

מדעי המחשב

פרק ראשון

שאלה 1

Java

```
//--- פעולה המקבלת מערך של מספרים שלמים חיוביים ושלייליים ---
//--- ומחזירה אמת אם כל המספרים החיוביים במערך ממוינים בסדר עולה ושקר אחרת ---
public static boolean posOrder (int[]arr)
{
    int x = 0;
    //--- טיפול בשאר האיברים ---
    for (int i = 0 ; i < arr.length ; i++)
    {
        if (arr[i] > 0)
        {
            if (arr[i] < x)
                return false;
            x = arr[i];
        }
    }
    return true;
}
```

C#

פתרון C# לשאלה זו נכתב ע"י: אחמד אגבאריה

```
public static bool PosOrder(int[] arr)
{
    int x = 0;
    for (int i = 0; i < arr.Length; i++)
    {
        //check only positive numbers
        if (arr[i] > 0)
        {
            //if new one >= the prev one (x)
            if (arr[i] >= x) { x = arr[i]; }
            //else - false
            else return false;
        }
    }
    return true;
}
```

Java

```
//---          2 שאלה          ---
//--- פעולה המקבלת מערך הפרשים שחסר בו איבר אחד ---
//--- ומחזירה את האיבר החסר. ---
//--- מערך הפרשים הוא מערך שהפרש בין ---
//--- כל שני איברים סמוכים הוא קבוע ---
//--- הנחות: האיבר החסר אינו הראשון או האחרון ---
//--- במערך יש לפחות 4 איברים ---
public static int missingNumber (int[]arr)
{
    int diff;
    int d1 = arr[1] - arr[0];           // int len = arr.length;
    int d2 = arr[2] - arr[1];           // int diff = (arr[len-1] - arr[0]) / len;
    if (d1 > 0) // סדרה חשבונית עולה
        diff = Math.min(d1, d2);
    else // סדרה חשבונית יורדת
        diff = Math.max(d1, d2);

    for (int i = 1 ; i < arr.length ; i++)
    {
        if (arr[i] - arr[i-1] != diff)
            return arr[i-1] + diff;
    }
    return -1;
}

//--- המערך יוצר סדרה חשבונית שחסר בה איבר אחד ---
//---  $a(n) = a(1) + d * (n-1)$  בסדרה ---
//---  $arr[n] = arr[0] + d * n$  במערך ---
//--- נחפש את האיבר בשיטה של חיפוש בינארי ---
//--- הנחות: האיבר החסר אינו הראשון או האחרון ---
//--- במערך יש לפחות 4 איברים ---
public static int missingNum (int[]arr)
{
    int len = arr.length;
    int d = (arr[len-1] - arr[0]) / len;

    int left = 0, right = arr.length-1, n;
    int num;
    while (right - left > 1)
    {
        n = (left + right) / 2;
        num = arr[0] + d * n;
        if (arr[n] > num)
            right = n; // האיבר החסר נמצא בצד שמאל
        else
            left = n; // האיבר החסר נמצא בצד ימין
    }
    return arr[left] + d;
}

missingNum [6,8,12,14,16,18]: 10
missingNum [2,6,8,10,12]: 4
```

שאלה 2

C#

פתרון C# לשאלה זו נכתב ע"י: אחמד אגבאריה

```
public static int MissingNum(int[] arr)
{
    //assume arr.Length>=4
    if (arr.Length < 4) return -1;

    int diff1 = arr[1] - arr[0];
    int diff2 = arr[2] - arr[1];
    int diff3 = arr[3] - arr[2];
    //check the real one
    int diff;
    if(diff1==diff2)    diff = diff1;
    else if(diff2==diff3)    diff = diff2;
    else                diff = diff3;

    for (int i = 0; i < arr.Length - 1; i++)
    {
        int d = arr[i + 1] - arr[i];
        if(diff != d)
        {
            return arr[i] + d / 2;
        }
    }
    //We shouldn't arrive here
    return -1;
}
```

שאלה 3

Java

```
//---          שאלה 3 - סעיף א          ---
//---          פעולה המחזירה אמת אם הרכב ביצע עבירת תנועה          ---
//---          עבירת תנועה היא רכב פרטי בנת"צ או מעל המבירות          ---
public boolean illegal (int maxSpeed)
{
    return this.privateCar || this.speed > maxSpeed;
}

//---          שאלה 3 - סעיף ב          ---
//---          פעולה המחזירה אמת אם במצלמה הנוכחית לא בוצעו עבירות תנועה ושקר אחרת          ---
public boolean allGood()
{
    for (int i = 0 ; i < this.cars.length ; i++)
    {
        if (this.cars[i] != null && this.cars[i].illegal(maxSpeed))
            return false;
    }
    return true;
}
```

```

//---          שאלה 3 סעיף ב(2)          ---
//---          פעולה המחזירה את מספר הערים שלא בוצעו בהן עבירות ---
//---          בכל עיר יכולה להיות יותר ממצלמה אחת. ---
public static int legalCity (CameraInfo[]cameras)
{
    //---          יצירת מערך בוליאני בהתאם לקוד העיר. ---
    //---          אם נבדקה מצלמה בעיר זו, יסומן התא המתאים ב-true ---
    boolean[] check = getCheckedArr();

    int count = 0;
    for (int i = 0 ; i < cameras.length ; i++) // עבור כל מצלמה
    {
        int cityCode = cameras[i].getCity();
        if (!check[cityCode]) // אם טרם נבדקו מצלמות העיר
        {
            check[cityCode] = true; // סמן את העיר כנבדקה
            if (legalCity(cameras, cityCode)) // בדוק מצב העיר
                count ++;
        }
    }
    return count;
}

//---          פעולה המקבלת קוד עיר ואת מערך המצלמות ---
//---          ומחזירה אמת אם בכל המצלמות בעיר זו ---
//---          לא בוצעו עבירות מהירות ושקר אחרת ---
public static boolean legalCity (CameraInfo[]cameras, int cityCode)
{
    for (int i = 0 ; i < cameras.length ; i++)
        if (cameras[i].getCity() == cityCode && !cameras[i].allGood())
            return false;
    return true;
}

//---          פעולה המחזירה מערך בוליאני לפי קוד הערים ---
//---          המערך מאותחל לשקר בכל תאיו. ---
//---          בהמשך כל עיר עיר שתיבדק תקבל ערך אמת ---
public static boolean[] getCheckedArr ()
{
    boolean check [] = new boolean [100];
    for (int i = 0 ; i < check.length ; i++)
        check[i] = false;
    return check;
}

