

מדעי המחשב

פרק ראשון

שאלה 1

Java	C#
<pre> //--- האם העפרון הנוכחי ארוך יותר? --- public boolean isLonger (Pencil other) { return this.length > other.length; } //--- האם העפרון המוכחי מחודד? --- public boolean isSharpened() { return this.sharpened; } </pre>	<pre> public bool IsLonger(Pencil other) { return (this.length > other.length); } public bool IsSharpened() { return sharpened == true; } </pre>
נכתב ע"י ראמי ג'בלי	

pencil1	pencil2	pencil1.isLonger(pencil2) && pencil1.isSharpened()	pencil2.isSharpened()	פלט
		F && T → false	false	pencil2 needs to be sharpened

Pencil	Pencil
12 : אורך T מחודד?	13 : אורך F מחודד?

שאלה 2

Java

```
//--- 0 - ל-0 --- בנאי המאחז את הדירוג
public Doctor(String name, String specialization)
{
    this.name = name;
    this.specialization = specialization;
    this.rate = 0;
}

//--- פעולה המעדכנת את הדירוג של כל רופא במערך ---
public static void doctorRating (Doctor[] doctors)
{
    System.out.print("enter rates --> ");
    for (int i = 0 ; i < doctors.length ; i++)
        if (doctors[i] != null)
            doctors[i].setRate(calcRate());
}

//--- פעולה הקולטת את ההערכות שניתנו לרופא ---
//--- ומחזירה את הדירוג שלו ---
public static double calcRate()
{
    int sum = 0, count = 0;
    int rate = input.nextInt();
    while(rate != -1)
    {
        sum += rate;
        count ++;
        rate = input.nextInt();
    }

    if (count == 0)
        return 0;
    return (double) sum/count;
}
```

C#

הפתרון לשאלה זו נכתב ע"י ראמי ג'בלי

```
public Doctor(string name, string specialization)
{
    this.name = name;
    this.specialization = specialization;
    this.rate = 0;
}
```

```
//פעולה מעדכנת במערך את הדירוג לפי הממוצעים
public static void DoctorRating(Doctor[] doctors)
{
    int rate, sum, count;
    double avg;
    for (int i = 0; i < doctors.Length; i++)
    {
        sum = 0; count = 0;
        rate = int.Parse(Console.ReadLine());
        while (rate != -1)
        {
            sum = sum + rate;
            count++;
            rate = int.Parse(Console.ReadLine());
        }
        avg = (double)sum / count;
        doctors[i].SetRate(avg);
    }
}
```

שאלה 3

Java

א.

```
//--- פעולה המקבלת מערך ומקום תקין במערך ---
//--- ומחזירה אמת אם האיבר במקום זה הוא איבר שיוויון ושקר אחרת ---
//--- איבר שיוויון הוא איבר שסכום האיברים משמאלו ומימינו שווים ---
public static boolean isEqualItem (int [] arr, int p)
{
    return sumLeft(arr, p) == sumRight(arr, p);
}

//--- פעולה המקבלת מערך ומקום ומחזירה את סכום ---
//--- האיברים מההתחלה עד אליו (לא כולל) ---
public static int sumLeft(int [] arr, int p)
{
    int sum = 0;
    for (int i = 0 ; i < p ; i++)
        sum += arr[i];
    return sum;
}

//--- פעולה המקבלת מערך ומקום ומחזירה את סכום ---
//--- האיברים מהאיבר שאחריו עד הסוף ---
public static int sumRight(int [] arr, int p)
{
    int sum = 0;
    for (int i = p+1 ; i < arr.length ; i++)
        sum += arr[i];
    return sum;
}
```

ב.

```
//--- פעולה המחזירה אמת אם המערך הוא מערך שיוויוני ושקר אחרת ---
public static boolean isEqualArray (int [] arr)
{
    int i = 0;
    while (i < arr.length)
    {
        if (isEqualItem(arr, i))
            return true;
        i++;
    }
    return false;
}
```

