

מדעי המחשב

פרק ראשון

שאלה 1

Java

פתרון 1: ריצה מהסוף לבתחלה

```
//--- פעולה המחזירה את המספר האי-זוגי האחרון ---
//--- הנחה: קיים לפחות מספר אי-זוגי אחד במערך ---
public int lastOddNum()
{
    int i = this.arrayNum.length - 1;
    while (i >= 0 && this.arrayNum[i] % 2 == 0)
        i--;
    return this.arrayNum[i];
}
```

פתרון 2: ריצה מהתחלה לסוף. בסיום הריצה יכיל num את המספר האי-זוגי האחרון במערך

```
//--- פעולה המחזירה את המספר האי-זוגי האחרון ---
//--- הנחה: קיים לפחות מספר אי-זוגי אחד במערך ---
public int lastOddNum()
{
    int num = 0;
    for (int i = 0 ; i < this.arrayNum.length ; i++)
        if (this.arrayNum[i] % 2 != 0)
            num = this.arrayNum[i];
    return num;
}
```

C#

```
//----- שאלה 1 דרך 1 -----
//--- פעולה המחזירה את המספר האי-זוגי האחרון ---
//--- הנחה: קיים לפחות מספר אי-זוגי אחד במערך ---
public int LastOdd()
{
    int place = 0;
    for (int i = 0 ; i < arrayNum.Length ; i++)
        if (arrayNum[i] % 2 != 0)
            place = i;
    return arrayNum[place];
}

//----- שאלה 1 דרך 2 -----
//--- פעולה המחזירה את המספר האי-זוגי האחרון ---
//--- הנחה: קיים לפחות מספר אי-זוגי אחד במערך ---
public int LastOdd2()
{
    for (int i = arrayNum.Length - 1 ; i >= 0 ; i--)
        if (arrayNum[i] % 2 != 0)
            return arrayNum[i];
    return -1;
}
```

Java

```

//--- פעולה המקבלת את מערך ההצבעות של השופטים ---
//--- ומדפיסה את מספרו של השיר הזוכה ---
public static void theWinner (Vote [] a)
{
    //--- הגדרת מערך הקולות לשירים ואיתחולו ---
    int [] arrScore = new int [41];
    for (int i = 1 ; i < arrScore.length ; i++)
        arrScore[i] = 0;

    for (int i = 0 ; i < a.length ; i++)
    {
        arrScore[a[i].getFirst()] += Vote.fst;
        arrScore[a[i].getSecond()] += Vote.sec;
        arrScore[a[i].getThird()] += Vote.thrd;
    }

    //--- מציאת מקום המקסימום במערך הקולות ---
    int max = 0;
    for (int i = 1 ; i < arrScore.length ; i++)
        if (arrScore[i] > arrScore[max])
            max = i;

    //--- הדפסת הניקוד שקיבל כל שיר - לא נדרש בבחינה ---
    for (int i = 0 ; i < arrScore.length ; i++)
        System.out.println(i + " - " + arrScore[i]);
    System.out.println();

    //--- הדפסת מספר השיר הזוכה ---
    System.out.println("the winner is song #" + max);
}

```

C#

```
//----- פתרון תרגיל 2 -----
//--- פעולה המקבלת את מערך ההצבעות של השופטים ---
//--- ומדפיסה את מספרו של השיר הזוכה ---
public static void Print(Vote[] arr)
{
    int[] a = new int[41];
    //--- אפוס מערך צוברים ---
    for (int i = 0; i < a.Length; i++)
        a[i] = 0;

    //--- הוספת הניקוד המצטבר למערך הניקוד ---
    for (int i = 0; i < arr.Length; i++)
    {
        int x1 = arr[i].GetFirst();
        int x2 = arr[i].GetSecond();
        int x3 = arr[i].GetThird();

        a[x1] = a[x1] + 7;
        a[x2] = a[x2] + 5;
        a[x3] = a[x3] + 1;
    }

    //--- מציאת המקסימום במערך הניקוד ---
    int max=0,place=0;
    for (int i = 0; i < arr.Length; i++)
    {
        if (a[i] > max)
        {
            max = a[i];
            place = i;
        }
    }
    Console.WriteLine(place);
}
```

