

## השאלות

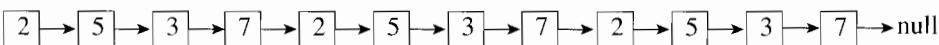
**בשאלוֹן זה שני פרקים: פרק ראשון ופרק שני.**  
**עליך לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.**

### **פרק ראשון (50 נקודות)**

ענה על שתיים מהשאלות 1-4 (לכל שאלה – 25 נקודות).

1. רשימה L תיקרא **משולשת** אם היא מקיימת את התנאים הבאים:

- \* הרשימה אינה ריקה.
  - \* מספר האיברים בה מחלק ב-3 בלי שארית.
  - \* האיברים בשליש הראשון של הרשימה מכילים את אותם ערכים שמכילים האיברים בשליש השני של הרשימה ואותם ערכים שמכילים האיברים בשליש השלישי של הרשימה. הערכים מסוורים באותו סדר בכל אחד מהשלשים.
- לדוגמה: הרשימה L שלפניך היא רשימה **משולשת** באורך 12.

L1: 

כתוב ב- C# או ב- Java פעולה חיצונית שתקבל רשימה L שהאיברים שלה הם מティפוס שלם.

אם L היא רשימה **משולשת**, הפעולה תחזיר true.  
 אחרת – הפעולה תחזיר false.

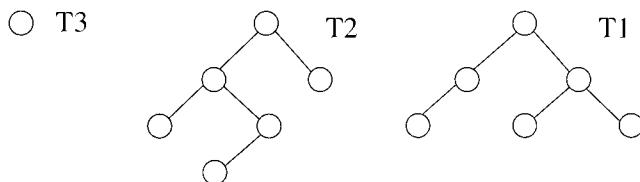
אתה יכול להשתמש בפעולות של המחלקות List < T >, Node < T > בלי למש אותן.

אם אתה משתמש בפעולות נוספות, עליך למש אותן.

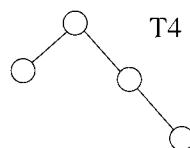
- .2. בשאלת זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.
- א. עץ ביניי ייקרא עץ **ימין-שמאל** אם לכל צומת בעץ אשר יש לו בן ימני, יש גם בן שמالي.

לדוגמא:

העצים  $T_1$ ,  $T_2$ ,  $T_3$ ,  $T_4$  הם עצים **ימין-שמאל**



והעץ  $T_4$  אינו עץ **ימין-שמאל**



כתבו ב- C# או ב- Java פעולה חיצונית שתקבל עץ ביניי  $T$  שהצמתים שלו הם מטיפוס שלם. הפעולה תחזיר true אם הוא עץ **ימין-שמאל**, אחרת — הפעולה תחזיר false.

אתה יכול להשתמש בפעולות המחלקה  $\text{BinTreeNode} < T >$  בלי למש אתן. אם אתה משתמש בפעולות נוספות, عليك למש אותן.

ב. אין קשר לסעיף א.)

נתון עץ ביניי  $T$  שהצמתים שלו מכילים ערכים מטיפוס two. אם סורקים את העץ  $T$  בסדר תחيلي (preorder), איזי סדר הערכים המתקיים מבחן בצלמים (משמאלי לימין) הוא:

X A I O N Y T D S

אם סורקים את העץ  $T$  בסדר תוכי (inorder), איזי סדר הערכים המתקיים מבחן בצלמים (משמאלי לימין) הוא:

I N O A X D T S Y

צייר את העץ  $T$ , ורשום את סדר הערכים המתקיים מבחן בצלמים אם סורקים את העץ בסדר סופי (postorder).

/המשך בעמוד 4/

פתרונות ב- C#

לפיניך 2 פועלות חיצונית הכתובות ב- C# .

הפעולה מקבלת תור לא ריק, המכיל מספרים שלמים. /\* \*

/\* הפעולה מחזירה... \*/

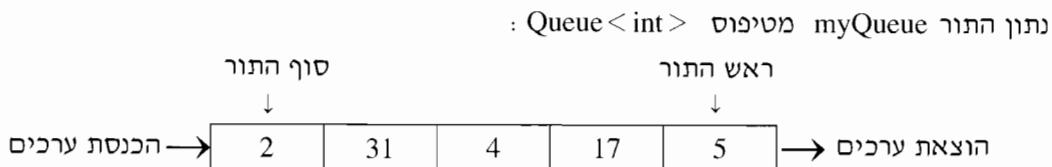
```
public static int Sod1 (Queue<int> q)
{
    int i = q.Remove();
    int result = i;

    if (!q.IsEmpty())
    {
        int j = Sod1 (q);
        if (result > j)
            result = j;
    }
    q.Insert(i);
    return result;
}
```

הפעולה מקבלת מספר שלם גדול מ- 0 או שווה לו /\* \*  
/\* הפעולה מחזירה... \*/

```
public static int Sod2 (int i)
{
    if (i == 0)
        return 0;
    int a = i % 10;
    int b = Sod2 (i / 10);
    if (a > b)
        return a;
    return b;
}
```

/המשך בעמוד 7/



א. מה יחזיר היזיון Sod1(myQueue) ? רשות את המעקב.  
רשות את התוור המתתקבל בתום המעקב. ציין את ראש התוור ואת סוף התוור.

ב. מה מבצעת הפעולה Sod1(queue) ? בעבור תוור queue לא ריק

מטיפוס Queue<int>

- ג. מה יחזיר היזיון Sod2(17852) ? רשות את המעקב.  
ד. מה מבצעת הפעולה Sod2(k) ? בעבור מספר k גדול מ- 0 מטיפוס שלם?  
ה. מה מבצעת הפעולה Sod2(Sod1(queue)) ? בעבור תוור queue לא ריק

מטיפוס Queue<int> המכיל רק מספרים שלמים וגדולים מ- 0 ?

## השאלות

בשאלון זה שני פרקים: פרק ראשון ופרק שני.  
עליך לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.

### פרק ראשון (50 נקודות)

**שים לב:** בכל שאלה שנדרש בה מימוש אתה יכול להשתמש בפעולות של המחלקות Stack<T>, Queue<T>, List<T>, Node<T>  
אם אתה משתמש בפעולות נוספות عليك למשם אותן.

ענה על שתיים מהשאלות 1-4 (לכל שאלה – 25 נקודות).

1. כתוב פעולה חיצונית המקבלת שתי מחסניות st1, st2 המכילות מספרים שלמים וגדולים מ-0. הפעולה תחזיר את **סכום** של זוג האיברים **הסמוכים** הקרוב ביותר בראש המחסנית st1, שסכום גודל מהסכום של כל זוג איברים סמוכים במחסנית st2. אם אין זוג כזה – הפעולה תחזיר 0.  
הנה שבכל אחת מהמחסניות יש לפחות שני איברים.

לדוגמא:

2	7
8	9
4	12
13	8
4	14
1	6
9	7
11	
	st1
	st2

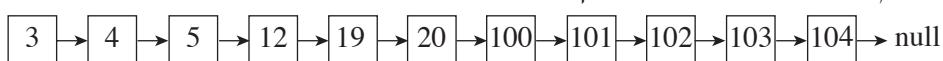
סכום האיברים הסמוכים 9 ו-12 במחסנית st1 גודל יותר מאשר סכום כל שני איברים סמוכים במחסנית st2, ולכן הפעולה תחזיר 21.

**שים לב:** במחסנית st1 יש גם זוג איברים סמוכים שסכוםם גדול יותר מ-21, אך הוא מופיע עמוק יותר במחסנית מזוג האיברים הסמוכים 9 ו-12.

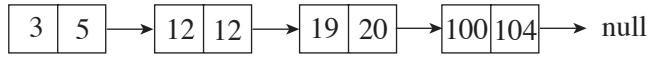
הערה: אין צורך לשמר על תוכנן המקורי של המחסניות.  
/המשך בעמוד 3/

L היא רשימה המכילה מספרים שלמים שונים זה מזה וממיינים בסדר עולה.  
**רשימת הטוחחים של L** היא רשימה חדשה שנבנית באופן הזה: בעבור כל רצף של מספרים עוקבים ב- L יהי רשימת הטוחחים איבר אחד שמכיל שני מספרים. מספר אחד הוא המספר הקטן ביותר ברצף, והמספר השני הוא המספר הגדול ביותר ברצף. רצף יכול להיות באורך 1 או יותר. אם הרצף הוא באורך 1, הוא מיוצג **רשימת הטוחחים על ידי איבר שני המספרים בו שוים.**

לדוגמה, בעבור הרשימה L שלפניך:



**רשימת הטוחחים של L** תהיה:



לפניך תיאור חלקו של המחלקה **RangeNode**, המיצגת איבר **רשימת הטוחחים**.

RangeNode
private int from; // המספר הקטן ביותר ברצף
private int to; // המספר הגדול ביותר ברצף
public RangeNode(int from, int to)

ממש ב- Java או ב- C# פעולה חיצונית שתקבל רשימה לא ריקה, המכילה מספרים שלמים שונים זה מזה וממיינים בסדר עולה, ותחזיר את **רשימת הטוחחים** שלה.

cotreta הפעולה ב- Java היא:

```
public static List<RangeNode> createRangeList(List<Integer> sourceList)
```

cotreta הפעולה ב- C# היא:

```
public static List<RangeNode> CreateRangeList(List<int> sourceList)
```

הנה שלכל אחת מהתכונות במחלקה **RangeNode** יש פועלות get ו- set. אתה יכול להשתמש בפעולות אלה וב פעולה הבונה של המחלקה **RangeNode** בלי למש אותה. אם אתה משתמש בפעולות נוספות, عليك למש אותה.

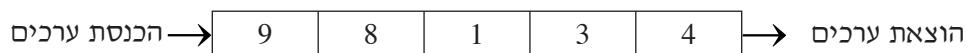
C# לפתרים ב-

. לפניך 2 פעולות חיצונית הכתובת ב- C#

```
public static void Sod1 (Queue<int> qu , Stack<int> st)
{
    if (! (qu.IsEmpty()))
    {
        int x = qu.Remove();
        Sod1(qu , st);
        st.Push(x);
        qu.Insert(x);
    }
}

public static void Sod2 (Queue<int> qu , Stack<int> st)
{
    if (! (qu.IsEmpty()))
    {
        int x = qu.Remove();
        Sod2(qu , st);
        int y = st.Pop();
        qu.Insert(x + y);
    }
}
```

נתון התור Queue<int> myQueue מטיפוס



ונתנו קטע הוכנית:

```
Stack<int> myStack = new Stack<int>();
Sod1(myQueue, myStack);
Sod2(myQueue, myStack);
```

א. (1) עקוב אחר הביצוע של קטע הוכנית ורשות את המעקב.

(2) ענה על התת-סעיפים (i)-(ii) על פי המעקב שרשמה.

(i) כתוב מה הכליל התור Queue<int> myQueue ומה הכלילה המחסנית myStack

Sod1 (myQueue, myStack); לאחר חזימון:

(ii) כתוב מה מכיל התור Queue<int> myQueue ומה מכילה המחסנית myStack

בסוף קטע הוכנית.