שאלה 3

C#

הפתרון לשאלה זו נכתב ע"י ראמי

```
// פתרון שאלה 3 חלק א
// פעולה מחזירה אמת אם האיבר place זה הוא איבר שוויון אחרת מחזירה הפעולה שקר
public static bool NumEqual(int[] arr, int place)
{
    int sumLeft = 0, sumRight = 0;
    for (int i = 0; i < place; i++)
        sumLeft = sumLeft + arr[i];
    for (int i = place + 1; i < arr.Length; i++)
        sumRight = sumRight + arr[i];
    return sumLeft == sumRight;
}
```

```
// פתרון שאלה 3 חלק ב
// פעולה מחזירה אמת אם המערך הוא שוויוני אחרת מחזיר שקר
public static bool ArrayEqual(int[] arr)
{
    int c = 0;
    for (int i = 0; i < arr.Length; i++)
        if (NumEqual(arr, i))
            c++;
    Console.WriteLine(c);
    return c > 0;
}
```

פרק שני

עלה 4

בפתרון שהצגתי החלפתי בין הרשימה המתקבלת והרשימה המוחזרת.
בחרתי להשאיר את שני הפתרונות, למרות שרק אחד מהם נכון

הפתרון הנכון: הפעולה מקבלת את רשימת המספרים ומחזירה את רשימת הספרות

Java

```
//--- פעולה המקבלת שרשרת של מספרים שלמים ---
//--- ומחזירה שרשרת של ספרות המרכיבות את המספרים ---
//--- בסיום כל מספר תופיע חוליה ובה המספר -9 ---
public static Node<Integer> buildNumbers (Node<Integer> lst)
{
    Node<Integer> nLst = new Node<Integer> (999); // חוליית דמה
    Node<Integer> pos = nLst;

    int num;
    while (lst != null)
    {
        num = lst.getValue();

        //--- הכנסת ספרות המספר לרשימה ---
        //--- בסדר הפוך, כך שספרת האחדות ראשונה ---
        //--- הספרה הכי משמעותית אחרונה ---
        while (num != 0)
        {
            pos.setNext(new Node<Integer>(num%10));
            num = num / 10;
            pos = pos.getNext();
        }
        pos.setNext(new Node<Integer>(-9));
        pos = pos.getNext();
        lst = lst.getNext();
    }
    return nLst.getNext(); // הרשימה ללא חוליית הדמה
}
```

C#

```
//--- פעולה המקבלת שרשרת של מספרים שלמים ---
//--- ומחזירה שרשרת של ספרות המרכיבות את המספרים ---
//--- בסיום כל מספר תופיע חוליה ובה המספר 9 ---
public static Node<int> BuildNumbers(Node<int> lst)
{
    Node<int> nLst = new Node<int>(999); // חוליית דמה
    Node<int> pos = nLst;
    int num;

    while (lst != null)
    {
        num = lst.GetValue();

        //--- הכנסת ספרות המספר לרשימה ---
        //--- בסדר הפוך, כך שספרות האחדות ראשונה ---
        //--- הספרה הכי משמעותית אחרונה ---
        while (num != 0)
        {
            pos.SetNext(new Node<int>(num % 10));
            num = num / 10;
            pos = pos.GetNext();
        }
        pos.SetNext(new Node<int>(-9));
        pos = pos.GetNext();
        lst = lst.GetNext();
    }
    return nLst.GetNext(); // הרשימה ללא חוליית הדמה
}
```

הפתרון הלא נכון - הפתרון ההפוך -

הפעולה מקבלת את רשימת הספרות ומחזירה את רשימת המספרים

Java

```
//--- פעולה המקבלת שרשרת של ספרות ---
//--- ומחזירה שרשרת של מספרים שלמים ---
public static Node<Integer> buildDigit (Node<Integer> lst)
{
    Node<Integer> nLst = new Node<Integer> (-1);
    Node<Integer> pos = nLst;

    int num = 0, p = 0;      // מעריך החזקה
    int digit;
    while (lst != null)
    {
        digit = lst.getValue();
        if (digit != -9)      //--- ספרה חוקית במספר ---
        {
            num += digit * Math.pow(10, p);
            p++;
        }
        else                  //--- נגמרו ספרות המספר הנוכחי ---
        {
            pos.setNext(new Node<Integer>(num));
            pos = pos.getNext();

            //--- אתחול המספר ומעריך החזקה ---
            num = 0;
            p = 0;
        }
        lst = lst.getNext();
    }

    return nLst.getNext();
}
```