שאלה 3

```
public class Time
{
    //--- תכונות ---
    private int hour;           // שעה. בין 0 - 23 כולל
    private int minute;        // דקה. בין 0 - 59 כולל

    //--- פעולה בונה ---
    public Time(int hour, int minute)
    {
        //--- אם הערכים לא מתאימים שים בהם 0 ---
        if (hour < 0 || hour > 23)
            hour = 0;
        if (minute < 0 || minute > 59)
            minute = 0;

        this.hour = hour;
        this.minute = minute;
    }

    //--- בנאי מעתיק ----
    public Time(Time t)
    {
        this.hour = t.hour;
        this.minute = t.minute;
    }
}

//----- סעיף ב' -----
public class Flight
{
    //--- תכונות ---
    private String name;        // שם חברת התעופה
    private String destination; // יעד הטיסה
    private String flightCode; // קוד טיסה
    private Time flightTime;    // זמן הטיסה

    //--- פעולה בונה ---
    public Flight(String name, String destination, String flightCode, Time flightTime)
    {
        this.name = name;
        this.destination = destination;
        this.flightCode = flightCode;
        this.flightTime = flightTime;
        // שימוש בבנאי מעתיק // this.flightTime = new Time (flightTime);
    }

    public String getName()
    {
        return name;
    }
}
```

כשעובדים עם עצם מסוג תאריך או שעה, עדיף להפעיל בנאי מעתיק

ב- C# string (משתנה) ולא String (עצם)

- (1) הפעולה תחזיר עבור הערך הראשון במערך (שערכו Cloud ולא Sky) ערך שקר. הפעולה לא תגיע לבדוק את ערכו של הערך השני במערך.
- (2) השגיאה: החזרת הערך צריכה להתבצע מחוץ ללולאה (אם סיימנו לעבור על כל המערך ולא מצאנו טיסה של חברת Sky נחזיר שקר)
- (3) הפעולה המתוקנת:

```
//----- סעיף 1 - פעולה שגויה -----
public boolean isFly()
{
    for (int i = 0 ; i < flights.length ; i++)
    {
        if (this.flights[i].getName().equals ("Sky"))
            return true;
        return false;
    }
    return false;
}
```

פרק שני

שאלה 4

```
//----- סעיף א -----
//--- פעולה המקבלת כפרמטר מחסנית של מספרים שלמים ---
//--- ומחזירה את המספר שבתחתית המחסנית ---
//--- הנחה: מספר האיברים במחסנית הוא זוגי ---
public static int lastAndRemove (Stack<Integer> stk)
{
    Stack<Integer> sTemp = new Stack<Integer>();

    //--- העברת כל האיברים למחסנית העזר ---
    while (! stk.isEmpty())
        sTemp.push(stk.pop());

    //--- (crta njxbh, vgzr --- שליפת האיבר האחרון) ---
    int x = sTemp.pop();
        /*
        int x = -1;
        if (! sTemp.isEmpty())
            x = sTemp.pop();
        */

    //--- חזרת האיברים למחסנית ---
    while (! sTemp.isEmpty())
        stk.push(sTemp.pop());

    return x;
}

//----- סעיף ב -----
//--- פעולה המקבלת כפרמטר מחסנית של מספרים שלמים ---
//--- ומחזירה מחסנית של זוגות איברים: ראשון ---
//--- ואחרון במחסנית. הפעולה הורסת את המחסנית המקורית ---
public static Stack<TwoItem> stackTwoItem (Stack<Integer> stk)
{
    Stack<TwoItem> s1 = new Stack<TwoItem>();

    int n1, n2;
    while (! stk.isEmpty())
    {
        n1 = lastAndRemove(stk);
        n2 = stk.pop();
        s1.push(new TwoItem(n2, n1));

        //--- ואפשר גם בשורה אחת ---
        // s1.push(new TwoItem(stk.pop(), lastAndRemove(stk)));
    }
    return s1;
}
```

```
//----- פתרון תרגיל 4 סעיף א -----
public static int LastAndRemove(Stack<int> s)
{
    Stack<int> s2 = new Stack<int>();
    int x;
    while (!s.IsEmpty())
    {
        x = s.Pop();
        s2.Push(x);
    }
    int y = s2.Pop();
    while (!s2.IsEmpty())
        s.Push(s2.Pop());
    return y;
}

//----- פתרון תרגיל 4 חלק ב -----
public static Stack<TwoItems> BuildStack(Stack<int> s)
{
    Stack<TwoItems> stk = new Stack<TwoItems>();
    while (!s.IsEmpty())
    {
        int x = s.Pop();
        int y = LastAndRemove(s);
        stk.Push(new TwoItems(x, y));
    }
    return stk;
}
```


שאלה 5

ch	הרשימה lst	lst == null	lst value == ch	ערך מוחזר
'v'	↓ ['c', 'd', 'v', 'h']	לא	לא	
	↓ ['c', 'd', 'v', 'h']	לא	לא	
	↓ ['c', 'd', 'v', 'h']	לא	כן	מוחזרת הפנייה לחוליה המכילה את התו 'v'

א. טבלת מעקב (1)

(2) הפעולה מחזירה הפניה לחוליה המכילה את התו ch אם הוא ברשימה, אחרת מחזירה null

(3) סיבוכיות הפעולה הוא $O(n)$.