ב. מה מבצעת הפעולה Sod1 בעברית תור כלשהו לא ריק Queue<int> myQueue מטיפוס

?Stack<int> myStack ריקה מטיפוס Queue<int>

ג. מה מבצע קטע הוכנית בעברית תור כלשהו לא ריק Queue<int> myQueue מטיפוס

?Stack<int> myStack ריקה מטיפוס Queue<int>

- א. לפניך כמה מחלקות. עקב אחר הפעולה Main שבמחלקה Program ורשות את הפלט. במקבב יש לכלול את ערכיו המשתנים, ובמעבר כל עצם – את ערכי התוכנות שלו.

```

public class One
{
    private double n;
    public One(double n) { this.n = n; }
    public double GetN() { return this.n; }
    public void F() { Console.WriteLine("F of one "); }
    public void G() { Console.WriteLine("G of one " + this.GetN()); }
}

public class Two
{
    private Queue<One> q;
    public Two(int n, int m)
    {
        this.q = new Queue<One>();
        for (int i = n; i < m; i++)
            this.q.Insert(new One(Math.Pow(2, i)));
    }

    public void F()
    {
        Console.Write("F of two ");
        if (!this.q.IsEmpty())
            Console.WriteLine(this.q.Remove().GetN());
    }
}

```

```

public class Together
{
    private double x;
    private int from , to;
    private One first;
    private Two second;

    public Together(double x , int from , int to)
    {
        this.x = x;
        this.from = from;
        this.to = to;
        this.first = new One(x);
        this.second = new Two(from , to);
    }

    public void MethodA()
    {
        Console.WriteLine("-- MethodA() -- ");
        this.first.F();
    }

    public void MethodB()
    {
        Console.WriteLine("-- MethodB() -- ");
        this.first.G();
        this.second.F();
    }
}

public class Program
{
    public static void Main(string[] args)
    {
        Together tg = new Together(5.0 , 2 , 6);
        tg.MethodA();
        tg.MethodB();
    }
}

```

**ב.** בהנחה כי הפעולה הבונה במחלקה Together מקבלת מספרים גדולים מ- 0 בלבד – מה מבצעת הפעולה () F במחלקה Two ? /המשך בעמוד 6/

פתרונות ב- C#

לפניך ממשק של המחלקה "**תור-ביטול**" (**UndoQueue**) :

המחלקה מדירה טיפוס אוסף עם פרוטוקול FIFO להכנסה והוצאה של ערכים שלמים וגדולים מ- 0.

שם הפעולה	תיאור
UndoQueue()	הפעולה בונה <b>תור-ביטול</b> ריק.
bool IsEmpty()	הפעולה מחזירה true אם <b>התור-ביטול</b> הנוכחי ריק, אחרת — הפעולה מחזירה false.
void Insert(int x)	הפעולה מכניסה את הערך x לסוף <b>התור-ביטול</b> הנוכחי.
int Remove()	הפעולה מוציאה את הערך שבראש <b>התור-ביטול</b> הנוכחי, ומחזירה אותו. הנחה: <b>התור-ביטול</b> הנוכחי אינו ריק.
int Head()	הפעולהמחזירה את הערך של האיבר שבראש <b>התור-ביטול</b> בלי להוציאו. הנחה: <b>התור-ביטול</b> הנוכחי אינו ריק.
void Undo()	הפעולה מבטלת את פעולה ה- Remove או את פעולה ה- Insert האחרונה שבוצעה, ומשוחררת את <b>התור-ביטול</b> כפי שהיא לפני ביצוע הפעולה. אם אין פעולות Remove או Insert שבוצעו, הפעולה אינה עושה דבר. <u>שים-לב:</u> הפעלה זו פעמים של הפעולה Undo, תזריר את <b>התור-ביטול</b> למצב שבו הוא היה לפני ביצוע ח-פעולות Remove או Insert.

דוגמאות:

הפעולה	מצב התוור-ביטול לאחר ביצוע הפעולה
UndoQueue uq = new UndoQueue();	התוור-ביטול ריק
uq.Insert(2);	→2→ התוור-ביטול ריק
uq.Insert(3);	→3 , 2→ התוור-ביטול ריק
uq.Remove();	→3→ התוור-ביטול ריק
uq.Insert(4);	→4 , 3→ התוור-ביטול ריק
uq.Remove();	→4→ התוור-ביטול ריק
uq.Undo();	→4 , 3→ התוור-ביטול ריק
uq.Undo();	→1 , 4 , 3→ התוור-ביטול ריק
uq.Undo();	→4 , 3→ התוור-ביטול ריק
uq.Undo();	→3→ התוור-ביטול ריק
uq.Undo();	→3 , 2→ התוור-ביטול ריק
uq.Undo();	→2→ התוור-ביטול ריק
uq.Undo();	התוור-ביטול ריק
uq.Undo();	התוור-ביטול ריק

א. לפניך קטע קוד.

```
UndoQueue q = new UndoQueue();
q.Insert(1);
q.Insert(2);
q.Insert(3);
q.Remove();
q.Insert(4);
q.Undo();
q.Undo();
```

הראה את מצב התוור-ביטול לאחר הביצוע של כל הוראה בקטע הקוד.

כתב ב- C# את כוורתת המחלקה **UndoQueue**, ואת התכונות שלה. רשום תיעוד לכל תכונה.ממש ב- C# את הפעולות **Insert**, **Remove** ו- **Undo** המוצגות במשק המחלקה **UndoQueue**.אתה יכול להשתמש בפעולות המשק האחרות של **UndoQueue** בלי למש אותן. אם אתה משתמש בפעולות נוספות, عليك למש אותן. המשך בעמוד 14 /

.4. א. לפניך כותרת של פעולה: ב- Java :

```
public static void leaves(BinTreeNode<Integer> t , Stack<Integer> s)
```

ב- C# :

```
public static void Leaves(BinTreeNode<int> t , Stack<int> s)
```

הפעולה מקבלת עץ BinTree Node t של מספרים שלמים, ומחסנית ריקה s של מספרים שלמים.

הפעולה מכניסה למחסנית את ערכי כל העלים של העץ t , על פי סדר סריקה מימין לשמאל.

משה ב- Java או ב- C# את הפעולה.

כתב ב- Java או ב- C# פעולה בוליאנית שתקבל 2 עצים ביןיריים לא ריקים:

של מספרים שלמים, ותחזיר true אם מתקיימים שני התנאים האלה:

— יש להם אותו מספר עלים

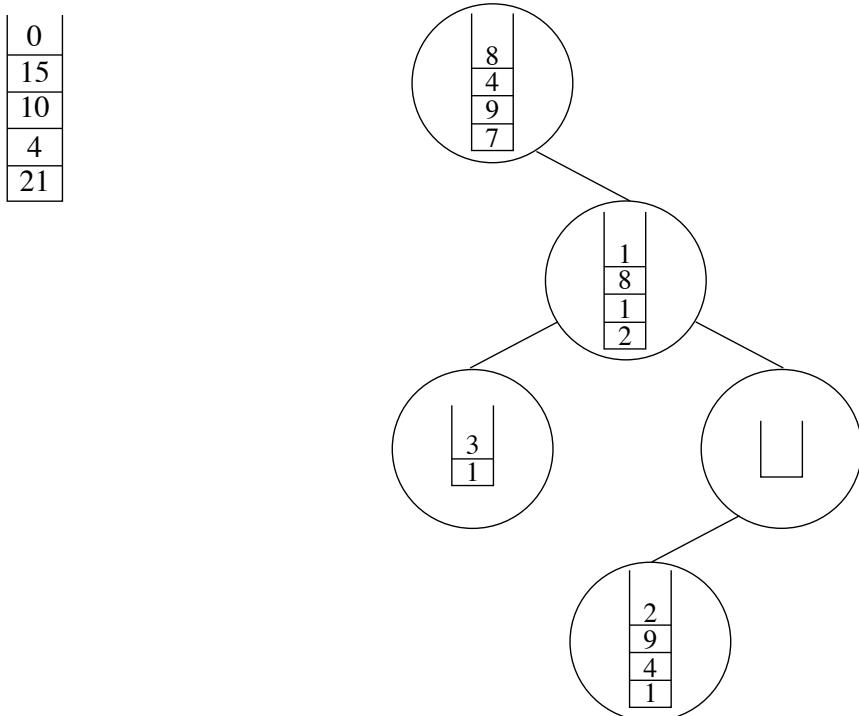
— על פי סדר הסריקה מימין לשמאל, ערכי העלים שווים  
אחרת — הפעולה תחזיר false.

עליך להשתמש בפעולת ש망שת בסעיף א.

- .2. א. כתוב פוליה חיצונית שתקבל עץ ביןари  $t$  לא ריק, שבו כל צומת מכיל מחסנית של מספרים שלמים גדולים מ-0.
- הפעולה תחזיר מחסנית. עבור כל צומת בעץ  $t$  יוכנס אליו למחסנית המוחזרת באופן הזה:
- אם במחסנית יש איבר אחד, יוכנס ערכו למחסנית שתוחזר. אם במחסנית שבצומת יש שני איברים, יוכנס סכוםם למחסנית שתוחזר. אם המחסנית שבצומת ריקה, יוכנס 0 למחסנית שתוחזר. במקרה אחר יוכנס למחסנית שתוחזר הסכום של שלושת האיברים העלילונים של המחסנית שבצומת.
- סדר האיברים במחסנית שתוחזר יהיה לפי סריקה בסדר תוכי (inorder) של צומתי העץ  $t$ .
- אין צורך לשמר את תוכן המחסניות שבצומתי העץ  $t$ .
- לדוגמה עבור העץ  $t$ :

המחסנית שתוחזר:

$t$



- .ב. מהי סיבוכיות זמן הריצה של הפעולה שכותבת בסעיף א? נמק את תשובהך.

לפתרונות ב- C#

ע"ז טרינארי הוא ע"ז שיש בו לכל היותר שלושה בניים לצומות, בן שמאל, בן אמצעי ובן ימני.  
לפניך ממשק חלקו של המחלקה "צומת טרינארי" `TriTreeNode<T>`.  
המחלקה מגדרה צומת טרינארי שבו ערך מטיפוס `T` ושלוש הפניות לצומתים טרינאריים.

שם הפעולה	תיאור
<code>TriTreeNode( T x )</code>	הפעולה בונה צומת טרינארי שערךו יהיה <code>x</code> , וערך שלוש הפניות יהיה <code>null</code> .
<code>TriTreeNode( TriTreeNode&lt;T&gt; left, TriTreeNode&lt;T&gt; middle, TriTreeNode&lt;T&gt; right, T x )</code>	הפעולה בונה צומת טרינארי שערךו יהיה <code>x</code> . <code>left</code> – הפניה לבן שמאל, <code>middle</code> – הפניה לבן אמצעי, ו <code>right</code> – הפניה לבן ימני. ערכי הפניות יכולות להיות <code>null</code> .
<code>T GetInfo()</code>	הפעולה מחזירה את הערך של הצומת הטרינארי.
<code>void SetInfo(T x)</code>	הפעולה משנה את הערך השמור בצוות הטרינארי ל- <code>x</code> .
<code>TriTreeNode&lt;T&gt; GetLeft()</code>	הפעולהמחזירה את הבן השמאלי של הצומת הטרינארי. אם אין בן שמאל, הפעולה מחזירה <code>null</code> .
<code>TriTreeNode&lt;T&gt; GetMiddle()</code>	הפעולהמחזירה את הבן האמצעי של הצומת הטרינארי. אם אין בן אמצעי, הפעולה מחזירה <code>null</code> .
<code>TriTreeNode&lt;T&gt; GetRight()</code>	הפעולהמחזירה את הבן הימני של הצומת הטרינארי. אם אין בן ימני, הפעולה מחזירה <code>null</code> .