C#

הפתרון לשאלה זו נכתב ע"י ראמי ג'בלי

פעולה מחזירה שרשרת ספרות המייצגת את המספרים ברשימה הנתונה

```
public static Node<int> ListDigit(Node<int> list)
{
    Node<int> pos = list;
    Node<int> pos2 = null;
    int x, num = 0;
    int p = 1;
    Node<int> listDigit = null;
    while (pos != null)
    {
        x = pos.GetValue();
        if (x != -9)
        {
            num = num + x * p;
            p = p * 10;
        }
        else
        {
            if (listDigit == null)
            {
                listDigit = new Node<int>(num);
                pos2 = listDigit;
            }
            else
            {
                pos2.SetNext(new Node<int>(num));
                pos2 = pos2.GetNext();
            }
            num = 0; p = 1;
        }
        pos = pos.GetNext();
    }
    return listDigit;
}
```

שאלה 5

Java

```
//--- פעולה המקבלת מחסנית של מספרים שלמים ---
//--- הפעולה תחזיר מחסנית המכילה את כל האיברים שאינם חלק מבלו מבלו ---
//--- "בלוק" יחשב רצף של לפחות 2 מספרים זהים ---
public static Stack<Integer> notInBlock (Stack<Integer> stk)
{
    Stack<Integer> sTemp = new Stack<Integer>();
    Stack<Integer> nStk = new Stack<Integer>();

    int count = -1, num = -1;

    //--- המספר שבראש המחסנית ---
    if (! stk.isEmpty())
    {
        num = stk.top();
        count = 1;
        sTemp.push(stk.pop());
    }

    while (! stk.isEmpty())
    {
        if (stk.top() == num)
            count++;
        else
        {
            //--- המספר שבראש המחסנית אינו חלק מהבלוק ---
            if (count == 1)
                nStk.push(num);
            num = stk.top();
            count = 1;
        }
        sTemp.push(stk.pop());
    }
    //--- טיפול באיבר האחרון ---
    if (count == 1)
        nStk.push(num);

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

    return nStk;
}
```

C#

הפתרון לשאלה זו נכתב ע"י ראמי ג'בלי

פעולה מחזירה מחסנית חדשה שתכיל כל האיברים מהמחסנית המקורית שאינם בבלוק

```
public static Stack<int> NonBlock(Stack<int> s)
```

```
{
    Stack<int> s2 = new Stack<int>();
    if (s.IsEmpty())
        return s2;
    int x = s.Pop();
    int c = 1;
    while (!s.IsEmpty())
    {
        int y = s.Pop();
        if (x == y)
            c++;
        else
        {
            if (c == 1)
                s2.Push(x);
            c = 1;
        }
        x = y;
    }
    if (c == 1)
        s2.Push(x);
    return s2;
}
```

שאלה 6

Java

```
//---          פעולה המקבלת עץ בינארי ומספר שלם ---
//---          ומחזירה כמה איברים בעץ משאירים שארית ---
//---          השווה למספר השלם כשמחלקים אותם ב-3 ---
public static int countRemains(BinNode<Integer> tree, int x)
{
    if (tree == null)
        return 0;
    if (tree.getValue() % 3 == x)
        return 1 + countRemains(tree.getLeft(), x)
            + countRemains(tree.getRight(), x);
    return countRemains(tree.getLeft(), x) +
        countRemains(tree.getRight(), x);
}

//---          פעולה המחזירה אמת אם העץ הוא עץ שאריות שוויוני ושקר אחרת ---
public static boolean isEqualRemains (BinNode<Integer> tree)
{
    int count0 = countRemains(tree, 0);
    int count1 = countRemains(tree, 1);
    int count2 = countRemains(tree, 2);

    return count0 == count1 && count0 == count2;
}
```