```

שאלה 3

C#

פתרון C# לשאלה זו נכתב ע"י: אחמד אגבאריה

```
//Q3 - A
1 reference
public bool Illegal(int maxSpeed)
{
    if (privateCar) return true;
    return speed > maxSpeed;
}

//Q3 - B - 1
1 reference
public bool AllGood()
{
    //assume cars[i] != null for all
    for (int i = 0; i < cars.Length; i++)
    {
        //if one car is illegal - we not all-good
        if (cars[i].Illegal(maxSpeed)) return false;
    }
    return true;
}

//Q3 - B - 2
0 references
public static int LegalCities(CameraInfo[] cameras)
{
    //array for cities ids - 0 means not good, 1 means goo
    int[] cities = new int[100];
    for (int i = 0; i < cities.Length; i++)
    {
        cities[i] = 0;
    }
    for (int i = 0; i < cameras.Length; i++)
    {
        if (cameras[i].AllGood())
        {
            //Add city only once to the queue
            int city = cameras[i].GetCity();
            cities[city] = 1;
        }
    }
    //sum all good ones
    int c = 0;
    for (int i = 0; i < cities.Length; i++)
    {
        //cities[i] either 0 or 1
        c += cities[i];
    }
    return c;
}
```

פרק שני

שאלה 4

Java

```
//--- פעולה המקבלת תור ומספר שלם ומחזירה אמת ---
//--- אם קיימים בתור שי מספרים שסכומם שווה למספר זה ושקר אחרת ---
//--- המספרים לא חייבים להיות סמוכים בתור ---
public static boolean twoSum (Queue<Integer> q, int x)
{
    while (!q.isEmpty())
    {
        int y = q.remove();
        if (exist(q, x-y))
            return true;
    }
    return false;
}

//--- פעולה המקבלת תור של מספרים שלמים ומספר שלם ---
//--- ומחזירה אמת אם קיים בתור מספר שזה ערכו ושקר אחרת ---
public static boolean exist (Queue<Integer>que, int x)
{
    Queue<Integer>qTemp = new Queue<Integer>();
    boolean found = false;
    while (!que.isEmpty())
    {
        if (que.head() == x)
            found = true;
        qTemp.insert(que.remove());
    }
    while (!qTemp.isEmpty())
        que.insert(qTemp.remove());
    return found;
}
```

שאלה 4

פתרון C# לשאלה זו נכתב ע"י: אשרת פז ו- רונית גל-אור מרציאנו

Java

```
public static boolean twoSum(Queue<Integer> q , int x)
{
    //input : queue with integers and num
    //output: true if found in the queue sum of two numbers = x,
    //        false otherwise

    boolean found = false;
    int n1,n2;

    for(int i=0; i< q.size() && !found; i++)
    {
        n1 = q.remove();
        for(int j=0; j< q.size() && !found; j++)
        {
            n2 = q.remove();
            if(n1+ n2== x)
            {
                found = true;
                q.insert(n2);
            }
            q.insert(n1);
        }
    }
    return found;
}
```


שאלה 4

פתרון C# לשאלה זו נכתב ע"י: אחמד אגבאריה

C#

```

public static bool TwoSum(Queue<int> q, int x)
{
    int len = Size(q);
    for (int i = 0; i < len; i++)
    {
        int x1 = q.Remove();
        Queue<int> temp = new Queue<int>();
        while(!q.IsEmpty())
        {
            int x2 = q.Remove();
            if (x1 + x2 == x) return true;

            temp.Insert(x2);
        }
        q = temp;
    }
    return false;
}

-
//Return clone of the given queue w/o changing the Queue 'q'
1 reference
public static Queue<int> Clone(Queue<int> q)
{
    Queue<int> temp = new Queue<int>();
    Queue<int> copy = new Queue<int>();
    while(!q.IsEmpty())
    {
        temp.Insert(q.Head());
        copy.Insert(q.Remove());
    }
    while(!temp.IsEmpty())
        q.Insert(temp.Remove());
    return copy;
}