נימוק: n מייצג את מספר האיברים ברשימה.

הפעולה עוברת לכל היותר פעם אחת על כל הרשימה ומבצעת פעולות בסיבוכיות $O(1)$

ב. הפעולה sod2 מחזירה אמת אם האות 'a' וגם האות 'b' מופיעות ברשימה ושקר אחרת.

```
//----- סעיף 1 -----
//--- פעולה המקבלת כפרמטר רשימה של תווים ומחזירה אמת ---
//--- אם קיימות בה שתי חוליות סמוכות שערךן a ו b, ושקר אחרת ---
public static boolean isContainsAB (Node<Character>lst)
{
    Node<Character>lst1 = sod1 (lst, 'a');
    if (lst1 == null) return false;

    Node<Character>lst2 = sod1 (lst, 'b');
    if (lst2 == null) return false;

    return lst1.getNext() == lst2 || lst2.getNext() == lst1;
}
```

ג. java

```
//----- פתרון שאלה 5 סעיף 1 -----
//--- נכתב ע"י ראמי ג'באלי ---
public static bool Check(Node<char> lst)
{
    Node<char> p1 = Sod1(lst, 'a');
    Node<char> p2 = Sod1(lst, 'b');
    if (p1 != null && p2 != null && p1.GetNext() == p2 )
        return true;
    if (p1 != null && p2 != null && p2.GetNext() == p1)
        return true;
    return false;
}
```

C#

```
//--- פתרון של אביטל EVI גרוינולד, העושה שימוש בפעולה sod2 ---
//--- פעולה המקבלת כפרמטר רשימה של תווים ומחזירה אמת ---
//--- אם קיימות בה שתי חוליות סמוכות שערךן a ו b, ושקר אחרת ---
public static boolean isABorBAexist(Node<Character> lst)
{
    if (!sod2(lst))
        return false;
    Node<Character> lstA = sod1(lst, 'a');
    Node<Character> lstB = sod1(lst, 'b');
    return lstA.getNext() == lstB || lstB.getNext() == lstA;
}
```

שאלה 6

סיבוכיות הפעולה $O(n)$: הפעולה עוברת פעם אחת על כל אחד מ- n הצמתים בעץ

```
//--- הפעולה הנתונה, אין צורך לממש אותה ---
//--- פעולה המחזירה אמת אם כל צמתי העץ גדולים מ- x ושקר אחרת ---
public static boolean lessThanTree (BinNode<Integer> t, int x)
{
    if (t == null)
        return true;
    if (t.getValue() <= x)
        return false;
    return lessThanTree (t.getLeft(),x) && lessThanTree(t.getRight(), x);
}
```

האם כל צמתי העץ $t1$ קטנים מכל אחד מצמתי העץ $t2$:

פתרון 1: סיבוכיות ליניארית:

נבדוק אם הערך הגדול ביותר בעץ $t1$ קטן מכל אחד מצמתי העץ $t2$.
אם כן - כל הצמתים האחרים בהכרח קטנים אף הם ולכן יוחזר ערך אמת
אחרת יוחזר ערך שקר.

```
//--- פעולה המקבלת שני עצים בינאריים לא ריקים ומחזירה אמת ---
//--- אם כל צמתי העץ הראשון קטנים מכל צמתי העץ השני ---
//--- פתרון ביעילות ליניארית  $O(n)$  ---
//--- אם הערך הכי גדול בעץ הראשון קטן מכל צמתי העץ יוחזר אמת ---
public static boolean isSmaller (BinNode<Integer> t1, BinNode<Integer> t2)
{
    return lessThanTree (t2, biggest(t1));
}

//--- פעולה המקבלת עץ בינארי ומחזירה את הערך הגדול ביותר בו ---
public static int biggest (BinNode<Integer> t)
{
    if (t == null) return Integer.MIN_VALUE;
    int bigLeft = biggest (t.getLeft());
    int bigRight = biggest (t.getRight());

    return Math.max(t.getValue(), Math.max(bigLeft, bigRight));
}
```

הפעולה `biggest` עוברת פעם אחת על n צמתי העץ ומחזירה את הערך הגדול ביותר בעץ $O(n)$