C#

הפתרון לשאלה זו נכתב ע"י ראמי ג'בלי

```
// פעולה מחזירה מספר הצמתים בעץ שמתחלקים ב 3 ללא שארית
public static int CountMod3_0(BinNode<int> t)
{
    if (t == null)
        return 0;
    else if (t.GetValue() % 3 == 0)
        return 1 + CountMod3_0(t.GetLeft()) + CountMod3_0(t.GetRight());
    else return CountMod3_0(t.GetLeft()) + CountMod3_0(t.GetRight());
}

// פעולה מחזירה מספר הצמתים בעץ שמתחלקים ב 3 עם שארית עם שארית 1
public static int CountMod3_1(BinNode<int> t)
{
    if (t == null)
        return 0;
    else if (t.GetValue() % 3 == 1)
        return 1 + CountMod3_1(t.GetLeft()) + CountMod3_1(t.GetRight());
    else return CountMod3_1(t.GetLeft()) + CountMod3_1(t.GetRight());
}

// פעולה מחזירה מספר הצמתים בעץ שמתחלקים ב 3 עם שארית עם שארית 2
public static int CountMod3_2(BinNode<int> t)
{
    if (t == null)
        return 0;
    else if (t.GetValue() % 3 == 2)
        return 1 + CountMod3_2(t.GetLeft()) + CountMod3_2(t.GetRight());
    else return CountMod3_2(t.GetLeft()) + CountMod3_2(t.GetRight());
}

// פעולה מחזירה אמת אם העץ הוא עץ שארית שוויוני אחרת מחזירה שקר
public static bool TreeEqual(BinNode<int> t)
{
    return (CountMod3_0(t) == CountMod3_1(t) && CountMod3_1(t) == CountMod3_2(t));
}
```

פרק שלישי

מערכות מחשב ואסמבלי

הפתרון לפרק זה נכתב ע"י: רן וינשטיין

סעיף א

שאלה 7

AL	AH	BX	SI	DI	BX + SI	BX + DI
		100h				
			0			
				7		
4						
	8					
					4	
						8
			1			
				6		
					4	
						8
			2			
				5		
					4	
						8
			3			
				4		
					4	
						8
			4			
				3		
					4	
						8
			5			
				2		
					4	
						8
			6			
				1		

מפת זיכרון - 16% (כל ערך בטבלה - 1%)

כתובת	107h	106h	105h	104h	103h	102h	101h	100h
תוכן	8h	8h	4h	4h	8h	8h	4h	4h

סעיף ב:

```

AND AL,00000001b
CMP AL,1
JE L1
AND AH,00000001b
CMP AH,1
JE L1
MOV DL,1
JMP SOF
L1:
MOV DL,0
SOF:

```

סעיף ג - 15%

aAX 9% וכל אחד מהדגלים 2%

AX	CF	SF	ZF
052Ah	0	0	0

שאלה 8

```

proc POW
    push bp
    mov bp,sp
    mov al,[bp+4]
    mul al
    pop bp
    ret 4
endp POW

mov bx, offset ARR
mov cx,10
AGAIN:
    push bx
    call POW
    cmp [bx],al
    jne SOF
    inc bx
    loop AGAIN
    mov [RANK],10
    jmp exit
SOF:
    mov [RANK],bl
EXIT:

```


חקר ביצועים

הפתרון לפרק זה נכתב ע"י: ???