```

Java

```
//--- פעולה המוסיפה את המספר לרשימה באופן הבא ---
//--- אם האיבר קיים רשימה, תעדכן את מונה מס' האיברים ---
//--- ואם לא - תוסיף חוליה חדשה כך שמיון הרשימה יישמר ---
public void insertNum(int num)
{
    Node<NumCount> pos = this.lst;
    Node<NumCount> prev = null;
    NumCount nc;

    //--- הוספת האיבר בתחילת הרשימה ---
    if (pos == null || pos.getValue().getNum() > num) // הוספה לרשימה ריקה
    {
        nc = new NumCount(num, 1);
        lst = new Node<NumCount>(nc, lst);
    }
    else if (pos.getValue().getNum() == num) // הוספה לרשימה לא ריקה
    {
        nc = pos.getValue();
        nc.setCount(nc.getCount() + 1);
    }
    else
    {
        //--- הוספת האיבר באמצע הרשימה ---
        while (pos != null && pos.getValue().getNum() < num)
        {
            prev = pos;
            pos = pos.getNext();
        }

        if (pos != null && pos.getValue().getNum() == num) // pos במיקום
        {
            nc = pos.getValue();
            nc.setCount(nc.getCount() + 1);
        }
        else if (prev.getValue().getNum() == num) // prev במיקום
        {
            nc = prev.getValue();
            nc.setCount(nc.getCount() + 1);
        }
        else
        {
            // prev לא קיים. יש להוסיפו בין pos ו- prev
            nc = new NumCount(num, 1);
            prev.setNext(new Node<NumCount>(nc, pos));
        }
    }
}
}
```

א. (2)

סיבוכיות הפעולה $O(n)$ כאשר n מייצג את מספר האיברים ברשימה.
הפעולה עוברת לכל היותר פעם אחת על איברי הרשימה כדי למצוא את מקום ההוספה.
כל צעדי ההתקדמות ובדיקת הערכים היא בסיבוכיות קבועה $O(1)$

ב.

```
//--- פעולה המחזירה את ערך המופע N ---
//--- ית ברשימה n-ערך החוליה ה ---
public int valueN(int n)
{
    Node<NumCount> pos = this.lst;
    NumCount nc;

    while (pos != null)
    {
        nc = pos.getValue();
        n -= nc.getCount();
        if (n <= 0)
            return nc.getNum();
        pos = pos.getNext();
    }
    return -1; // אם לא קיים ברשימה
}
```

שאלה 5

פתרון C# לשאלה זו נכתב ע"י: אשרת פז ורונית גל-אור מרציאנו

java

```
public void insertNum(int x)
{
    // input: integer number
    // output: put new Node in the parm value
    Node<NumCount> pos = this.lst;
    Node<NumCount> posOfNum = null;
    Node<NumCount> newNode;

    NumCount newNum;
    if (pos == null)
    {
        //empty list
        newNum = new NumCount(x, 1);
        newNode = new Node<NumCount>(newNum);
        this.lst = newNode;
    }
    else
    {
        // if exist in list
        posOfNum = existInList(x); // O(n)
        if (posOfNum != null)
        {
            // if num is in list
            NumCount nc = posOfNum.getValue();
            nc.setCount(nc.getCount() + 1);
        }
        else
        {
            if (pos.getValue().getNum() > x)
            {
                // if first in list
                newNum = new NumCount(x, 1);
                newNode = new Node<NumCount>(newNum);
                newNode.setNext(this.lst);
                this.lst = newNode;
            }
        }
    }
}
```

```

else
{
    // insert in middle or last in list
    System.out.println(pos.getValue().getNum());
    // O(n)
    while (pos.getNext() != null &&
           pos.getNext().getValue().getNum() < x)
    {
        //find place in list
        pos = pos.getNext();
    }
    newNum = new NumCount(x, 1);
    newNode = new Node<NumCount>(newNum);
    newNode.setNext(pos.getNext());
    pos.setNext(newNode);
}
}
}

private Node<NumCount> existInList(int x)
{
    // input: int x
    // output: return true or false if x is in list
    Node<NumCount> pos = this.lst;
    while (pos != null)
    {
        if (pos.getValue().getNum() == x)
            return pos;
        pos = pos.getNext();
    }
    return null;
}

```

2. n מספר האיברים ברשימה

הפעולה existInList היא ביעילות $O(n)$ חיפוש איבר ברשימה (n מספר האיברים ברשימה).

בפעולה insertNum עוברים על כל איבר ברשימה פעם אחת. השימוש ב existInList הוא לפני המעבר על הרשימה ולכן היעילות היא:

$$O(n) + O(n) = O(2n) = O(n)$$

שאלה 5

פתרון C# לשאלה זו נכתב ע"י: אחמד אגבאריה

C#

```
//Q - A - 1
//method is O(n)
5 references
public void InsertNum(int x)
{
    //if lst is empty
    if (lst == null) lst = new Node<NumCount>(new NumCount(x, 1));
    //check if to change first node
    else if (lst.GetValue().GetNum() == x)
        lst.GetValue().SetCount(lst.GetValue().GetCount() + 1);
    //if to add as first
    else if (lst.GetValue().GetNum() > x)
        lst = new Node<NumCount>(new NumCount(x, 1), lst);
    else
    {
        Node<NumCount> prev = lst, node = lst.GetNext();
        while (node != null)
        {
            if (node.GetValue().GetNum() == x)
            {
                node.GetValue().SetCount(node.GetValue().GetCount() + 1);
                return;
            }
            if (node.GetValue().GetNum() > x)
            {
                prev.SetNext(new Node<NumCount>(new NumCount(x, 1), node));
                return;
            }
            prev = node;
            node = node.GetNext();
        }
        //add at the end
        prev.SetNext(new Node<NumCount>(new NumCount(x, 1)));
    }
}
2 references
public int ValueN(int n)
{
    //shouldn't arrive here
    if (lst == null) return -1;

    int c = 0;
    Node<NumCount> node = lst;
    while (node != null)
    {
        c += node.GetValue().GetCount();
        if (c >= n) return node.GetValue().GetNum();
        node = node.GetNext();
    }
    //shouldnt arrive here
    return -1;
}
```