א. כתוב ב- C# פעלת החיזונית Big שתתקבל ע"ז טרינארי `t` של מספרים שלמים וגדולים מ- 0, ותחזיר את המספר הגדל ביותר בע"ז. אם הע"ז ריק, הפעולה תחזיר -1.

ב. לפניך כוורתה של פעולה:

```
public static bool NoThree(TriTreeNode<int> tr)
```

הפעולה מקבלת ע"ז טרינארי `tr` של מספרים שלמים ומחזירה `true`, אם בכל אחד מהצומטים שבעץ יש לכל היותר שני בניים. אחרת – הפעולה מחזירה `false`. אם הע"ז ריק, הפעולה תחזיר `true`. ממש את הפעולה ב- C#.

הערה: אתה יכול להשתמש בפעולות המחלקה `TriTreeNode` בלי למש אותה.

אם אתה משתמש בפעולות נוספות, عليك למש אותה.  
המשך בעמוד 8/

C# ב- לפוטרים

בשאלה זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

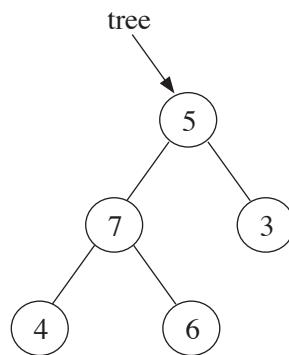
א. לפני הפעולה Amir .

```
public static void Amir(BinTreeNode<int> t, int x)
{
    if (t != null)
    {
        Amir(t.GetLeft() , x + 1) ;

        if ((t.GetLeft() == null) && (t.GetRight() == null))
            t.SetInfo (x) ;

        Amir(t.GetRight() , x + 1) ;
    }
}
```

לפני העץ tree .



i. רשום את העץ tree שיתקבל לאחר הפעלת הזימון: Amir(tree , 4)

ii. מה מבצעת הפעולה Amir בעבור עץ קלשו ו-  $x = 0$  ?

(שים לב: סעיף ב בעמוד הבא.)

.ב. (אין קשר לסעיף א.)

נתון עץ חיפוש ביןארי  $t$ , המכיל לפחות שלושה צמתים שערכיהם:  $x$  ,  $y$  ,  $z$  , המקיימים את הדרישה:  $x < y < z$ .

(1) אם  $x < y < z$  הם ערכי שלושה צמתים עוקבים בסיריקה תווכית (inorder) אז  $x = z$ .

(2) אם  $x < y < z$  הם ערכים של שני עליים ב- $t$  (משמאל לימין), אז  $y > x$ .

לכל אחת מהטענות (1)-(2) בחר באפשרות המתאימה מבין שלוש האפשרויות שלפניך:

- הטענה תמיד נכונה. נמק את תשובתך.

- הטענה תמיד אינה נכונה. נמק את תשובתך.

- הטענה נכונה במקרה חלק מהmarker. הבא דוגמה לקרה שבו הטענה נכונה, ודוגמה לקרה שבו הטענה אינה נכונה.

## השאלות

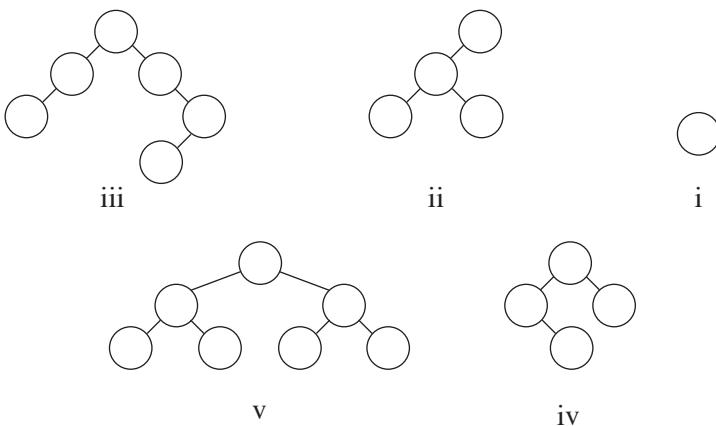
בשאלון זה שני פרקים: פרק ראשון ופרק שני.  
עליך לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.

### פרק ראשון (50 נקודות)

**שים לב:** בכל שאלה שנדרש בה שימוש אתה יכול להשתמש בעולות של המחלקות רשיימה, תור, מחסנית, עץ בינרי וחוליה, בלי למשם אותן. אם אתה משתמש בעולות נוספות, עליך למשם אותן.

ענה על שתים מהשאלות 1-4 (לכל שאלה – 25 נקודות).

1. **עץ בינארי מאוזן** הוא **עץ בינארי שבכל צומת שלו הערך המוחלט של הפרש הגבהים בין התת-עץ הימני לתת-עץ השמאלי הוא לכל היותר 1.**
- א. לפניך 5 עצים בינאריים – a. לכל אחד מן העצים, קבע אם הוא **עץ בינארי מאוזן** או אינו **עץ בינארי מאוזן**. אם העץ **אינו עץ בינארי מאוזן**, העתק אותו למחרתך וסמן x בצומת שմפר את האיזון.



- ב. כתוב ב- Java או ב- C# פעלת שתתקבל **עץ בינארי** כלשהו, ותחזיר true אם הוא **עץ בינארי מאוזן**, אחרת – הפעלה תחזיר false.
- אם הParmeter שמועבר לפעלת הוא null – הפעלה תחזיר true.

פתרונות ב- C#

המחלקה **Collec** היא אוסף של מספרים שלמים וגדולים מ- 0. לאוסף זה אפשר להוסיף איבר רק אם אין בו איבר אחר גדול ממנו.

- א. כתוב ב- C# את כוורתה המחלקה **Collec** ואת התכונות שלה.
- ב. כתוב ב- C# שתי פועלות בונות למחלקה **Collec** :
  - פעללה בונה, בלי פרמטרים, היוצרת אוסף ריק.
  - פעללה בונה שתתקבל מספר שלם  $n$  גדול מ- 0, ותיצור אוסף שיש בו איבר אחד שערךו  $n$ .
- ג. ממש ב- C# במחלקה **Collec** פעללה **Add**, שתתקבל מספר שלם וגדול מ- 0 ותוסיף אותו לאוסף, אם אפשר. אם המספר צורף לאוסף, הפעולה תחזיר `true`. אחרת — הפעולה תחזיר `false`.
- ד. ממש ב- C# במחלקה **Collec** פעללה **Small**, שתחזיר את המספר הקטן ביותר באוסף.
- ה. ממש ב- C# במחלקה **Collec** פעללה בשם **Smallest**, שתתקבל עצם `c` מטיפוס **Collec**, ותחזיר את המספר הקטן ביותר מבין שני האוספים — האוסף הנוכחי ו- `c`. הנה כי שני האוספים אינם ריקים.

4. **שים לב:** לשאלת זו שני נוסחים:  
 נוסח אחד לפתרים על פי התכנית הוכחית (ב- Java וב- C# – עמודים 7-11),  
 ונוסח אחד לפתרים על פי התכנית החדשה (ב- Java וב- C# – עמודים 12-16).  
עובד על פי התכנית שלמדת.

לפתרנים על פי התכנית החדשהבשאלה זו שני סעיפים א-ב, שאון קשור ביניהם. עזה על שניהם.א. נתונה פעללה המתקבלת רישומה *lst* של מטורים שלמים:

C#	Java
<pre>public static int What(Node&lt;int&gt; lst) {     if (lst == null)         return 0;     int x = lst.GetValue();     lst = lst.GetNext();     if (x &gt;= 0)         return What(lst); //(*)     return 1 + What(lst); }</pre>	<pre>public static int what(Node&lt;Integer&gt; lst) {     if (lst == null)         return 0;     int x = lst.getValue();     lst = lst.getNext();     if (x &gt;= 0)         return what(lst); //(*)     return 1 + what(lst); }</pre>

נתונה הרשימה *lst: -2 → -9087 → 16 → -43 → 5 → ||*(1) עקב אחר הפעולה הנתונה בעבור הרשימה *lst*.רשום את הערך שיזוזו, וסדרת את הרשימה *lst* לאחר ביצוע הפעולה.

(2) מה הפעולה מבצעת?

(3) החליטו את החראה המסומנת ב- (\*) להראה: `return 0;`

i. מה תבצע הפעולה לאחר ההחלפה?

ii. מה תכיל הרשימה *lst* לאחר ביצוע הפעולה?

(שים לב: סעיף ב בעמוד הבן.)

C# – לפוטרים ב-

לפניך הפעולה Sum הכתובת במחילה ראשית.

```
public static int Sum (Stack<int> s)
{
    if(s.IsEmpty())
        return 0;
    int x = s.Pop();
    if(x%6 == 0)
        return x + Sum(s);
    return Sum(s);
}
```

נתונה המחסנית s .

ראש המחסנית →	12
	4
	33
	6
	30
	0

עקב אחר הפעולה Sum בעבור המחסנית s , ורשות את הערך שיווצר. (1)

במשך הראה את המעבר על המחסנית s .

לפניך הפעולה Sod (2)

```
public static void Sod(Queue<Stack<int>> qq, Queue<int> qm)
{
    if (!qq.IsEmpty())
    {
        int x = Sum(qq.Remove());
        qm.Insert(x);
        Sod(qq , qm);
    }
}
```

/המשך בעמוד 16/

(שים לב: המשך תתי-סעיף (2) בעמוד הבא.)

לפניך קטע תכנית המשתמש בפעולה `Sod`.  
סרטט במחברתך את כל אחד משני התווים `q` ו- `qr`, לפני הקריאה לפעולה `Sod` ואחריו ביצוע הפעולה `Sod`.

```
public static void Main(string[] args)
{
    Stack<int> s1 = new Stack<int>();
    Stack<int> s2 = new Stack<int>();

    Queue<Stack<int>> q1 = new Queue<Stack<int>>();
    s1.Push(0);
    s1.Push(30);
    s1.Push(6);
    s1.Push(33);
    s1.Push(4);
    s1.Push(12);
    s2.Push(23);
    s2.Push(36);
    s2.Push(1);
    q1.Insert(s1);
    q1.Insert(s2);
    Queue<int> qr = new Queue<int>();
    Sod(q1 , qr);
}
```

## השאלות

בשאלון זה שני פרקים: פרק ראשון ופרק שני.

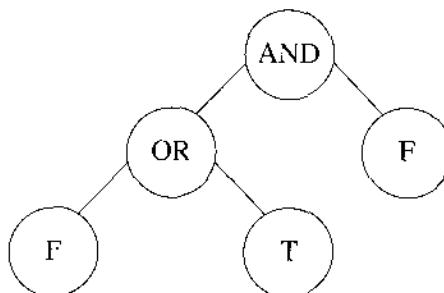
עליך לענות על שאלות משני הפרקים, לפי ההוראות בכל פרק.