שאלה 9

שאלה 10

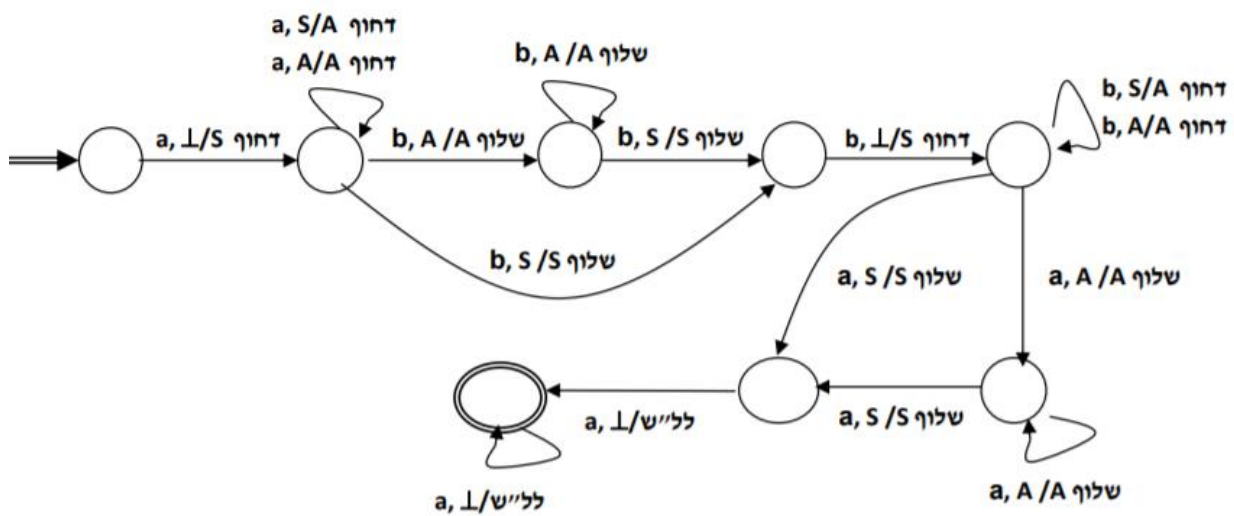
מודלים חישוביים

הפתרון לפרק זה נכתב ע"י: רחל לוזמר

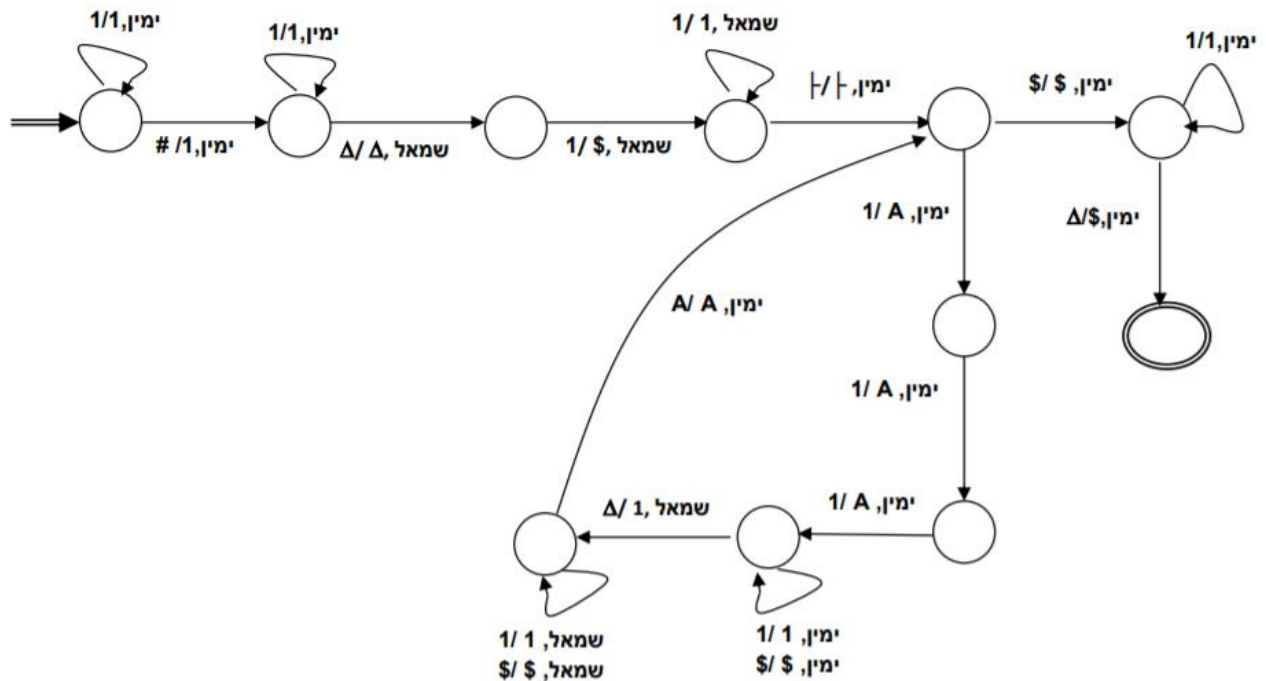
שאלה 11

$$L = \{a^n b^m a^k \mid n > 0, m > n, n + k > m\}$$

$$= \{a^n b^n b^{m-n} a^k \mid n > 0, k > m - n\}$$



שאלה 12

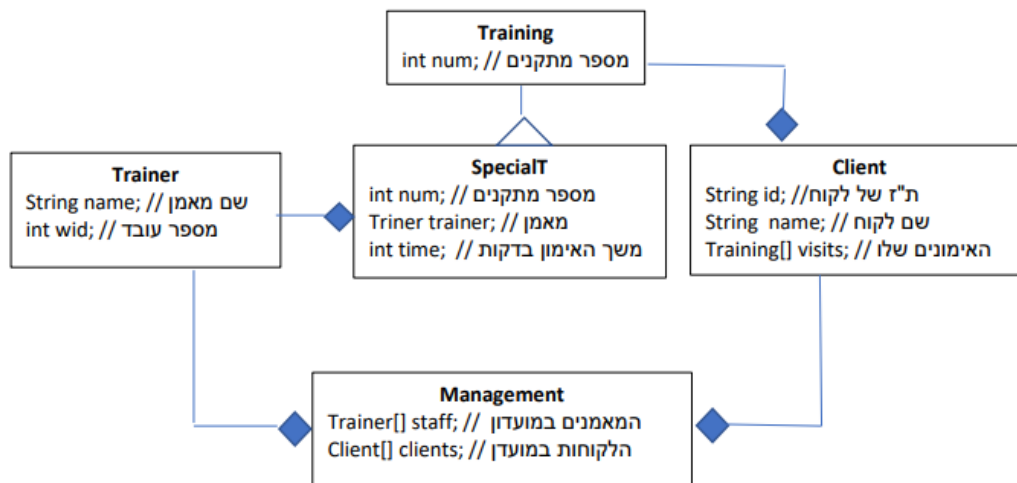


תכנות מונחה עצמים - Java

הפתרון לפרק זה נכתב ע"י: EVI גרינוואלד

שאלה 13

א- תרשים היררכיית המחלקות:



ב- כותרות ותכונות המחלקות

public class Trainer	מחלקת מאמן
private String name; // שם מאמן	התכונות:
private int wid; // מספר עובד	
public class Training	מחלקת אימון
protected int num; // מספר מתקנים שהשתמש	התכונות:
public class SpecialT extends Training	מחלקת אימון מיוחד
private Trainer trainer; // האמן של שלקוח	התכונות:
private int time; // משך האימון בדקות	
public class Client	מחלקת לקוח
private String id; // ת"ז של לקוח	התכונות:
private String name; // שם הלקוח	
private Training[] visits; // אוסף האימונים שלקח	
אפשרות שמקלה על הקוד:	
private int numOfVisits; // מספר האימונים בפועל שלקח הקלוח	
public class Management	מחלקת הנהלה