הפעולה `isSmaller` מפעילה את הפעולה `lessThanTree` $O(n)$

ופעם אחת את הפעולה `biggest` $O(n)$

$$O(n) \Leftrightarrow f(n) = 2n$$

פתרון 2: סיבוכיות ריבועית

הפעולה עוברת על כל אחד מ- n צמתי העץ $t1$ ובודקת עבורו האם הוא קטן מכל צמתי העץ $t2$ $O(n)$

$$f(n) = n * O(n) \Leftrightarrow O(n^2)$$

פתרון ב- java:

```
//--- פעולה המקבלת שני עצים בינאריים לא ריקים ומחזירה אמת ---
//--- אם כל צמתי העץ הראשון קטנים מכל צמתי העץ השני ---
//--- פתרון ביעילות ריבועית  $O(n^2)$  ---
//--- בדיקת כל הצמתים של העץ הראשון מול אלו של העץ השני ---
public static boolean smaller (BinNode<Integer> t1, BinNode<Integer> t2)
{
    if (t1 == null) return true;
    if (! lessThanTree(t2, t1.getValue() )
        return false;
    return smaller (t1.getLeft(), t2) && smaller (t1.getRight(), t2) ;
}
```

פתרון ב- C#: נכתב ע"י ראמי ג'באלי

```
//--- פתרון שאלה 6 - לא נדרש לממש ---
public static bool LessThanTree(BinNode<int> t, int x)
{
    if (t == null)
        return true;
    if (x >= t.GetValue())
        return false;
    return LessThanTree(t.GetLeft(), x) && LessThanTree(t.GetRight(), x);
}

//--- פתרון שאלה 6 יעילות ריבועית ---
public static bool TreeLessThanTree(BinNode<int> t1, BinNode<int> t2)
{
    if (t1 == null)
        return true;
    int x = t1.GetValue();
    if (LessThanTree(t2, x) == false)
        return false;
    return TreeLessThanTree(t1.GetLeft(), t2) && TreeLessThanTree(t1.GetRight(), t2);
}
```

פרק feי (25 נקודות)

לפניך שאלות מ-4 מסלולים שונים: מערכות מחשב ואסמבלי (שאלות 7-8), מבוא לחקר ביצועים (שאלות 9-10), מודלים חישוביים (שאלות 11-12), תכנות מונחה עצמים (שאלות 13-14). ענה על שאלה אחת מתוך השתיים בפרק שלמדת (25 נקודות).

מערכות מחשב ואסמבלי ענה על שאלה אחת מהשאלות 7-8 (25 נקודות)

הפתרון לפרק זה נכתב ע"י: רונית (מרציאנו) גל-אור

7 אfe (בחירה, 25 נקודות)

א. (1) בצע טבלת מעקב לקטע, עבור $AX = > 0109H$

```

CMP AH,AL
JAE AA
XCHG AH,AL
AA:  CMP AL,AH
      JAE CC
      INC AL
      DEC AH
      JMP AA
CC:  MOV SI,100H
      MOV [SI],AL
    
```

AX	
AH	AL
01H	09H
09H	01H
08H	02H
07H	03H
06H	04H
05H	05H

תמונת הזיכרון

100H	101H	

(2) מה מבצע הקטע עבור מספרים אי זוגיים ב אוגרים AH, AL.

הקטע מתייחס לערכים באוגרים AH, AL כמספרים לר=א מסומנים ומחשב את הממוצע השלם ביניהם. הקטע בודק מי יותר גדול בין אוגרים AH, AL ואם AH יותר קטן הערכים ביניהם יוחלפו. חישוב הממוצע נעשה ע"י הוספת 1 ל AL והורדת 1 ל AH עד שהם משתווים (במידה ושניהם היו אי זוגיים)

ב. עבור כל היגד יש לקבוע אם נכון או לא נכון, אם אינו נכון יש להסביר

(1) שני הקטעים מבצעים אותו הדבר:
 קטע 1: SHL AX,1
 קטע 2: SHR AX,1
 SHL AX,1

תשובה: לא נכון

במקרה ואוגר AX יכיל מספר אי זוגי ו / או שלילי לפני ביצוע הקטעים (כלומר ביט 0 דלוק ו/או ביט מספר 15) הערכים לאחר ביצוע הקטעים לא יהיו זהים.

לדוגמא: עבור $AX = 1000111111110001B$ התוצאה בשני הקטעים לא תהיה זהה ואילו עבור $AX = 0111000011110000B$ התוצאה תהיה זהה

(2) שני הקטעים מבצעים אותו הדבר:
 קטע 1: AND AX,0FEH
 קטע 2: SHR AX,1
 SHL AX,1

תשובה: לא נכון - קטע 1 מאפס את אוגר AH ואת ביט מספר 0 באוגר AL בקטע 2 אוגר AH נשאר וביט 0 באוגר AL מתאפס.