### פרק ראשון (50 נקודות)

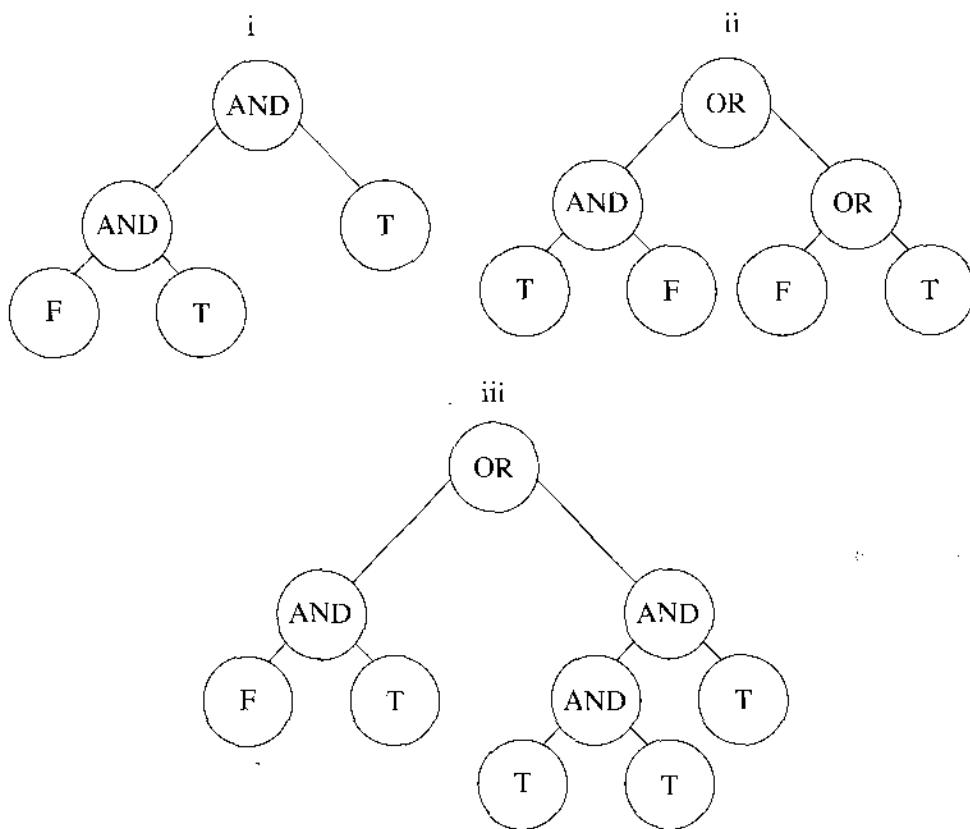
**שים לב:** בכל שאלה שנדרש בה שימוש אתה יכול להשתמש בפעולות של המחלקות רשימה, תור, מתרנית, עץ בינרי וחוליה, בלי למשוך אותן. אם אתה משתמש בפעולות נוספות, עליך למשוך אותן.

ענה על שתיים מהשאלות 1-4 (לכל שאלה – 25 נקודות).

1. **ע"ז ביטוי בولיאני** הוא ע"ז ביןאי לא ריק מטיפוס **מחירות**, המייצג ביטוי בוליאני, **true**, בכל עלה שלו נמצאת אחת מן המחריות: "T" או "F". המחריות "T" מייצגת **true**, והמחריות "F" מייצגת **false**.
- בכל צומת שאינו עלה נמצאת אחת מן המחריות: "AND" או "OR".
- המחריות "AND" מייצגת את הפעולה הבוליאנית "זגמ", והמחריות "OR" מייצגת את הפעולה הבוליאנית "או".
- לכל צומת שאינו עלה יש שני בנים.
- כדי לחשב את הביטוי הבוליאני שהע"ז מייצג, מפעילים את הפעולה הבוליאנית שבסכום שאינו עלה על הערכים המתקבלים מהתתי-יעדים השמאלי והימני של צומת זה.
- לדוגמה: הביטוי הבוליאני שמייצג הע"ז שלפניך הוא  $(F \text{ OR } T) \text{ AND } (F)$  וערךו הוא **false**.



- א. לפניך שלושה עצים – נז'ן שכל אחד מהם הוא עץ ביטוי בוליאני.  
 לכל אחד מן העצים כתוב את הביטוי הבוליאני שהוא מייצג, ואת הערך המתתקבל ממנו.



- ב. כתוב ב- Java או ב- C# פעלת חישונית שתקבל עץ ביטוי בוליאני, ותחזיר את הערך הבוליאני (או true או false) של הביטוי שהעץ מייצג.

/המשך בעמוד 4/

**פרק שני (50 נקודות)**

**שים לב:** בכל שאלה שנדרש בה מימוש אתה יכול להשתמש בפעולות של המחלקות תור, מחסנית, עץ, בירוי וחוליה, בלי למשוך אותן. אם אתה משתמש בפעולות נוספות נוספת, עליך למשוך אותן.

ענה על **שתיים** מהשאלות 4-6 (לכל שאלה → 25 נקודות).

4. לפניה המחלקה טבעת — **Ring** שליה שתי תכונות:

גודל הטעעת מטיפוס מחזרות ("S" – טבעת קטנה, "L" – טבעת גדולה);

מספר שלם המיצג את צבע הטעעת.

**Java**

```
public class Ring
{
    private String size; // גודל הטעעת
    private int color; // צבע הטעעת
    public Ring()
    {
        this.size = "L";
        this.color = 0;
    }
    public Ring(String str, int c)
    {
        this.size = str;
        this.color = c;
    }
    public String getSize()
    {
        return this.size;
    }
    public int getColor()
    {
        return this.color;
    }
}
```

**C#**

```
public class Ring
{
    private string size; // גודל הטעעת
    private int color; // צבע הטעעת
    public Ring()
    {
        this.size = "L";
        this.color = 0;
    }
    public Ring(string str, int c)
    {
        this.size = str;
        this.color = c;
    }
    public string GetSize()
    {
        return this.size;
    }
    public int GetColor()
    {
        return this.color;
    }
}
```

## לפניך ממשק המחלקה מוט – Pole

Java	C#	הפעולה
public Pole()	public Pole()	פעולה הבונה מוט ריק. סיבוכיות זמן הריצה של O(1) הפעולה היא (1)
public void add(Ring r)	public void Add(Ring r)	פעולה המכניתה טבעת r לראש המוט. סיבוכיות זמן הריצה של O(1) הפעולה היא (1)
public Ring remove()	public Ring Remove()	פעולה המוחזירה את הטעבת שבראש המוט. בנוסף, הפעולה מציאה את הטעבת מראש המוט. סיבוכיות זמן הריצה של O(1) הפעולה היא (1)
public boolean isEmpty()	public bool IsEmpty()	אם המוט ריק הפעולה מחזירה true. אחרת – היא מחזירה false. סיבוכיות זמן הריצה של O(1) הפעולה היא (1)
public void sort()	public void Sort()	פעולה המסדרת את הטעבות של המוט כך: הטעבות הגדולות "מנוחות" בתחתיית המוט והטעבות הקטנות מעילן.

- א. ממש את הפעולה sort() ב- Java או Sort() ב- C#, שבמחלקה Pole.
- אתה יכול להשתמש בפעולות הנוספות של המחלקה Pole ביל' למשר Ott.
- בשותבון השתמש רק בפעולות המחלקות Ring ו- Pole.
- ב. מה סיבוכיות זמן הריצה של הפעולה שמיימת בסעיף א? נמק את תשובתך.

## לפותרים ב-C#

נגיד רשותה דואציוונית כאוסף סדר של חוליות מטיפוס `BinNode<int>` המקשורות כן:

לכל זוג חוליות `p1, p2` ברשימה, אם מתקיים `p1.GetRight() == p2`, אז מתקיים גם

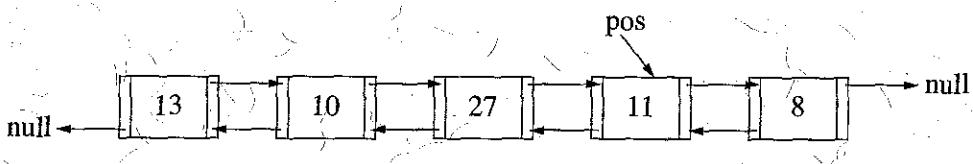
`p1 == p2.GetLeft()`, ברשימה דואציוונית יש לפחות שתי חוליות.

כלומר: כל חוליה ברשימה – חוץ מהחוליה שבקצה הימני של הרשימה והחוליה שבקצה השמאלי,

של הרשימה – מצביעה על החוליה שלפניה ועל החוליה שאחריה.

לפניך דוגמה לרשותה דואציוונית ומשתנה `pos` מטיפוס `BinNode<int>` המצביע על חוליה

לשיה ברשימה דואציוונית.



הפעולה `FirstLeft` מקבלת מצביע `pos` שונה מ-`null` מטיפוס `BinNode<int>` המצביע על

חוליה כלשהי ברשימה דואציוונית ומחזיר אותה החוליה השמאלית ביותר ברשימה.

הפעולה `FirstRight` מקבלת מצביע `pos` שונה מ-`null` מטיפוס `BinNode<int>` המצביע על

חוליה כלשהי ברשימה דואציוונית ומחזיר אותה החוליה הימנית ביותר ברשימה.

א. לפניך שلد של הפעולה `FirstLeft`

העתק אותו למחברתך והשלם אותו, כך שהפעולה תבצע את הנדרש.

`public static BinNode<int> FirstLeft(BinNode<int> pos)`

```
{
    while (_____)
        pos = _____;
    return _____;
}
```

/המשך בעמוד 11/

ב' לפניה הפעולה `What(BinNode<int> pos)` מקבלת מבצע לחוליה כלשהי ברשימה דורכיוונית ומוחזירה ערך בוליאני. הרשימה הדורכיוונית מכילה לפחות 3 חוליות.

- (1) עקוב אחר ביצוע הפעולה בעבור המשתנה `pos` והרשימה שבדוגמה המוצגת בתחילת השאלה.

במשך הראה את הרשימה הדורכיוונית ואת ערכי המשתנים `pos, left, right, sum`

```
public static bool What(BinNode<int> pos)
{
    BinNode<int> left = FirstLeft(pos);
    BinNode<int> right = FirstRight(pos);

    int sum = left.GetValue() + right.GetValue();
    left = left.GetRight();
    right = right.GetLeft();

    while ((left != right) && (left.GetRight() != right) &&
           (left.GetValue() + right.GetValue() == sum))
    {
        left = left.GetRight();
        right = right.GetLeft();
    }
    if (left == right)
        return right.GetValue() == sum;
}
```

```
if (left.GetRight() == right)
    return left.GetValue() + right.GetValue() == sum;
return false;
```

- (2) קבע אם אפשר או אי אפשר להחליף את 3 השורות האחרונות שבפעולה – השורות המוקפות בסגרת – בהוראה:

```
return left.GetValue() + right.GetValue() == sum;
```

נק את קבועה.

6. עץ מספרים הוא עץ בינארי לא ריק מטיפוס שלם, שהערכים בצלמתים שלו הם מספרים שלמים וגדולים מד' 0 והשונים זה מהו.