שאלה 5

פתרון C# לשאלה זו נכתב ע"י: דיתה אוהב ציון ו- טובי סטפ

C#

```
public void InsertNum(int x)
{
    if (lst == null) // השרשרת ריקה
        lst = new Node<NumCount>(new NumCount(x,1));
    else
        if (x < lst.GetValue().GetNum()) // אם קטן מזוליה ראשונה
            lst = new Node<NumCount>(new NumCount(x,1), lst); // הוספה כזוליה ראשונה
        else
            { // סריקת השרשרת
                Node<NumCount> n = lst;
                bool found = false;
                while (n.GetNext() != null && !found)
                {
                    if (x == n.GetValue().GetNum()) // עידכון המונה
                    {
                        n.GetValue().SetCount(n.GetValue().GetCount() + 1);
                        found = true;
                    }
                    else
                        if (x < n.GetNext().GetValue().GetNum()) // אם קטן מהבא אחריו
                        {
                            n.SetNext(new Node<NumCount>(new NumCount(x,1), n.GetNext())); // הוספת זוליה
                            found = true;
                        }
                    n = n.GetNext();
                }
                // יש לבדוק זוליה אחרונה לעידכון או להוסיף אחריה
                if (x == n.GetValue().GetNum()) // עידכון המונה
                    n.GetValue().SetCount(n.GetValue().GetCount() + 1);
                else
                    n.SetNext(new Node<NumCount>(new NumCount(x,1))); // הוספה כזוליה אחרונה
            }
}
```

א/2 סיבוכיות זמן הריצה היא $O(n)$ כאשר n הוא מספר החוליות בשרשרת. עוברים על השרשרת פעם אחת.

```
public int ValueN(int n)
{
    Node<NumCount> pos = lst;
    int countPlace = 0;
    bool found = false;
    while (pos != null && !found)
    {
        if (countPlace + pos.GetValue().GetCount() < n)
        {
            countPlace += pos.GetValue().GetCount();
            pos = pos.GetNext();
        }
        else
            found = true;
    }
    return pos.GetValue().GetNum();
}
```

שאלה 6

Java

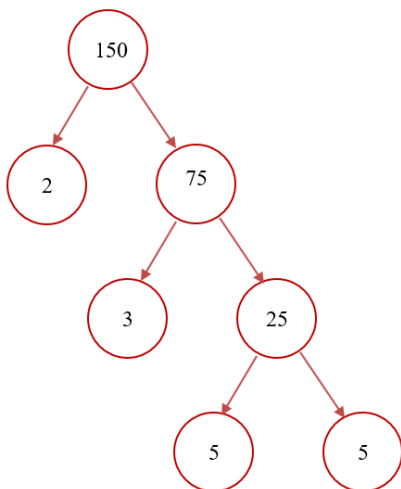
```

//--- פעולה המקבלת עלה בעץ בינארי שערכו לא שלילי ---
//--- אם ערך הצומת הוא ראשוני, יוחזר שקר ---
//--- אחרת, יתווספו לצומת שני בנים שמכפלתם שווה לערך הצומת ---
public static boolean addNodes (BinNode<Integer> tr)
{
    if (tr == null)
        return false;
    int num = tr.getValue();
    if (isPrime(num))
        return false;

    //--- חיפוש מחלקים לערך הצומת ---
    int k = 2;
    while (num % k != 0)
        k ++;
    //--- הוספת בנים ---
    tr.setLeft(new BinNode<Integer>(k));
    tr.setRight(new BinNode<Integer>(num/k));
    return true;
}

```

ב הפעולה what יוצרת עץ בינארי שכל העלים בו מכילים מספרים ראשוניים וכל הצמתים שאינם עלים מכילים מספרים פריקים בכל צומת (שאינו עלה) מכפלת ערך בנו שווה לערך האב



שאלה 6

פתרון C# לשאלה זו נכתב ע"י: אחמד אגבאריה

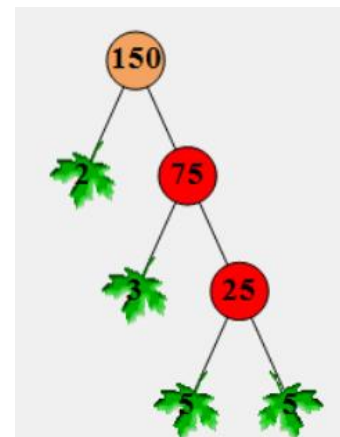
C#

```
public static bool AddNodes(BinNode<int> tr)
{
    if (tr == null || tr.GetValue() < 1) return false;
    if (tr.HasLeft() || tr.HasRight()) return false;

    if (IsPrime(tr.GetValue())) return false;

    int v = tr.GetValue();
    for (int i = 2; i < v / 2; i++)
    {
        if (v % i == 0)
        {
            tr.SetLeft(new BinNode<int>(i));
            tr.SetRight(new BinNode<int>(v / i));
            return true;
        }
    }
    //shouldnt arrive here
    return false;
}
```

```
public static void What(BinNode<int> tr)
{
    if(AddNodes(tr))
    {
        What(tr.GetLeft());
        What(tr.GetRight());
    }
}
```