(3) שני הקטעים מבצעים אותו הדבר:
 קטע 1: PUSH AX
 PUSH BX
 POP AX
 POP BX
 קטע 2: XCHG AX,BX

תשובה: נכון

(4) שני הקטעים מבצעים אותו הדבר:
 קטע 1: SUB AX,0
 קטע 2: CMP AX,0
 JNE A1
 JNZ A1

תשובה: נכון

שניהם בודקים אם אוגר $AX = 0$ וקופצים לתווית אם לא, בשני הקטעים ה zf ידלק אם 0

(5) במקטע הנתונים הוגדר מערך:
 ARR DW 50 dup (?)
 הנח שבאוגר BX מאוחסנת כתובת של איבר מסויים במערך ARR
 קטע תוכנית זה מאחסן באוגר AX את האינדקס של אותו איבר

LEA SI,ARR
 SUB BX,SI
 INC BX

תשובה: לא נכון

הקטע לא מתייחס לעובדה שהמערך הוא של מילים לכן החיסור של כתובת המערך מכתובת האיבר לא תיתן אינדקס של התא כיוון שכל תא הוא בעצם 2 בתים, 2 כתובות.

(6) אם הערך של המספר 11101011 שמאוחסן לפי שיטת המשלים ל 2 - ב 8 סיביות הוא (-21) בבסיס 10?

תשובה: נכון

$$0 = 21d + 11101011b$$

$$21d = 00010101b \text{ המשלים ל } 2 \text{ של הערך הנרשם למעלה.}$$

עאפה 8 (בחירה, 25 נקודות)

יש לכתוב תוכנית הבדקת אם מערך של מספרים מכוונים הוא "גלי":
הראשון יותר קטן מהשני השני יותר גדול מהשלישי השלישי קטן מהרביעי וכו'.

```

MOV DH, 1 ; flag Up or Down in the Wave
LEA DI, ARR ; Address of ARR
LEA SI, REZ ; Address of REZ
MOV BL, 0
MOV [SI], BL
MOV CX, LEN-1 ; number of Couples to check
AGAIN:
MOV AH, [DI]
MOV AL, [DI+1]
CMP DH, -1
JE DOWN
CMP AH, AL
JGE SOF ; Not oK
JMP CONT
DOWN:
CMP AH, AL
JLE SOF ; Not oK
CONT:
NEG DH ; Change flag
INC DI
LOOP AGAIN
MOV BL, 1
MOV [SI], BL
JMP SOF
SOF:

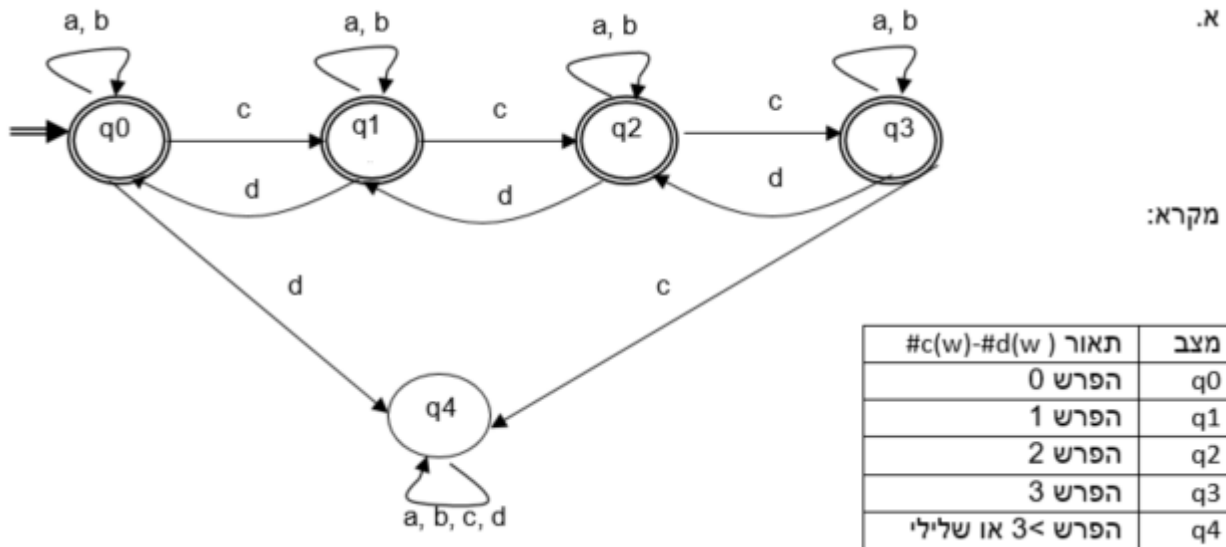
```

פרק פיסי (25 נקודות)

מודלים חישוביים ענה על שאלה אחת מהשאלות 11-12 (25 נקודות)

פתרון פרק זה נכתב ע"י: רחל לודמר

11 אלה (בחירה, 25 נקודות)



ב. 1. $\bar{L}_1 = \Phi$

נתון ש $L_1 = \Sigma^*$, כלומר כל המילים מעל Σ כולל המילה הריק. לכן קבוצת המשלים תהיה הקבוצה הריקה.