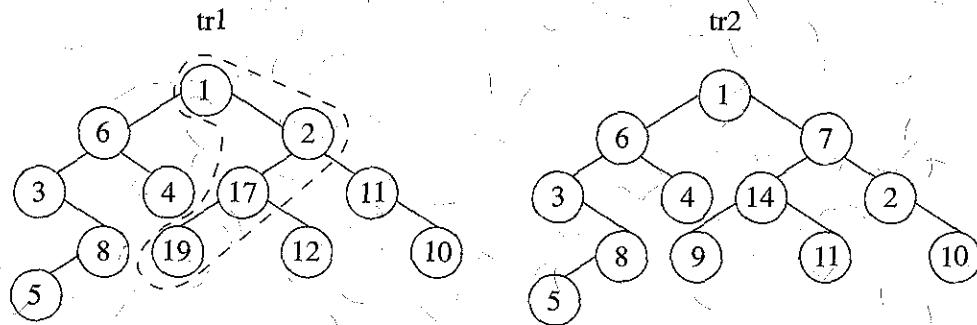
על עץ מספרים מוגדרת פעולה "מסלול-עליה", המחזירה  $\text{true}$  אם יש בעץ מסלול המתחיל בשורש העץ ומסתיים באחד העלים שלו, וערך הצלמתים ממוקנים בסדר עולה מהשורש לעלה.

אם אין מסלול נושא – הפעולה מחזירה  $\text{false}$ .

לדוגמה:

בעבור עץ מספרים  $\text{tr1}$  הפעולה "מסלול-עליה" מחזירה  $\text{true}$ , המסלול מוקף בקו שבור.

בעבור עץ מספרים  $\text{tr2}$  הפעולה "מסלול-עליה" מחזירה  $\text{false}$ .



ממש ב- Java או ב- C# את הפעולה "מסלול-עליה" בעבור עץ מספרים.

כותרת הפעולה ב- Java: `public static boolean upPath(BinNode<Integer> tr)`

כותרת הפעולה ב- C#: `public static bool UpPath(BinNode<int> tr)`

/המשך בעמוד 13/

**פרק שני (50 נקודות)**

**שים לב:** בכל שאלה שנדרש בה שימוש אתה יכול להשתמש בפעולות של המחלקות **תור, מחסנית, עץ ביני וחוליה**, בלי למשוך אותן. אם אתה משתמש בפעולות **נותפות, עליין** למשוך אותן.

ענה על **שתיים** מהשאלות 4-6 (לכל שאלה — 25 נקודות).

רשימה L תקרא **מושולשת** אם היא מקיימת את התנאים הבאים:

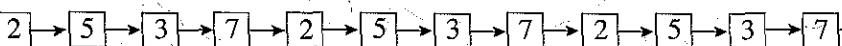
\* הרשימה אינה ריקה.

\* מספר האיברים בה מחלק ב-3 בלי שארית.

\* האיברים בשליש הראשון של הרשימה מכילים את אותם ערכים שמכילים האיברים בשליש השני של הרשימה ואוטם ערכים שמכילים האיברים בשליש השלישי של הרשימה.

הערכים מסודרים באותו סדר בכל אחד מהשלישים.

לדוגמא: הרשימה **L** **שלפניך** היא רשימה **מושולשת** באורך 12.

L1; 

בתוב ב-#C או ב- Java פעולה **חיצונית** שתקבל רשימה L שהאיברים שלה הם

מティפוס שלם.

אם L היא רשימה **מושולשת**, הפעולה תחזיר true .

אחרת — הפעולה תחזיר false .

המשך בעמוד 8/

- .ג. א. ממש פועלה חיצונית `exist` ב- Java או `Exist` ב- C#.
- הפעולה מקבלת עץ ב内幕י t מטיפוס שלם ומספר שלם x. הפעולה תחזיר true אם יש בעץ שערכו x, אחרת – הפעולה תחזיר false. אם העץ ריק – הפעולה תחזיר false.
- .ב. לפניה הפעולה ב- Java `check(t1, t2)` וב- C# `Check(t1, t2)`.
- הפעולה מקבלת שני עצים ב内幕ים לא ריקים מטיפוס שלם, t1 ו- t2, ומוציאת רשימה המכילה את כל המספרים הנמצאים בעץ t1 **ואינעם** נמצאים בעץ t2. הפעולה מזמנת פעולה נוספת המתקבלת שלושה פרמטרים.

Java

```
public static Node<Integer> check(BinNode<Integer> t1, BinNode<Integer> t2)
{
    Node<Integer> first = new Node<Integer> (-1);
    first = check(t1 ,t2 ,first);
    return first.getNext();
}
```

C#

```
public static Node<int> Check(BinNode<int> t1 ,BinNode<int> t2)
{
    Node<int> first = new Node<int> (-1);
    first = Check(t1 ,t2 ,first);
    return first.GetNext();
}
```

ממש את הפעולה:

ב- Java :

```
public static Node<Integer> check( BinNode<Integer> t1,
                                    BinNode<Integer> t2 ,Node<Integer> list)
```

או ב- C# :

```
public static Node<int> Check(BinNode<int> t1 ,BinNode<int> t2 ,Node<int> list)
```

אתה יכול להשתמש בפעולת שימושה שמיינשת בסעיף א.

- .ג. מה היא סיבוכיות זמן הריצה של הפעולה שמיינשת בסעיף ב ? נמק את תשובתך / המשך בעמוד 9/

**פרק שני (50 נקודות)**

**שים לב:** בכל שאלה שנדרש בה מימוש אתה יכול להשתמש בפעולות של המחלקות: **טור, מחסנית, עץ ביןרי וחוליה,** בלי למשם אותן. אם אתה משתמש בפעולות **נוספות, עליך** למשם אותן.

ענה על **שתיים** מן השאלות 4-6 (לכל שאלה — 25 נקודות).

4. א. כתוב פעולה חיצונית `lastAndRemove` ב-`C#`, `Java` או `C#`, המקבלת מחסנית מטיפוס `Slm`, מוחקת את האיבר התיכון במחסנית, ומוחזירה את ערכו. בסיום הפעולה האיברים האחרים במחסנית נשארים ללא שינוי. הנח שהמחסנית אינה ריקה.

לאחר הפעולה יוחזר הערך 9, והמחסנית תיראה כך:

1	←———— רأس המחסנית
6	
32	
5	
5	
7	
4	

לדוגמה, עבור המחסנית שלפנינו:

1	←———— רأس המחסנית
6	
32	
5	
5	
7	
4	
9	

(שים לב: סעיף ב של השאלה בעמוד הבא.)

/המשר בעמוד 7/

ב. נתונה המחלקה:

TwoItems	
int num1	תכונות:
int num2	
TwoItems (int number1, int number2)	פעולה בונה:

הנח שלכל תכונה הוגדרו ב- Java פועלות get ו- set וב- C# פועלות Get ו- Set .  
 כתוב פעולה חיצונית(stackTwoItems) ב- Java או C# , המקבלת מחסנית stk1  
 שאינה ריקה, מטיפוס שלם ובגודל זוגי, ומחזירה מחסנית מטיפוס TwoItems .

האיבר התיכון במחסנית המוחזרת יכול הפניה למופיע של TwoItems שהתכונה שלו num1 היא האיבר  
 שבראש המחסנית stk1 , והתכונה שלו num2num היא האיבר התיכון במחסנית stk1 .  
 האיבר שמעל האיבר התיכון במחסנית המוחזרת יכול הפניה למופיע של TwoItems שהתכונה שלו num1  
 היא האיבר שמתחתי לאיבר שבראש המחסנית stk1 , והתכונה שלו num2num היא האיבר שמעל האיבר התיכון  
 במחסנית stk1 , וכן הלאה, כך שהאיבר שבראש המחסנית המוחזרת יכול הפניה למופיע של TwoItems  
 שהתכונות שלו num1 ו- num2num הם שני איברים סמוכים, הנמצאים באמצע המחסנית 1 .



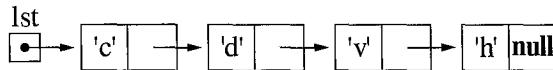
עליך להיעזר בפעולה שכתבת בסעיף א.  
 הערה: אין צורך לשמור על תוכן המקורי של מחסנית stk1 .

C# לשutrims בשפת

נתונה הפעולה Sod1 :

```
public static Node <char> Sod1(Node <char> lst, char ch)
{
    if (lst == null)
        return null;
    if (lst.GetValue() == ch)
        return lst;
    return Sod1(lst.GetNext(), ch);
}
```

א. (1) עקוב אחר הפעולה וכותב מה יוחזר עבור 'a' = ch וההפנייה lst לשרשרת חוליות של תווים:



(2) מהי מטרת הפעולה Sod1 ?

(3) מהי סיבוכיות זמן הריצה של הפעולה Sod1 ? נמק.

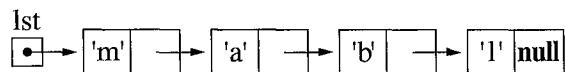
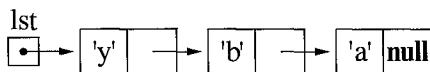
ב. נתונה הפעולה Sod2 :

```
public static bool Sod2(Node <char> lst)
{
    if (Sod1(lst,'a') != null && Sod1(lst,'b') != null)
        return true;
    return false;
}
```

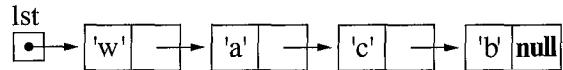
מהי מטרת הפעולה Sod2 ?

כתב פעללה בוליאנית המקבלת הפניה לשרשרת חוליות של תווים ומחזיר true אם מופיעות בה שתי חוליות סמוכות שערכיהן 'b' ו 'a' או 'a' ו 'b'. אחרת — הפעולה מחזירה false .

דוגמא לשרשרת חוליות שמופיעים בה 'b' ו 'a' בראצ':



דוגמא לשרשרת חוליות ש-'b' ו 'a' ו 'b' אינם מופיעים בה בראצ':



עליך להשתמש בפעולה Sod1 .

הנח שכל התווים שונים זה מזה.

.6 נתונה הפעולה:

**ב Java :**

```
public static boolean lessThanTree (BinNode <Integer> t, int x)
```

**ב C# :**

```
public static bool LessThanTree (BinNode <int> t, int x)
```

הפעולה מחזירה true אם x קטן מכל הערכים בעץ t. אחרת — הפעולה מחזירה false. סיבוכיות זמן הריצה של הפעולה היא  $O(n)$ . או מייצגת את מספר הצמתים בעץ t.

א. כתוב פעלת חיצונית treeLessThanTree ב- Java או C# TreeLessThanTree ב- C#, המתקבלת שני עצים בינאריים t1 ו- t2 של ערכים שלמים. נתון שב- t2 קיים לפחות צומת אחד. הפעולה מחזירה true אם כל ערך בעץ t1 קטן מכל אחד מהערכים בעץ t2, אחרת — הפעולה מחזירה false.

אם t1 הוא null — הפעולה תחזיר true.

אפשר להשתמש בפעולה הנתונה בלי למש אותה. אם אתה משתמש בפעולות אחרות, عليك למש אותה.

ב. מהי סיבוכיות זמן הריצה של הפעולה שכותבת בסעיף א? נמק.