שאלה 6

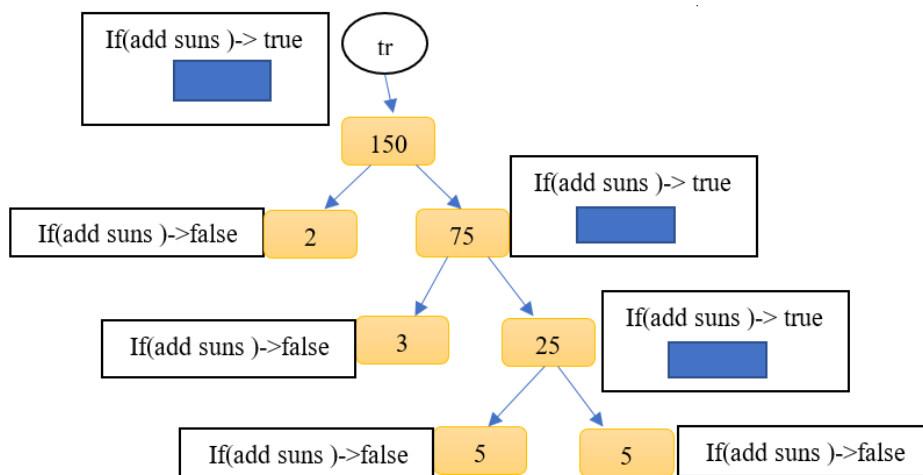
פתרון C# לשאלה זו נכתב ע"י: דיתה אוהב ציון ו- טובי סטפ

C#

```
public static bool AddNodes(BinNode<int> tr)
{
    if (IsPrime(tr.GetValue()))
        return false;
    int x = GetDivisor(tr.GetValue());
    tr.SetLeft(new BinNode<int>(x));
    tr.SetRight(new BinNode<int>(tr.GetValue() / x));
    return true;
}
```

```
public static int GetDivisor(int num)
{
    int x = (int)Math.Sqrt(num);
    for (int i = 2; i <= x; i++)
        if (num % i == 0)
            return i;
    return 1;
}
```

פעולת עזר: מקבלת מספר ומחזירה מחלק אחד שלו



נוצר עץ שעליו הם מספרים ראשוניים.
שאר הצמתים אינם מספרים ראשוניים.

פרק פיסי

אלגוריתמים

הפתרון לפרק זה נכתב ע"י: _____

אזה 7

אזה 8

מודלים חישוביים

הפתרון לפרק זה נכתב ע"י: חן קופלמן

שאלה 9

$$\Sigma = \{a, b\}$$

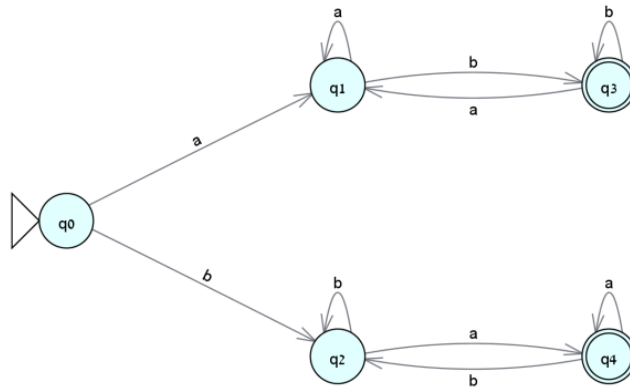
$$L_1 = \{w | w \in \Sigma^*, \#_a(w) = \#_b(w)\}$$

$$L_2 = \{w | w \in \Sigma^*, \#_a(w) > \#_b(w)\}$$

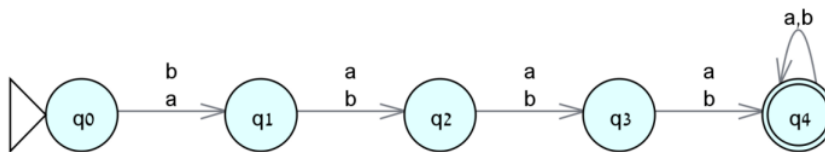
$$L_3 = \{w | w \in \Sigma^*, |w| > 3\}$$

$$L_4 = \{w | w, x \in \Sigma^*, (w = axb \vee w = bxa)\} = \{a\} \cdot \Sigma^* \cdot \{b\} \cup \{b\} \cdot \Sigma^* \cdot \{a\}$$

א. נבנה אוטומט סופי המקבל את השפה L_4 :



ב. נבנה אוטומט סופי המקבל את השפה L_3 :



היות וקיימים אוטומטים סופיים המקבלים את השפות L_3 ו- L_4 , מכאן שהשפות L_3 ו- L_4 רגולריות. עפ"י חוקי סגירות, משפחת השפות הרגולריות סגורה תחת חיתוך, מכאן שהשפה $L_3 \cap L_4$ רגולרית גם כן.

ג. נביע את השפה $L_1 \cup L_2$:

$$L_1 \cup L_2 = \{w | w \in \Sigma^*, \#_a(w) \geq \#_b(w)\}$$

השפה $L_1 \cup L_2$ אינה רגולרית, היות וקיימת תלות אינסופית בין כמות ה- a במילה לכמות ה- b במילה.

ד. עפ"י הגדרת משלים ואיחוד, איחוד של שפה עם השפה המשלימה לה הוא השפה האוניברסאלית.