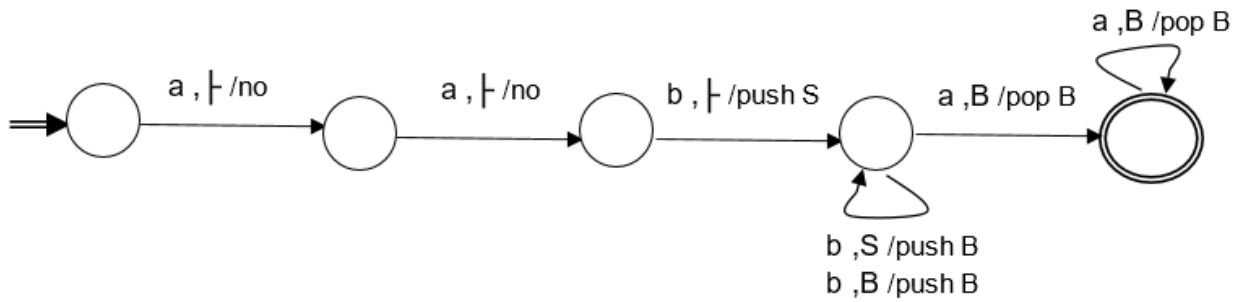
2. $L_3 = L_2 \cap \bar{L}_1 = L_2 \cap \Phi = \Phi$

מאחר והקבוצה הריקה היא שפה רגולרית, גם L_3 היא שפה רגולרית.



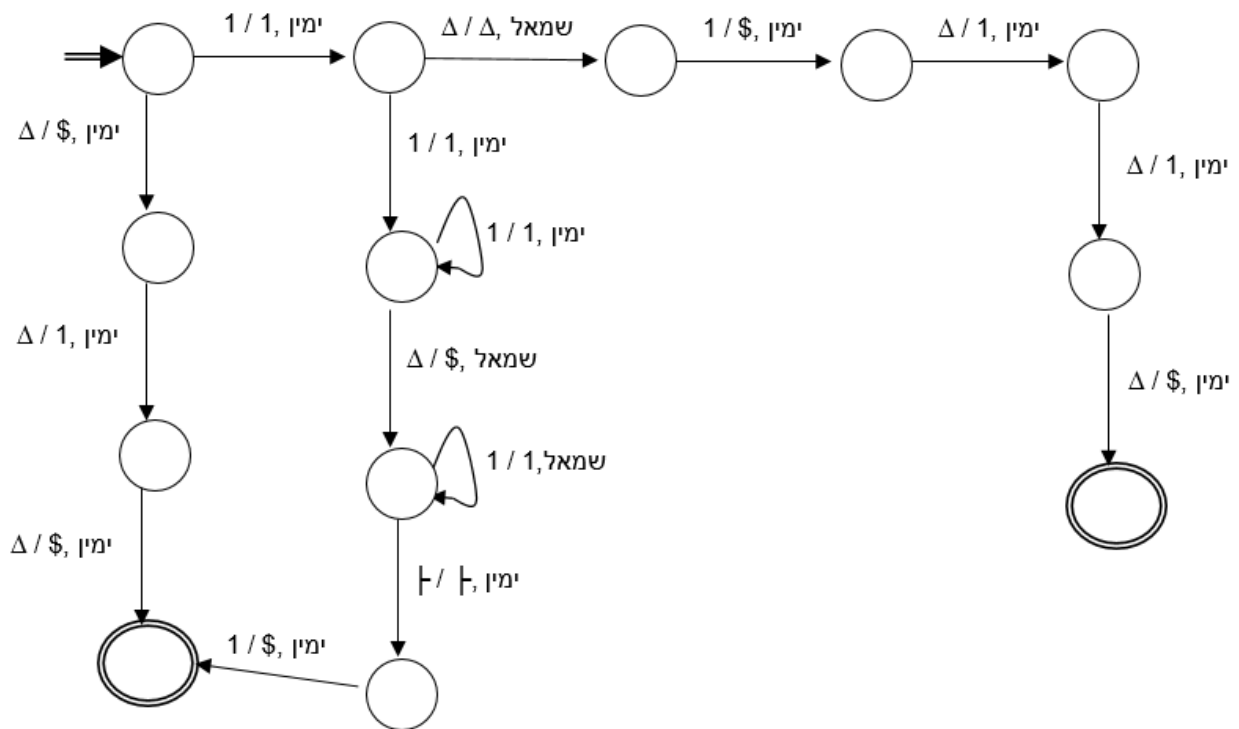
שאלה 12 (בחירה, 25 נקודות)

א.



ב.