לפותרים בשפת C#

לפניר הפעולה Secret1

```
public static bool Secret1 (int num, int digit )
{
    if ( num < 10 )
        return ( num % 2 == digit % 2 );
    if ( num % 2 != digit % 2 )
        return false;
    return Secret1 ( num / 10, digit );
}
```

- א. (1) כתוב מה יחזיר זימון הפעולה (937,5). Secret1 . עלייך להראות מעקב.  
 הבא דוגמה למספר num בעל 3 ספרות שעבורו זימון הפעולה Secret1 (num,5) יחזיר ערך שונה ממה שהתקבל בתת-סעיף א (1). עלייך להראות מעקב.  
 (2) כתוב במשפט אחד מה מביצעת הפעולה הבוליאנית Secret1 , בולם מהי השאלה שעלייה הפעולה מחזירה  
 (3) false או true

(שים לב: המשך השאלה בעמוד הבא.)

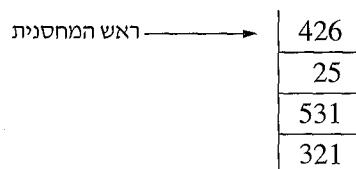
לפניך הפעולה Secret2:

```

public static bool Secret2 ( Stack <int> s )
{
    bool ok;
    int x;
    if ( s.IsEmpty () )
        ok = true ;
    else
    {
        x = s.Pop () ;
        if ( !(Secret1 (x, x % 10)) )
            ok = false;
        else
            ok = Secret2 (s);
    }
    return ok;
}

```

ב. (1) עבור המחסנית s שלפניך:



(1) כתוב מה תחזיר הפעולה Secret2 . עליך להראות מעקב (אין צורך להראות מעקב אחריו פועלות).

(2) כתוב במשפט אחד מה מבצעת הפעולה הבוליאנית Secret2 , כלומר מהי השאלה שעלייה הפעולה מחזירה

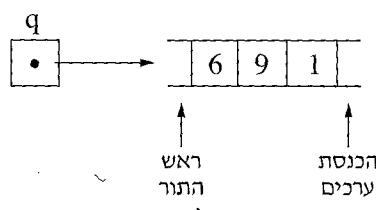
. false או true

.5. א. "תור מספר" הוא תור של ספרות בין 1 ל- 9 (כולל), המיצג מספר שלם — האיבר הראשון (**ראש התור**) הוא

**ספרת האחדות**, האיבר השני הוא ספרת העשרות וכן הלאה.

הנח שמספר הספרות האפשרי בתור לא גדול ממספר הספרות שיכל להכיל משתנה מטיפוס `int`.

לדוגמא: התור **שלפניך** מייצג את המספר 196.

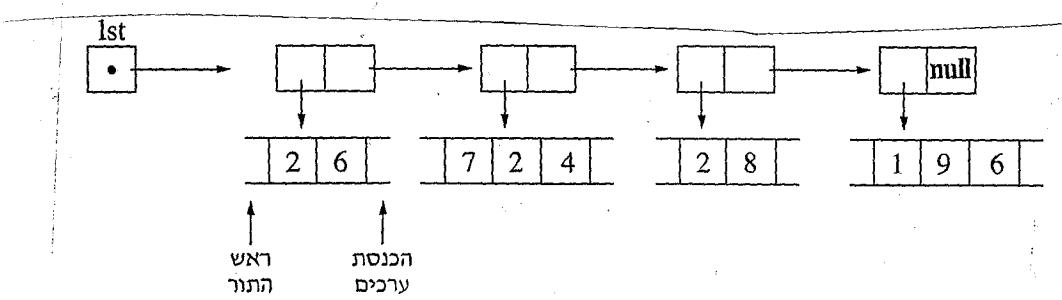


כתוב פוליה בשם `toNumber` ב- C# או `ToNumber` ב- Java או מקבלת "תור מספר" — `q`, ומחזירה את המספר המיוצג בתור.

**הערה:** אין חובה לשמור על מבנה התור.

.ב. נתונה שרשרת חוליות ובה כל חוליה מכילה **"תור מספר"**.

לדוגמא: שרשרת החוליות שלפניך מייצגת את המספרים 62 , 427 , 82 , 691 .



כתוב פוליה בשם `bigNumber` ב- C# או `BigNumber` ב- Java או מקבלת הפניה `lst` לשרשרת החוליות, וממחזירה את המספר הגדל ביותר המיוצג בשרשרת החוליות.

עבור שרשרת החוליות שתוארה בדוגמא הפוליה ממחזירה את המספר 691.

**חובה להשתמש בפועלה שהוגדרה בסעיף א.**

. 6. נתונה המחלקה **Range** שיש לה שתי תכונות:

low — מספר מטיפוס שלם.

high — מספר מטיפוס שלם.

high גדול מ- low .

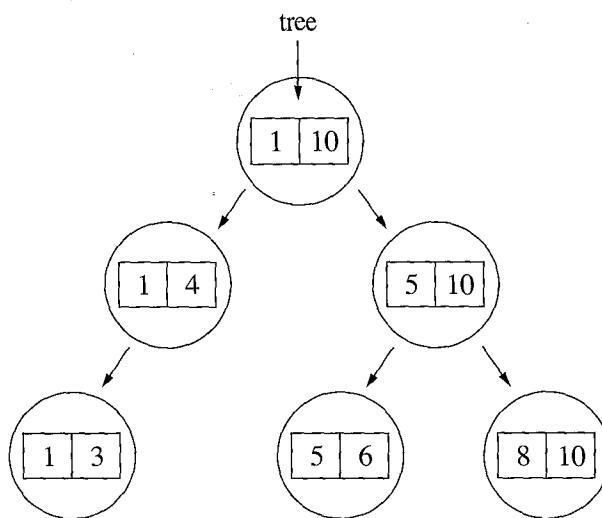
הנח שלכל תכונה הוגדרו ב- Java הפעולות get ו- set וב- C# הפעולות Get ו- Set .

ע"ז טוחים הוא ע"ז שאיבריו הם מטיפוס **Range** .

ע"ז טוחים מסודר הוא ע"ז ריק או ע"ז טוחים שבו עבור כל צומת מתקיימים התנאים הבאים:

- אם יש בן שמאל, אז ה- low של הצומת שווה ל- low של חבן השמאלי, וה- high של הצומת גדול או שווה ל- high של חבן השמאלי.
- אם יש בן ימני, אז ה- high של הצומת שווה ל- high של הבן הימני, וה- low של הצומת קטן או שווה ל- low של הבן הימני.
- אם יש שני בני, אז ה- high של הבן השמאלי קטן מה- low של הבן הימני.

דוגמה לע"ז טוחים מסודר:



כתב פועלה חיצונית בוליאנית בשם order ב- Java או Order ב- C# , מקבלת ע"ז טוחים או ע"ז ריק וממחזרה true אם העץ הוא ע"ז טוחים מסודר, אחרת — הפעולה מחזירה false .

## פרק שני (50 נקודות)

**שים לב:** בכל שאלה שנדרש בה מימוש אתה יכול להשתמש בפעולות של המחלקות: **תור, מחסנית, עץ בינרי וחוליה**, בלי למשן אותן. אם אתה משתמש בפעולות **נוספות**, עליך למשן אותן.

ענה על **שתיים** מן השאלות 4-6 (לכל שאלה – 25 נקודות).

- .**a.** כתוב פעולה חיצונית `isExist` בשפת Java או `IsExist` בשפת C#. הפעולה מקבלת מספר `num` – מטיפוס `שלם` בין 0 ל- 9 (כולל), ומחסנית `stk` – מטיפוס `שלם`. הפעולה תחזיר `true` אם יש במחסנית מספר שמספרת האחדות שלו שווה למספר `num`, אחרת היא תחזיר `false`. הנח שהמספרים במחסנית `stk` אינם שליליים.

**דוגמה:** עבור `num` שווה ל- 8 והמחסנית `stk` שלפניך:

162	ראש המחסנית
251	
568	
77	

הפעולה תחזיר `true`, משום שבמחסנית יש מספר 568 שמספרת האחדות שלו היא 8.  
**הערה:** חובה לשמור על מבנה המחסנית עם סיום הפעולה.

- .**b.** לשם פתרון סעיף זה בלבד תוכל להשתמש בפעולת שלפניך לא לצורך למשן:

כותרת הפעולה	תיאור הפעולה
<code>public static Stack &lt;Integer&gt; clone ( Stack &lt;Integer&gt; s )</code> – <b>בשפת Java</b>	הפעולה מקבלת מחסנית מטיפוס <code>שלם</code> ומחזירה העתק מדויק של המחסנית בלי לשנות את המחסנית המקורית.
<code>public static Stack &lt;int&gt; Clone ( Stack &lt;int&gt; s )</code> – <b>בשפת C#</b>	

**נדיב:** הספרה המשמעותית במספר היא הספרה היכי שמאלית שלו.

לדוגמה הספרה 3 היא הספרה המשמעותית במספר 32, והספרה 5 היא הספרה המשמעותית במספר 541.

כתב פעולה חיצונית `allExist` בשפת Java או `AllExist` בשפת C#, המקבלת מחסנית `stk` שאינה ריקה, מטיפוס `שלם`. הפעולה תחזיר `true` אם כל הספרות המשמעותיות במספרים שבמחסנית מופיעות בספרת האחדות במספרים כלשהם במחסנית, אחרת היא תחזיר `false`.  
 הנח שהמספרים במחסנית `stk` אינם שליליים.

דוגמה: עבור המחשבנית `st` שלפניך:

122	← ראש המחשבנית
251	
565	
12334	
28	
7	

הפעולה תחזיר `true` כי כל הספרות המשמעותיות במספרים שבמחשבנית — 7, 2, 5, 1 — מופיעות בספרת האחדות במספרים שבמחשבנית.

דוגמה נוספת: עבור המחשבנית `st` שלפניך:

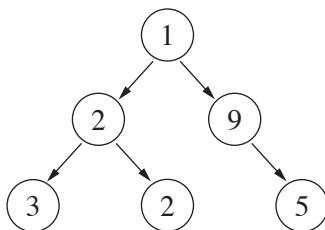
1223	← ראש המחשבנית
245	
521	
12334	

הפעולה תחזיר `false` כי מתוך הספרות המשמעותיות שבמחשבנית — 5, 2, 1 — הספרה 2 אינה מופיעה בספרת האחדות בשום מספר מן המספרים שבמחשבנית.

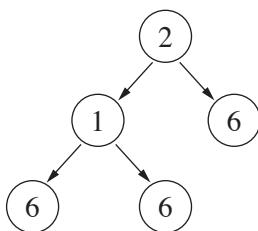
הערה: אפשר להשתמש בעולה שכתבת בסעיף א.

ונגיד: "עַצְמָסְפִּים" הוא עַצְמָה בִּינָרִי מִטְיפּוֹס שֶׁלִם, שֶׁלְ צוֹמָת בּוּ מִכְלֵסֶפֶרֶת בֵּין 1 לְ- 9 (כָּלְלָה), וְכָל מַסְלָול בְּעַצְמָה מִן  
הַשׂוֹרֶשׁ לְעַלְהָ מִיצְגָּמָה מִסְפֵּר: הַעֲלָה מִיצְגָּמָה אֶת סְפִּרְתַּת הַאַחֲדָות, הַרְמָה שְׁמַעַלְיוֹ אֶת סְפִּרְתַּת הַעֲשָׂרוֹת וְכֵן הַלְאָה עַד הַשׂוֹרֶשׁ  
שֶׁל הַעַזְמָה.