private Trainer[] staff; // אוסף מאמנים במועדון	התכונות:
private Client[] clients; // אוסף לקוחות במועדון	
אפשרות שמקלה על הקוד:	
private int numOfClients; // מספר הלקוחות הפעילים במועדון	

ג- פעולה במחלקה Client

תנאי קדם:

תנאי בתר: מחזירה את סכום כל הדקות שהלקוח התאמן אימון מיוחד. 0 - אם אין כזה.

```
public int totalTime()
```

```
{
    int totalT = 0;
    for (int t = 0; t < this.visits.length; t++)
    {
        if (this.visits[t] != null && this.visits[t] instanceof SpecialT)
            totalT += ((SpecialT) this.visits[t]).getTime();
    }
    return totalT;
}
```

קוד במקרה שאין תכונה נוספת ששומרת את מספר האימונים בפועל

```
public int totalTime()
```

```
{
    int totalT = 0;
    for (int t = 0; t < this.numOfVisits; t++)
    {
        if (this.visits[t] instanceof SpecialT)
            totalT += ((SpecialT) this.visits[t]).getTime();
    }
    return totalT;
}
```

קוד במקרה שיש תכונה נוספת ששומרת את מספר הלקוחות בפועל.

ד- במחלקה Management :

תנאי קדם:

תנאי בתר: מספר הלקוחות שהתאמנו באימון מיוחד לפחות פעם אחת.