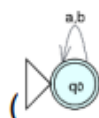
$$L_2 \cup \overline{L_2} = \Sigma^*$$

בנוסף, איחוד כל שפה עם השפה האוניברסאלית הוא השפה האוניברסאלית.

$$L_1 \cup \Sigma^* = \Sigma^*$$

$$\frac{L_1 \cup L_2 \cup \overline{L_2}}{L_1 \cup \Sigma^*} \\ \Sigma^*$$

$$L_1 \cup L_2 \cup \overline{L_2} = \Sigma^*$$



(אם היינו צריכים להוכיח רגולריות, היינו בונים אסד"מ המקבל את Σ^*):

ה. בכל המילים בשפה L_1 , כמות ה- a במילה שווה לכמות ה- b במילה, אך בכל המילים בשפה L_2 , כמות ה- a במילה גדולה מכמות ה- b במילה. מכאן, עפ"י הגדרת החיתוך, בכל המילים בשפת החיתוך $L_1 \cap L_2$, כמות ה- a במילה צריכה להיות גם שווה וגם גדולה מכמות ה- b במילה. דבר זה אינו אפשרי, כלומר לא קיימות מילים המקיימות את דרישות שתי השפות (L_1 ו- L_2) בו זמנית.

$$L_1 \cap L_2 = \emptyset$$

מכאן שהשפות זרות, כלומר $L_1 \cap L_2 = \emptyset$.
השפה הריקה היא שפה סופית (בעלת מספר סופי של מילים, במקרה של השפה הריקה – אפס מילים). כל שפה סופית היא גם שפה רגולרית, מכאן שהשפה הריקה (ומכאן שגם $L_1 \cap L_2$) רגולרית.



(אם היינו רוצים להוכיח רגולריות באמצעות בניית אסד"מ, היינו בונים אסד"מ המקבל את \emptyset):

א. תחילה, נביע את השפה $\overline{L_3}$:

$$\overline{L_3} = \{w \mid w \in \Sigma^*, |w| > 3\}$$

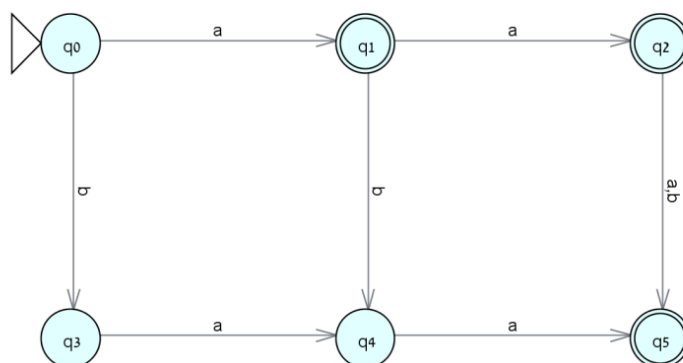
משלים ל- $>$ הוא \leq , נקבל:

$$\overline{L_3} = \{w \mid w \in \Sigma^*, |w| \leq 3\}$$

נעת נביע את השפה $L_2 \cap \overline{L_3}$:

$$L_2 \cap \overline{L_3} = \{a, a^2, a^3, a^2b, aba, ba^2\}$$

נבנה אוטומט סופי המקבל את השפה $L_2 \cap \overline{L_3}$:

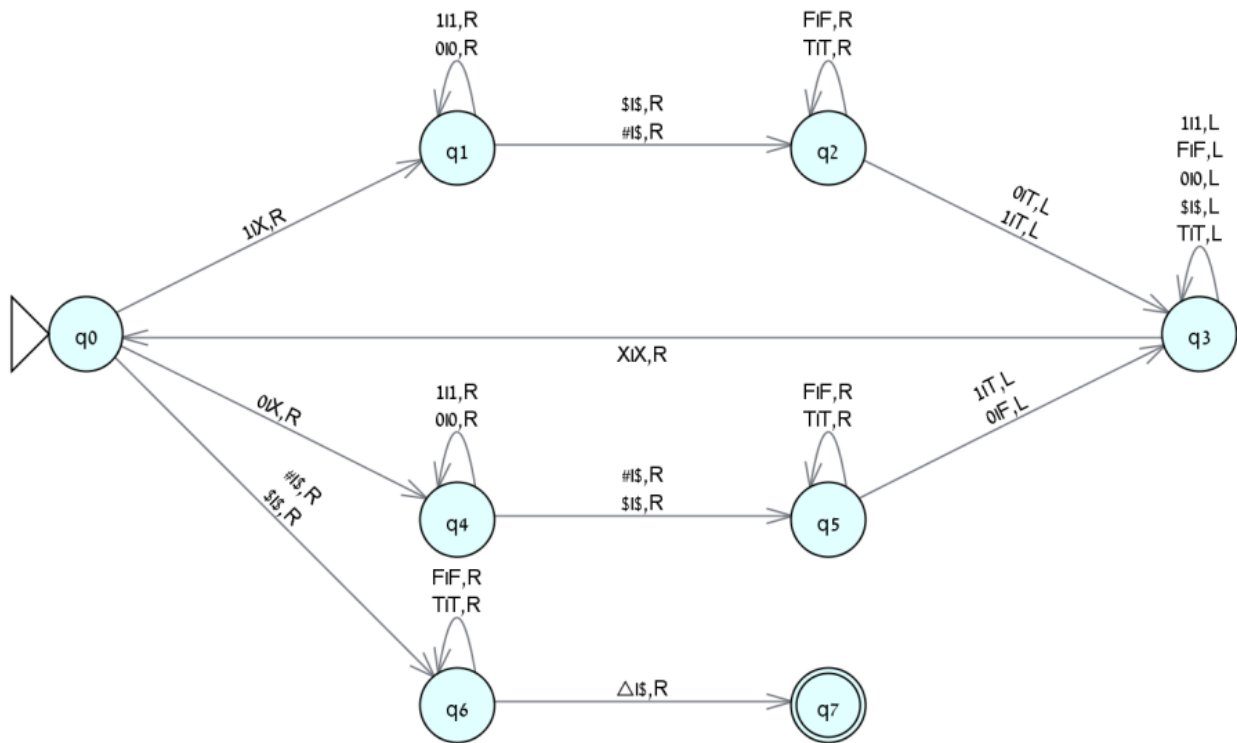


ז. תחילה נביע את השפה $R(L_4)$:

$$R(L_4) = \{w \mid w, x \in \Sigma^*, (w = bxa \vee w = axb)\}$$

קל לראות כי $R(L_4) = L_4$, מכאן ש- $L_4 \cap R(L_4) = L_4$.