נתייחס ל 3 מקרים: עבור $x=0$, $x=1$, $f(x)=x+1$, ועבור $x \geq 2$, $f(x)=x-1$



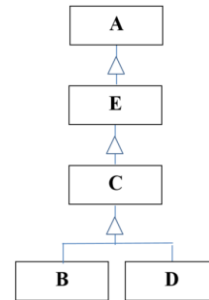
פרק feיפי (25 נקודות)

תכנות מונחה עצמים Java ענה על שאלה אחת מהשאלות 13-14 (25 נקודות)

פתרון פרק זה נכתב ע"י: אביטל EVI גרינוולד

שאלה 13 (בחירה, 25 נקודות)

א- עץ ירושה של כל המחלקות



1. A a1 = new A();
2. A e1 = new E();
3. E c1 = new C();
4. C b1 = new B();
5. C d1 = new D();
6. B d2 = new D(); **Syntax Error**
7. a1.f(); **Syntax Error**
8. ((E)e1).f1(); // prints bye-bye
9. ((B)b1).f1(); // prints hello
10. ((D)b1).f1(); // **Run time error**

שורה זו קובעת עץ הורשה יחיד

הסבר:

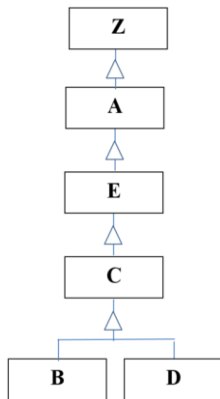
E יורש מ A, על פי הוראה (2)

C יורש מ E על פי הוראה (3)

B ו D יורשות מ C, על פי הוראות (4),(5), אבל הן זרות אחת לשנייה

על פי הוראה (6) והוראה (10)

הפעולה f() מומשה במחלקות E ו-B על פי הוראות (8) ו (9)



ב. את המחלקה Z נוסף כמחלקת על של מחלקה A.

על סמך הוראה Z x = new A()

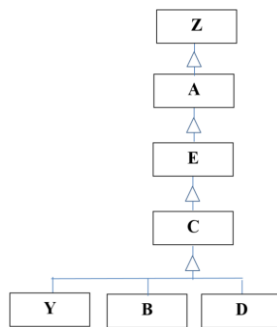
נתון: קוד תקין

A a2 = new A();

Z z1 = new Z();

Z a3 = new A();

i. a2 = z1;	שגיאת תחביר. אין התאמה בטיפוסים. מחלקה Z אינה מכירה את מחלקה A הפניייה מסוג תת-מחלקה אינה יכולה להפנות לעצם מסוג מחלקת העל (ההיפך כן)
ii. a3 = z1;	תקין. z1 נוצר ממחלקה Z ומציבים אותו במשתנה מטיפוס המחלקה Z
iii. ((A)z1).g();	שגיאת זמן ריצה. יש ניסיון להמיר את עצם שנוצר מטיפוס Z לעצם מטיפוס תת-המחלקה A. בזמן ריצה התוכנית תגלה שלא נוצר ממנה.
iv. a2.g();	תקין. a2 נוצר מהמחלקה A אשר יורשת מהמחלקה Z. במחלקה Z קיימת פעולה g(). מחלקת הבת יורשת את כל הפעולות ממחלקת העל ולכן יכולה להפעיל אותה.
v. ((A)a3).g();	תקין. a3 נוצר מטיפוס המחלקה A שהיא מחלקת בת של המחלקה Z לכן ניתן לבצע המרה למחלקה ממנה העצם נוצר. הפעולה g קיימת במחלקה Z ולכן קיימת גם במחלקה היורשת.



ג. את מחלקה Y נוסף כמחלקת בת של המחלקה C. בעיקרון מחלקה Y יכולה להיות גם מ חלקת בת של D או B.

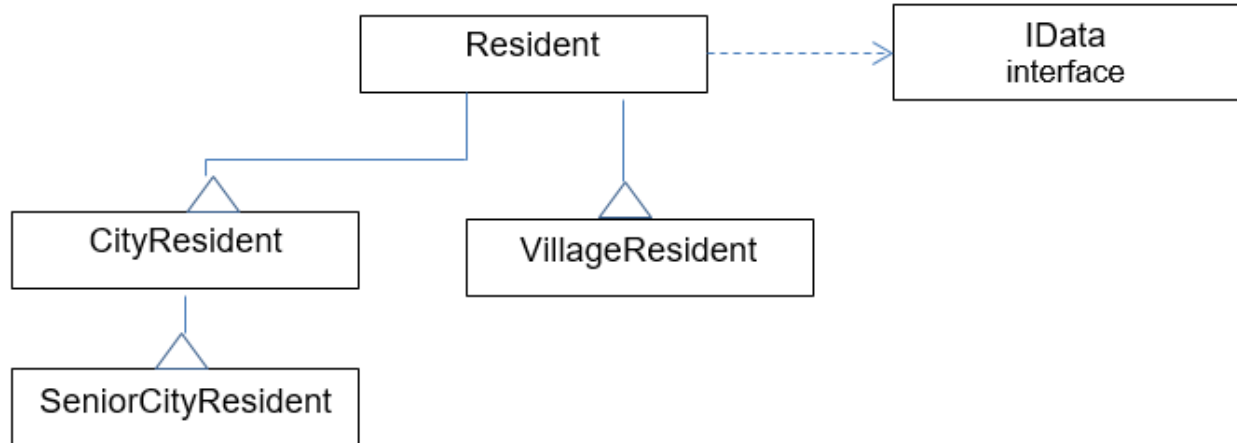
A a2 = new A();
Y y1 = new Y();
C y3 = new T();

