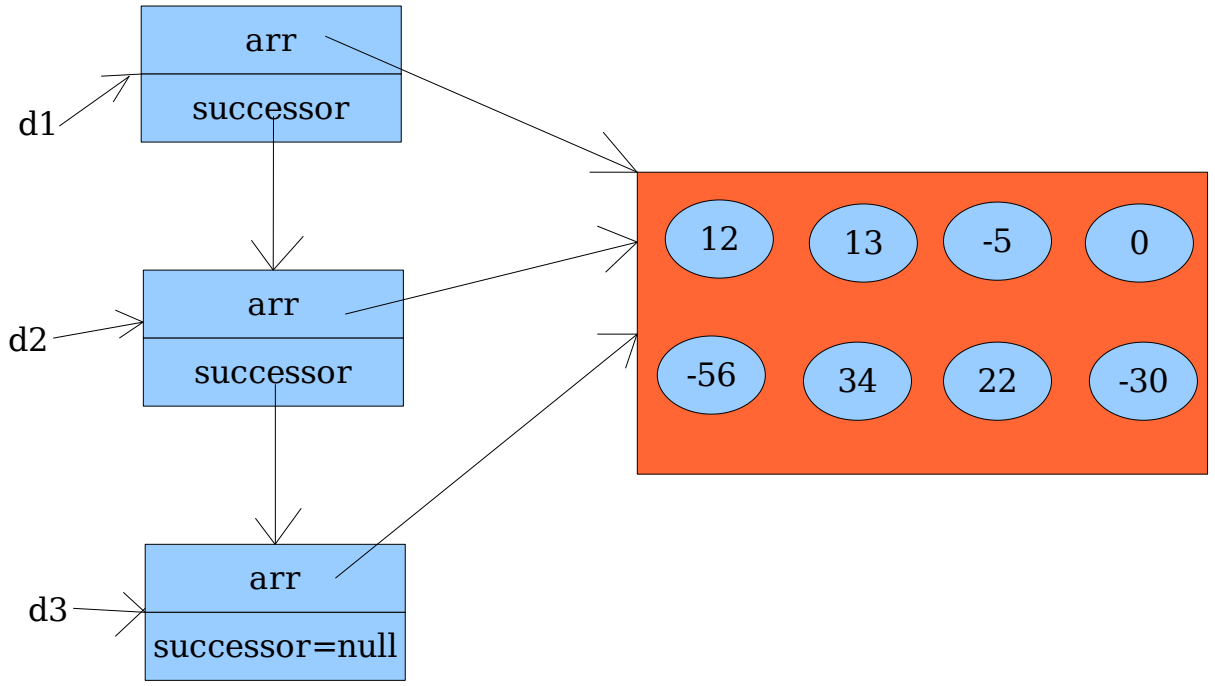
- צילום תמונה
- הצגת תמונה

ד.

```
public class Service{
   private ArrayList v;   // מאגר דינמי של מכשירי טלפון
   ...
}
```

שאלה 20.

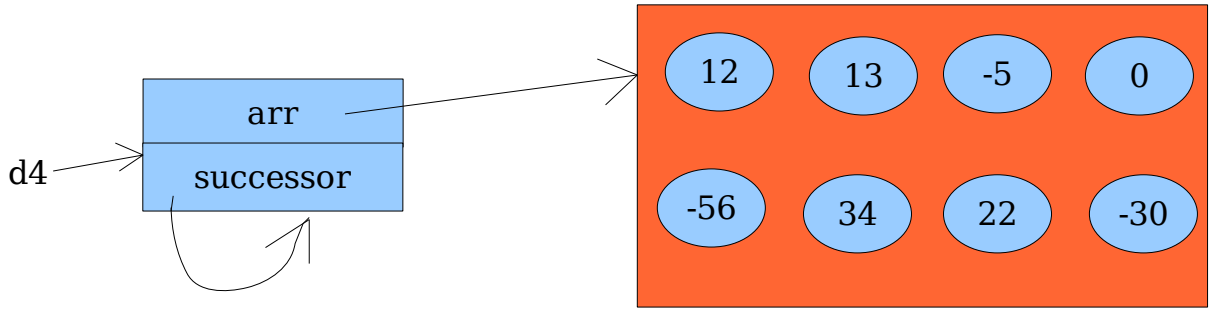
א.



הנה ציור העצמים אשר נוצרו בתהליך ביצוע התוכנית.

הפלט:

```
Sum:81
Counter:4
12
13
-5
*
-56
34
22
-30
Counter: 6
```

ב.

הפעם התוכנית אינה מסתיימת מפני שה-successor לעולם לא יקבל ערך null.