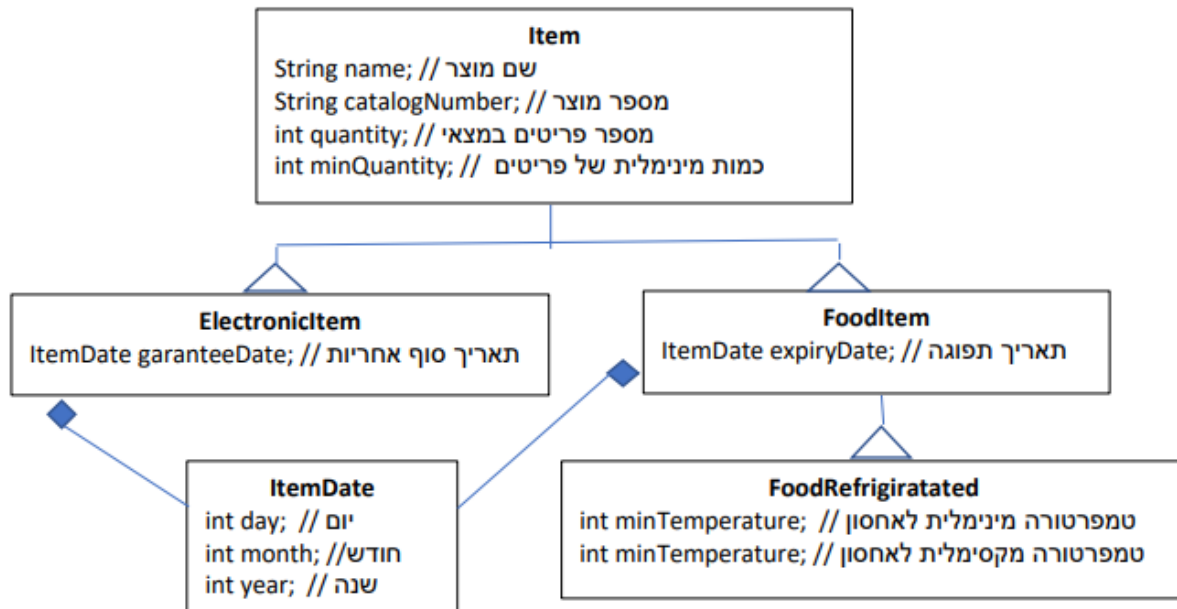
```
public int numClientsSpecialTraining()
```

```
{
    int countSpecailT = 0;
    Client client;
    for (int c = 0; c < this.clients.length; c++)
    {
        if (this.clients[c] != null)
        {
            client = this.clients[c];
            if (client.totalTime() > 0)
                countSpecailT++;
        }
    }
    return countSpecailT;
}
```

הסבר: אם הפעולה של לקוח מחזירה סך דקות באימון מיוחד שגדול מ 0, סימן שהלקוח התאמן אימון מיוחד אחד לפחות ולכן נמנה לקוח זה.

שאלה 14

א- תרשים היררכיית המחלקות:



ב- פעולה המחזירה 'אמת' אם חסרים מוצרים במלאי, 'שקר' -אחרת

```

public boolean isMissing()
{
    return this.quantity < this.minQuantity;
}

```

הפעולה נכתבת במחלקה
. Item

ב- פעולה המחזירה 'אמת' אם חסרים מוצרים במלאי, 'שקר' -אחרת

```

public boolean isMissing()
{
    return this.quantity < this.minQuantity;
}

```

הפעולה נכתבת במחלקה
. Item

ג- מאחר והפעולה עובדת על עצם מטיפוס Item, נוסיף את הפעולה למחלקה זו.
 במחלקת FoodRefrigerated נדרוש את הפעולה ונתאים את החזרת הערך מהפעולה בהתאם
 לכך שהטמפרטורה תהייה בתווך הרצוי.
 כשנפעיל את הפעולה על מערך עצמים מטיפוס Item, בזכות דריסת הפעולה והפולימורפיזם,
 התכנית תדע לזהות עבור כל מוצר האם ניתן לשמור את המוצר בטמפרטורה temp או לא.

במחלקה Item:

תנאי קדם: מקבלת טמפרטורה temp
 תנאי בתר: אמת אם ניתן לשמור את המוצר בטמפרטורה temp, שקר – אחרת

```
public boolean canBeSotored(int temp)
{
    return true;
}
```

אין הגבלה את מידת
 הטמפרטורה לשמירה
 על המוצר

במחלקה FoodRefrigerated

תנאי קדם: מקבלת טמפרטורה temp
 תנאי בתר: אמת אם ניתן לשמור את המוצר בטמפרטורה temp, שקר – אחרת

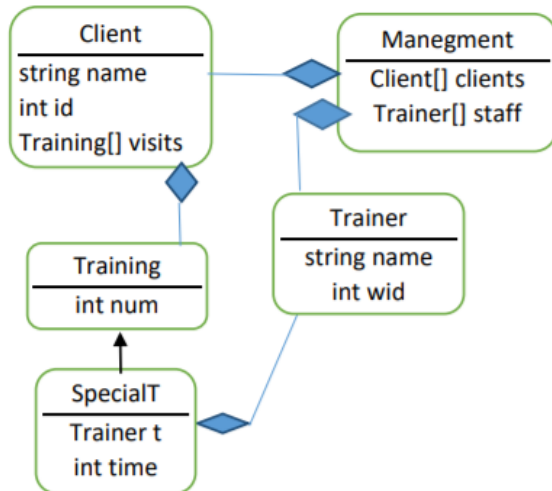
```
public boolean canBeSotored(int temp)
{
    return temp >= this.minTeperature && temp <= this.maxTeperature;
}
```

במקרה זה, ניתן לשמור על המוצר אם הטמפרטורה שלו היא בתווך בין טמפרטורה מינימלית
 למקסימלית, כולל.