דוגמָה: בעַצְמָסְפִּים שֶׁלְפָנֵיךְ מִזְכָּרִים הַמִּסְפֵּרִים: 123 , 122 , 195 (בַּמַּסְלָולִים בְּעַצְמָה מִשְׁמָאל לִימִין).



דוגמָה נוֹסְפָּת: בעַצְמָסְפִּים שֶׁלְפָנֵיךְ מִזְכָּרִים הַמִּסְפֵּרִים: 216 , 216 , 26 (בַּמַּסְלָולִים בְּעַצְמָה מִשְׁמָאל לִימִין).



כתב פועלה חיצונית printAll בשפת C# או PrintAll Java בשפת Java. הפעולה מקבלת עַצְמָסְפִּים tree מִטְיפּוֹס שֶׁלִם  
ו吐דפס את כל המספרים שהמסלולים בעַצְמָסְפִּים.

אם tree הוא null הפעולה לא吐דפס דבר.

הערה: אין חשיבות לסדר שבו המספרים מודפסים.

## פרק שני (40 נקודות)

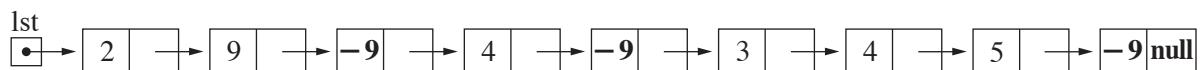
**שים לב:** בכל שאלה שנדרש בה מימוש, תוכל להשתמש בפעולות של המחלקות: תור, מחסנית, עץ ביניי וחוליה, בלי למשוך אותן. אם תשתמש בפעולות נוספות, יהיה עליך למשוך אותן.

ענה על שתיים מן השאלות 4-6. (לכל שאלה – 20 נקודות)

- 4. "שרשרת מספרים שלמים חיוביים" היא שרשרת חוליות שכל חוליה בה מכילה מספר שלם הגדל מ-0.
- "שרשרת ספרות" היא שרשרת חוליות שכל חוליה בה מכילה ספרה בין 0 ל-9 (כולל) או את המספר (-).

כל רצף ספרות בשרשראת מייצג מספר: הספרה הראשונה מייצגת את היחידות, הספרה השנייה את העשרות וכן הלאה. לאחר כל רצף של ספרות מופיע המספר (9-), והוא מסמן סוף של מספר בשרשראת.

לפניך דוגמה ל"שרשרת ספרות" המייצגת את המספרים: 92, 4, 543.



כתב פולה חיצונית בשפת Java בשם buildDigit בשם C# BuildDigit, המקבלת הפניה lst שאינה null ל"שרשרת מספרים שלמים חיוביים". הפולה תחזיר "שרשרת ספרות" המייצגת את המספרים שב"שרשרת מספרים שלמים חיוביים" לפי הסדר.

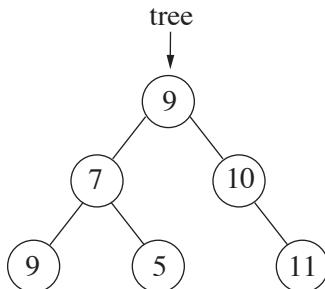
- .5. נגידר "בלוק" במחסנית כרצף של לפחות שני איברים זהים.
- כתב בשפת Java או בשפת C# פוללה חיצונית המקבלת מחסנית stk — מטיפוס שלם, ומחזירה מחסנית חדשה.
- המחסנית המוחזרת תכיל את כל האיברים מהמחסנית stk שאינם ב"בלוק".
- אין חשיבות לסדר האיברים במחסנית המוחזרת.
- הערות: — אם מספר מסוים מופיע לא ב"בלוק", הוא יהיה במחסנית המוחזרת גם אם נוסף על כך אותו המספר מופיע ב"בלוק".
- אם מספר מסוים מופיע כמה פעמים, לא ב"בלוק", הוא יופיע אותה כמות פעמים במחסנית המוחזרת.
- אם המחבנית stk ריקה, תוחזר מחסנית ריקה.

דוגמה:

מחסנית המוחזרת	מחסנית stk
15	14 ← ראש המחבנית
3	15
5	5
15	5
14	0
	0
	0
	5
	3
	-4
	-4
	15

- .6. עץ בינרי מטיפוס שלם של מספרים שאינם שליליים הוא "**עץ שאריות שוויוני**" במקרה זה:  
 כמות האיברים שמספריהם מתחלקים ב- 3 עם שארית 1 שווה לכמות האיברים שמספריהם מתחלקים ב- 3 עם שארית 2, ושווה לכמות האיברים שמספריהם מתחלקים ב- 3 ללא שארית.

דוגמה של "**עץ שאריות שוויוני**":



עץ בינרי זה הוא "**עץ שאריות שוויוני**" משום שיש בו שני מספרים שמתחלקים ב- 3 ללא שארית (9,9), שני מספרים שמתחלקים ב- 3 עם שארית 1 (7,10) ושני מספרים שמתחלקים ב- 3 עם שארית 2 (11,5).

כתב פועלה חיצונית בוליאנית בשפת C# או בשפת Java בשם TreeEqual בשם TreeEqual המתקבל עץ בינרי מטיפוס שלם, לא ריק, של מספרים שאינם שליליים ובודקת אם הוא "**עץ שאריות שוויוני**".  
 אם כן — תחזיר הפעלה true, אחרת היא תחזיר false.

## פרק שני

**שים לב:** בכל שאלה שנדרש בה מימוש אתה יכול להשתמש בפעולות של המחלקות: תור, מחסנית, עץ ביניי וחוליה, בלי למשמש אותן. אם אתה משתמש בפעולות נוספות, عليك למשמש אותן.

.4. שאלה זו תוכל להשתמש בפעולה החיצונית שלפניך בלי למשמש אותה.

cotoraת הפעולה	תיאור הפעולה
<b>Java</b> <pre>public static Node&lt;Integer&gt; delete (int num, Node&lt;Integer&gt; lst)</pre> <b>C#</b> <pre>public static Node&lt;int&gt; Delete (int num, Node&lt;int&gt; lst)</pre>	<p>הפעולה מקבלת מספר – num ופניה לתחילת שרשרת חוליות .lst – .lst – הפניה מוחקמת את החליות שבן הערך num ומזהירה הפניה לתחילת שרשרת החליות.</p>

נתונה המחלקה **BiList** – דו-שרשרת, ולה שתי תכונות:

- lst1 – הפניה לתחילת שרשרת חוליות מטיפוס שלם
- lst2 – הפניה לתחילת שרשרת חוליות מטיפוס שלם

לפניך ממשך חלקו של המחלקה **BiList** בשפות **Java** ו- **C#**. יש להשתמש בפעולות המשך ללא צורך למשמש אותן.

cotoraת הפעולה	תיאור הפעולה
<pre>public BiList ()</pre>	פעולה הבונה את העצם עם הפניות לשתי שרשרות ריקות.
<b>Java</b> <pre>public void addNum (int num, int codeList)</pre> <b>C#</b> <pre>public void AddNum (int num, int codeList)</pre>	<p>פעולה המוסיפה חוליה שבה הערך num לסוף השרשרת lst1 או לסוף השרשרת lst2 בהתאם ל- : codeList .lst1 , num יוכנס lst1 , codeList = 1 כאשר 2 , num יוכנס lst2 .lst2 , codeList = 2 והנה שהערך של הפרמטר codeList תקין.</p>

(שים לב: המשך השאלה בעמוד הבא.)

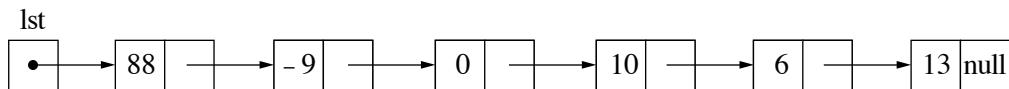
כתב פוליה חיצונית ששם generateBilist Java בשפת C# מקבלת שרשרת חוליות — `lst` של מספרים שלמים. מספר החוליות ב-`lst` זוגי והמספרים בחוליות שלה שוניים זה מזה. הפעולה תחזיר עצם מטיפוס **BiList** שמתקיים בו התנאים הבאים:

- כל אחד מן המספרים בשרשראת `lst` יופיע באחת מן השרשרות `lst1` ו-`lst2`.
- כל המספרים בשרשראת `lst1` יהיו גדולים מכל המספרים בשרשראת `lst2`.
- מספר החוליות בשתי השרשרות `lst1` ו-`lst2` יהיה זהה.

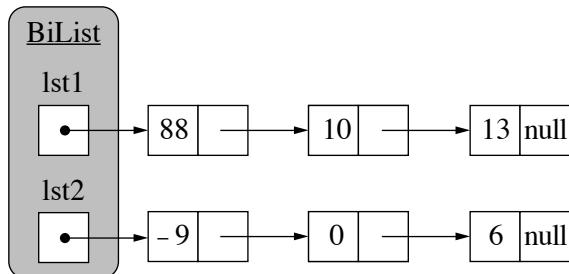
שים לב: אין להוסיף פעולות, גם לא פעולות `get` ו-`set` בשפת Java או `Set` בשפת C#, למחלקה **BiList**.

דוגמיה:

בעבור השרשרת `lst` ש לפניו:



הפעולה תחזיר את העצם הזה:



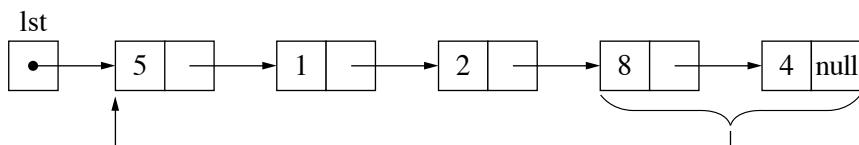
הערות:

- אין צורך לשמור על השרשרת `lst`.
- אין חשיבות לסדר האיברים בשרשראת `lst1` ובשרשת `lst2`.

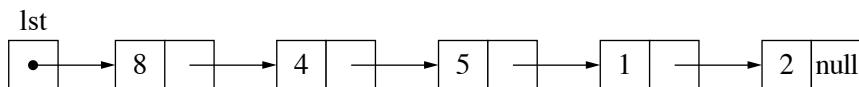
5. בשרשראת חוליות, "העברה מעגלית של n חוליות" היא העברת n החוליות האחרונות לתחילת השרשראת (בלי לשנות את סדר הופעתן).

בדוגמה שלפניך בעבור 2 = n : מעבירים את שתי החוליות האחרונות לתחילת השרשראת.

#### שרשרת החוליות לפני העברה



#### שרשרת החוליות לאחר העברה



- a. כתוב פעולה חיצונית בשם move Java או Move C# המתקבלת שרשרת חוליות — lst מתויפוס שלם ומספר n מטיפוס שלם. הפעולה תחזיר את שרשרת החוליות לאחר "העברה מעגלית של n חוליות".  
הנמ:  $n \geq 0$  ומספר החוליות בשרשראת גדול מ- n .
- b. מהי סיבוכיות זמן הריצה של הפעולה שכותבת בסעיף א? נמק.