שאלה 10

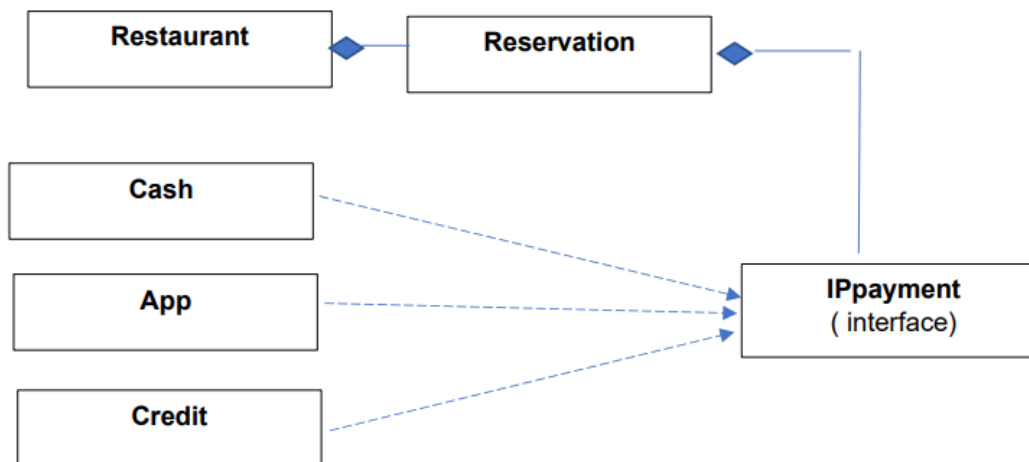


תכנות מונחה עצמים java
פתרון לפרק זה נכתב ע"י: אביטל Evi גרינוואלד

אלף 11

השאלה עוסקת בזיהוי קשרים בין מחלקות וממשקים השלמת פעולות במחלקות ובממשק

סעיף א-1 היררכית מחלקות וממשקים



הסברים:

Resaurant מורכב / מכיל מערך של Reservation. יחס הכלה
Reservation מכיל מערך מטיפוס הממשק IPayment. יחס הכלה בין מחלקה לטיפוס ממשק המשותף בין המחלקות Cash, App, Credit זה התיפקוד.

סעיף א-2

מימוש הפעולה getPrice במחלקות השונות.

<p>הפעולה מחזירה את המחיר לתשלום במזומן. אם המחיר מעל 200 יש הנחה של 10%</p> <pre> public double getPrice() { double price = this.sumCash; if (price > 200) price = price * 0.9; return price; } </pre>	במחלקה Cash
<p>הפעולה מחזירה את המחיר לתשלום באפליקציה לאחר הנחה של 5%</p> <pre> public double getPrice() { return this.sumApp * 0.95; } </pre>	במחלקה App
<p>הפעולה מחזירה את התשלום שמשלם בכרטיס אשראי</p> <pre> public double getPrice() { return this.num * this.part; } </pre>	במחלקה Credit

סעיף ב

Reservation פעולה במחלקה

הפעולה מחזירה את סכום הכסף במזומן שהתקבל בהזמנה כולל שקלול הנחה.

```
public double cashTotal()
{
    double totalCash = 0;
    for (int p = 0; p < this.payments.length; p++)
    {
        if (this.payments[p] instanceof Cash)
            totalCash += this.payments[p].getPrice();
    }
    return totalCash;
}
```

סעיף ג

במחלקה Reservation קיימת הפעולה, אשר מדפיסה עבור סועד שמשלם במזומן את השם שלו, עבור סועד שמשלם באפליקציה את מספר הטלפון שלו ועבור סועד שמשלם בכרטיס אשראי את מספר כרטיס האשראי.

```
public void printDetails()
{
    for (int p = 0; p < this.payments.length; p++)
    {
        System.out.println(this.payments[p].getDetails());
    }
}
```

כדי שהפעולה תתבצע כנדרש נוסיף את הפעולות הבאות לממשק ולמחלקות

public String getDetails();	בממשק Payment
public String getDetails() { return "name is "+this.name; }	במחלקה Cash
public String getDetails() { return "phone is " + this.phoneNumber; }	במחלקה App
public String getDetails() { return "credit card is " + this.creditNumber; }	במחלקה Credit

שאלה 12

השאלה עוסקת במעקב כולל תרשים עצמים וזיהוי שגיאות

א- מעקב בעזרת תרשים עצמים עבור השורות הבאות

1. First f1 = new First(40);
2. First f2 = new First(40,50);
3. First f3 = new Second(100);
4. Second s1 = new Second(100);
5. Second s2 = new Second(100);