תכנות מונחה עצמים - C#
הפתרון לפרק זה נכתב ע"י: דיתה אוהב ציון

שאלה 15

א. תרשים מחלקות



ב. כותרות המחלקות

<pre> public class Manegment { private Trainer[] staff; private Client[] clients; } </pre>	
<pre> public class Trainer { private string name; private int wid; } </pre>	
<pre> public class Client { private string name; private int id; Training[] visits; } </pre>	
<pre> public class Training { int num; } </pre>	
<pre> public class SpecialT:Training { private Trainer train; private int time; } </pre>	מספר המכשירים נמצא באימון. המחלקה תכיל רק את הנתונים שהם הרחבה של האימון.

הערה והסבר: (לא נדרש בבחינה) הוספת אימונים נעשית ע"י פעולות בלקוח – המוסיפה אימון או אימון מיוחד למערך האימונים של הלקוח, כך שמערך visits מכיל את שני סוגי האימונים.

<pre> public void Add(Training t) { int i = 0; while (visits[i] != null) i++; visits[i] = t; } </pre>	<pre> public void Add(SpecialT s) { int i = 0; while (visits[i] != null) i++; visits[i] = s; } </pre>
---	---

ג. פעולה במחלקה Client המחזירה כמה דקות אימון מיוחד עשה (0 אם אין)

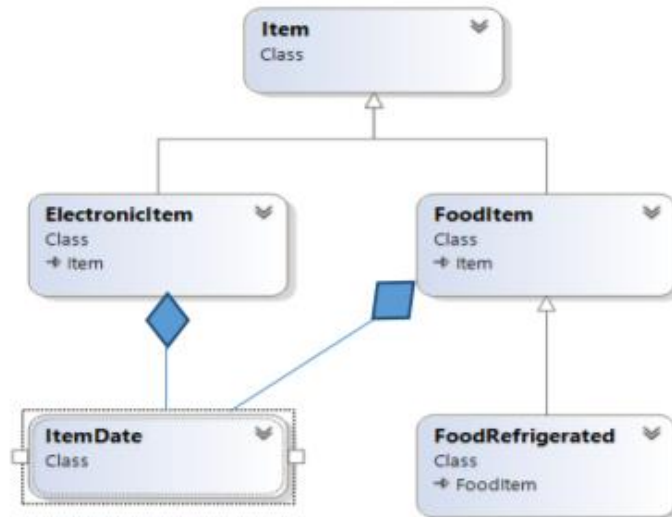
```
public int SpecialTime()
{
    int total = 0;
    foreach(Training t in visits)
    {
        if(t!=null)
        {
            if (t is SpecialT)
                total += ((SpecialT)t).GetTime(); // נדרשת המרה מפורשת כלפי מטה.
        }
    }
    return total;
}
```

ד. פעולה המחלקה Manegment המחזירה כמה לקוחות התאמנו אימון מיוחד.

```
public int CountSpecialClient()
{
    int count = 0;
    foreach(Client c in clients)
    {
        if(c!=null)
        {
            if (c.SpecialTime() > 0)
                count++;
        }
    }
    return count;
}
```

שאלה 16

א. תרשים מחלקות



ב. במחלקה `Item`

```

public bool IsMissing()
{
    return (quantity < minQuantity);
}
    
```

ג.

במחלקה `FoodRefrigerated` תהיה פעולה המחזירה אמת אם הטמפרטורה `temp` בטווח המתאים

```

public bool SuitableTemperature(int temp)
{
    return (temp <= maxTemperature && temp >= minTemperature);
}
    
```

במחלקה `Item` נכתוב את הפעולה המבוקשת

```

public bool CanBeStored( int temp)
{
    if (this is FoodRefrigerated)
        return ((FoodRefrigerated)this).SuitableTemperature(temp);
    else
        return true;
}
    
```

בהצלחה !