נתון: קוד תקין

i. a2 = y1;	תקין. מחלקה Y יורשת ממחלקה C אשר יורשת ממחלקה E אשר יורשת ממחלקה A, לכן ניתן לבצע המרה כלפי מעלה לא מפורשת.
ii. y2 = y1;	תקין. מחלקה Y יורשת ממחלקה C לכן ניתן לבצע המרה כלפי מעלה.
iii. ((A)y1).m();	שגיאת תחביר. למחלקה A אין פעולה כזו. שינינו את נקודת המבט למחלקה A
iv. y2.m();	שגיאת תחביר. נקודת המבט של המחלקה C. למחלקה זו אין פעולה m
v. ((A)y2).m();	שגיאת תחביר. למחלקה A אין פעולה כזו. שינינו את נקודת המבט למחלקה A

שאלה 14 (בחירה, 25 נקודות)

היררכיית המחלקות



א. הגדרת המחלקה Resident

```

public class Resident implements IData
{
    public static final double PAY_M2_APP = 10; // מחיר ארנונה למ"ר שטח למגורים
    public static final double PAY_M2_LAND = 0.5; // מחיר ארנונה למ"ר שטח קרקע נוספת
    protected String name; // שם התושב
    protected double areaLiving; // שטח המיועד למגורים במ"ר
    protected double areaLand; // שטח קרקע נוספת במ"ר
}
    
```

כתיבת הפעולה הבונה (לא בקשו) והפעולה getPropertyTax במחלקה Resident

```

//--- פעולה בונה לתושב (לא נדרש בבחינה) ---
public Resident (String name, double areaLiving, double areaLand)
{
    this.name = name;
    this.areaLiving = areaLiving;
    this.areaLand = areaLand;
}

//--- מחזירה מס ארנונה לתשלום ---
public double getPropertyTax()
{
    return this.areaLiving * Resident.PAY_M2_APP +
           this.areaLand * Resident.PAY_M2_LAND;
}
    
```

ב. הגדרת שלושת המחלקות והפעולה `getPropertyTax()`

```
//--- מחלקת תושב עיר ---
public class CitiResident extends Resident
{
    private static final int GIFT = 250; // מענק בשקלים

    //--- פעולה בונה לתושב העיר (לא נדרש בבחינה) ---
    public CitiResident(String name, double areaLiving, double areaLand)
    {
        super(name, areaLiving, areaLand);
    }
    //--- מחזירה את תשלום מס ארנונה ---
    public double getPropertyTax()
    {
        double price = super.getPropertyTax();
        price = price - GIFT;
        if (price < 0)
            price = 0;
        return price;
    }
}

//--- מחלקת תושב עיר ותיק ---
public class SeniorCityResident extends CitiResident
{
    private static final int SENIOR_AGE = 60; // גיל תושב ותיק
    private static final double EXTRA_PERCENT = 0.01; // אחוז הנחה על כל שנת מעבר ל 60
    private int age; // גיל התושב

    //--- פעולה בונה לתושב עיר ותיק (לא נדרש בבחינה) ---
    public SeniorCityResident(String name, double areaLiving, double areaLand, int age)
    {
        super(name, areaLiving, areaLand);
        this.age = age;
    }
    //--- מחזירה את תשלום מס ארנונה ---
    public double getPropertyTax()
    {
        double price = super.getPropertyTax();
        int diffYear = this.age - SENIOR_AGE; // הפרש השנים מעל 60
        if (diffYear > 0) // הוספת הנחה על כל שנה מעל גיל 60
            price = price - diffYear * EXTRA_PERCENT * price;
        return price;
    }
}
```

ענה על שאלה אחת מהשאלות 15-16 (25 נקודות)

C# תכנות מונחה עצמים

15 אלה (בחירה, 25 נקודות)

כמו פתרון תרגיל 13 בתמי"ע java

16 אלה (בחירה, 25 נקודות)

בהצלחה!