1) <First> f1 →

First

x = 40

y = 40

Output: First 1

2) <First> f2 →

First

x = 40

y = 50

Output: First 2

3) <First> f3 →

Second

x = 100

x = 100

z = 100

Output:

First 1

Second

4) <Second> s1 →

Second

x = 100

x = 100

z = 100

Output:

First 1

Second

4) <Second> s2 →

Second

x = 100

x = 100

z = 100

Output:

First 1

Second

Line#	First.count
1	0 1
2	1 2
3	2 3
4	3 4
5	4 5

The output is:

First 1

First 2

First 1

Second

First 1

Second

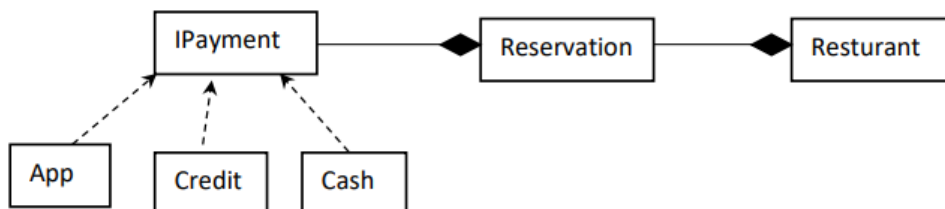
First 1

Second

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

שאלה 13

א. 1. תרשים מחלקות



א. 2. מימוש פעולת המימשק

class Cash	public double GetPrice() { return sumCash > 200? sumCash * 0.9:sumCash; }
class Credit	public double GetPrice() { return part * num; }
class App	public double GetPrice() { return sumApp * 0.95; }

ב. פעולה המחזירה כמה תשלום במזומן

```

public double CashTotal()
{
    double total = 0;
    foreach (IPayment p in payments)
        if (p is Cash)
            total += p.GetPrice();
    return total;
}

```

ג. כדי להפעיל את הפעולה PrintDetails() לכל עצם במערך התשלומים מטיפוס IPayment, הטיפוס IPayment חייב להכיר את הפעולה GetDetails(). לכן צריך להוסיף את הפעולה GetDetails() למימשק, כל שכל אחת ממחלקות התשלום תחזיר את הערך המבוקש.

```

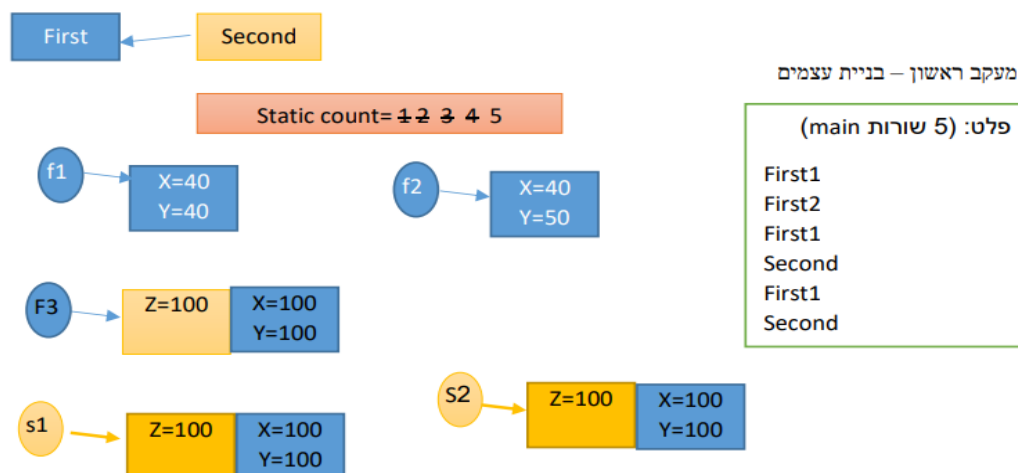
interface IPayment
{
    double GetPrice();
    string GetDetails();
}

```

שינוי המימשק

class Cash	public string GetDetails() => \$"Name: {name}";
class Credit	public string GetDetails() => \$"credit {creditNumber}";
class App	public string GetDetails() => \$"phone: {phoneNumber}";

שאלה 14



מעקב ראשון - בניית עצמים

```

פלט: (5 שורות main)
First1
First2
First1
Second
First1
Second
    
```

1. CW("Total=" + First.GetCount())	1 תקין. הפלט Total=5
2. CW ("Total=" + Second.GetCount())	2 תקין הפלט Total=5. (המחלקה קבלה הירושה את הפעולה)
3. CW("sum="+s1.sum())	3 מחזיר סכום x+y+z הפלט: sum=300 (הפעול וירטואלית - הפעולה של Second)
4. CW("sum="+f3.sum())	4 מחזיר סכום x+y+z הפלט: sum=300 (מופעלת הפעולה של העצם המוחזק בפועל Second)
5. s1=new First(100);	5 שגוי. בן אינו יכול להחזיק אב.
6. f1.Add(s2);	6 תקין. הפלט x=140 y=140
7. s1.Add(s2)	7 תקין. הפלט: x=200 y=200 z=200
8. s1.Add(f3)	8 תקין הפלט: x=200 y=200 z=200
9. ((First)s1).Add(f1);	9 תקין. פלט: x=140 y=140 z=100
10. s1= new Second(100,100);	10 שגוי. למחלקה Second אין בנאי המקבל 2 פרמטרים.

```

פלט: (המשך הרצה )
Total=5
Total=5
Sum=300
Sum=300
X=140 y=140
X=200 y=200 z=200
x=200 y=200 z=200
X=140 y=140 z=100
    
```

בהצלחה !