C#

א. לפני הפעולה Sod1 .

```
public static bool Sod1 (int[] arr, int x, int i)
{
    if ( i == -1) return false;
    if (arr[i] == x) return true;
    return Sod1(arr, x, i - 1);
}
```

(1) כתוב את הערך המוחזר בעבר זימון הפעולה Sod1(a, 8, a.Length - 1) והערך a שלפניך.  
עליך להראות מעקב.

0	1	2	3	4
a	5	4	15	12

(2) בעבר מערך כלשהו a ומספר x , מהי מטרת הפעולה ? Sod1(a, x, a.Length - 1)  
(3) מהי סיבוכיות זמן הריצה של הפעולה Sod1 ? נמק.

ב. לפני הפעולה Sod2 .

```
public static bool Sod2 (int[] arr, int x, int i)
{
    if ( i == 0) return false;
    if (Sod1(arr, x - arr[i], i - 1)) return true;
    return Sod2(arr, x, i - 1);
}
```

(1) כתוב את הערך המוחזר בעבר זימון הפעולה Sod2(a, 16, a.Length - 1) והערך a שלפניך.  
עליך להראות מעקב.

0	1	2	3	4
a	5	4	15	12

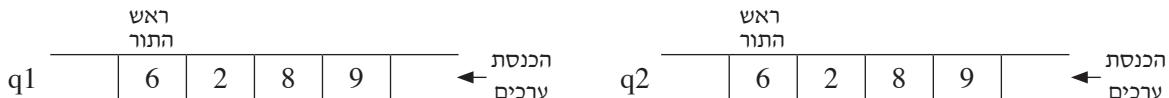
בסעיף זה, אין צורך לבצע מעקב אחר הפעולה Sod1 .  
(2) בעבר מערך כלשהו a ומספר x , מהי מטרת הפעולה ? Sod2(a, x, a.Length - 1)  
(3) מהי סיבוכיות זמן הריצה של הפעולה Sod2 ? נמק.

.7. בשאלת זו תוכל להשתמש בפעולה החיצונית שלפניך בלי למש אותה.

כותרת הפעולה	תיאור הפעולה
public static int size (Queue<Integer> q) – Java	הפעולהמחזירה את מספר האיברים בתור q
public static int Size (Queue<int> q) – C#	בשפת C# – ביל לשנות את התור.

- א. שני תורים,  $q_1$  ו-  $q_2$ , יהיו "תורים זהים" אם מספר האיברים בשני התורים זהה, ובשני התורים מופיעים בד同 אותם ערכים ובאותו הסדר.

דוגמיה לשני תורים זהים:



כתב פעלת חיצונית ששם IsIdentical בשת Java או IsIdentical # בשת C# מקבלת שני תורים מטיפוסם שלם  $q_1$  ו-  $q_2$ , ומחזירה true אם התורים זהים, אחרת היא מוחזירה false.

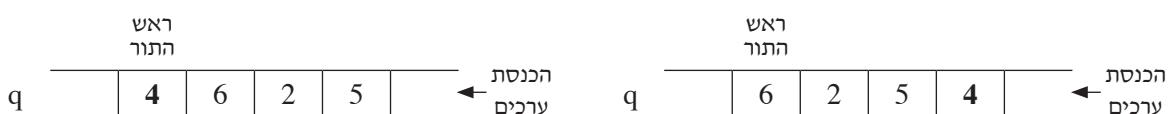
הערה: עם סיום הפעולה, חובנה לשמור על מבנה התורים המקורי שהתקבל.

- ב. העברת מההתחלת לסוף" היא העברת מספר מראש התור לסופו.

דוגמיה:

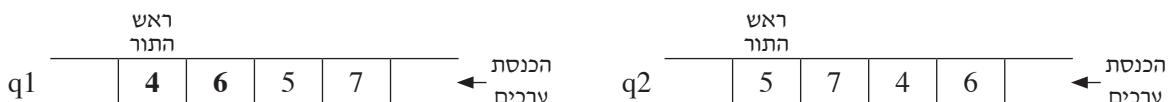
התור המקורי

"התור אחרי העברת מההתחלת לסוף"



כתב פעלת חיצונית ששם IsSimilar בשת Java או IsSimilar # בשת C# מקבלת שני תורים מטיפוסם שלם,  $q_1$  ו-  $q_2$ . הפעולה מוחזירה true אם התורים  $q_1$  ו-  $q_2$  זהים – בין שהם זהים כמו שהם ובין שם יהיו זהים לאחר שנבצע ב-  $q_1$  "העברה מההתחלת לסוף", פעם אחת או יותר. אחרת הפעולה מוחזירה false.

דוגמיה: הפעולה תחזיר true בעבור שני התורים  $q_1$  ו-  $q_2$  שלפניך:



זאת מכיוון שלאחר שבתור  $q_1$  נבצע פעמיים "העברה מההתחלת לסוף", הוא יהיה זהה לתור  $q_2$ , וייראה כך:



הערות: – חובנה להשתמש בפעולה שכותבת בסעיף א.

– אין צורך לשמור על מבנה התורים המקורי שהתקבל.

## פרק שני

**שים לב:** בכל שאלה שנדרש בה שימוש, תוכל להשתמש בפעולות של המחלקות: תור, מחסנית, עץ ביןרי וחוליה, בלי למשוך אותן. אם תשתמש בפעולות נוספות, יהיה عليك למשוך אותן.

.4. "שרשרת מאורגנת" היא שרשרת חוליות מתיפוס שלם שמתקיימים בה התנאים הבאים:

- מספר החוליות בשרשראת הוא זוגי.
- כל המספרים בחצי הראשון של שרשרת קטנים מכל המספרים בחצי השני של השרשרת.

דוגמה ל"שרשרת מאורגנת":



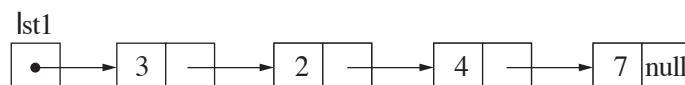
- a. כתוב פעולה חיצונית ששם `isArranged` בשפת C# או `IsArranged` בשפת Java או `isArranged` בפונקציה `IsArranged` המתקבלת שרשרת חוליות –  
 .lst מטיפוס שלם, שאינה `null`, ומחזירה `true` אם היא "שרשרת מאורגנת", אחרת היא מחזירה `false`.  
הערה: חובה לשמר על השרשרת `lst`.
- b. מהי סיבוכיות זמן הריצה של פעולה זו? נמק.

- .5. א. שרשרת החוליות  $lst_1$  היא "תת-שרשרת תחילה" של שרשרת החוליות  $lst_2$  אם כל הערכים של  $lst_1$  מופיעים באותו רצף מתוך  $lst_2$  (ייתכנו ערכים נוספים ב-  $lst_2$  אחרי רצף זה).

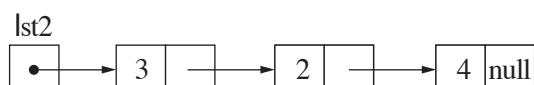
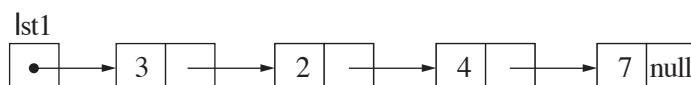
דוגמה ל-  $lst_1$  שהיא "תת-שרשרת תחילה" של  $lst_2$  :



דוגמה ל-  $lst_1$  ש אינה "תת-שרשרת תחילה" של  $lst_2$  :



דוגמה נוספת נוסףת ל-  $lst_1$  ש אינה "תת-שרשרת תחילה" של  $lst_2$  :



כתב פוליה חיצונית שנקרא `isPrefix` בשפת C# או `isPrefix` בשפת Java או מקבלת שתי שרשרות חוליות  $lst_1$  ו-  $lst_2$  שאינן `null`.

הפעולה תחזיר `true` אם  $lst_1$  היא "תת-שרשרת תחילה" של  $lst_2$ , אחרת היא תחזיר `false`.  
הערה: חובה לשמור על השרשראות  $lst_1$  ו-  $lst_2$ .

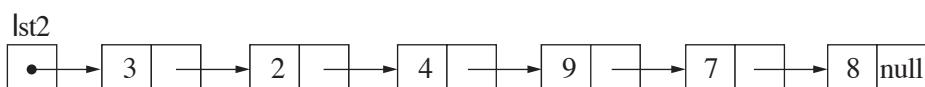
(שים לב: המשך השאלה בעמוד הבא.)

**ב.** שרשרת החוליות  $\text{lst1}$  היא "תת-שרשרת" של שרשרת החוליות  $\text{lst2}$  אם כל הערכים של  $\text{lst1}$  מופיעים במקום כלשהו ב- $\text{lst2}$  באותו הרצף.

דוגמיה ל- $\text{lst1}$  שהיא "תת-שרשרת" של  $\text{lst2}$ :



דוגמיה ל- $\text{lst1}$  שאייה "תת-שרשרת" של  $\text{lst2}$ :



כתב פוליה חיצונית ששםה `isSubChain` בשפת C# או `Java` או `isSubChain` מקבלת שתי שרשרות חוליות מטיפוסם של  $\text{lst1}$  ו- $\text{lst2}$  שאינם `null`.

הפעולה תחזיר `true` אם  $\text{lst1}$  היא "תת-שרשרת" של  $\text{lst2}$ , אחרת היא תחזיר `false`.  
הערה: חובה להשתמש בפעולת שכבתה בסעיף א.

.**6.** נתונה המחלקה **CovidTest**, המייצגת אדם שנבדק בדיקת קורונה, ולה 4 תכונות:

- name — שם הנבדק מטיפוס מחווזת
  - id — מספר זהות מטיפוס מחווזת
  - cityCode — קוד של עיר המגורים, מטיפוס שלם (לדוגמה: 1030 בעבר אשדוד, 23 בעבר עכו)
  - sick — משתנה מטיפוס בוליאני, מקבל true אם הנבדק חולה, אחרת הוא מקבל false
- הנח שיש פעולות get ו-set בשפת Java ופעולות Get ו- Set בשפת C# בעבר תכונות המחלקה.

כתב פעולה חיצונית mostSick Java או MostSick Java בשפת C# המתקבלת תור — q שאינו ריק מטיפוס **CovidTest**. הפעולה תחזיר את הקוד של העיר שבה כמות החולים היא הגדולה ביותר.

- הערות:
- מיקום הנבדקים בתור אינו לפי סדר כלשהו.
  - כל נבדק מופיע רק פעם אחת בתור.
  - הקוד של העיר אינו קשור לגודל התור (לדוגמה: ייתכן שמספר האיברים בתור הוא 1,000 , וקיים קוד עיר שמספרו 5000 ).
  - אין צורך לשמר על התור.

הנח שיש רק עיר אתנית שבה כמות החולים היא הגדולה ביותר.

## פרק שני (50 נקודות)

**シומו לב:** בכל שאלה שנדרש בה מימוש אפשר להשתמש בפעולות של המחלקות: תור, מחסנית, עץ ביןרי וחוליה, בלי למשתמש אותן. אם משתמשים בפעולות נוספות, יש למשתמש אותן.

ענו על שתיים מהתוצאות 4–7 (לכל שאלה – 25 נקודות).

4. נתונה המחלקה **Range** – טווח, ולה שתי תכונות:

- – מספר מטיפוס שלם low
- – מספר מטיפוס שלם high

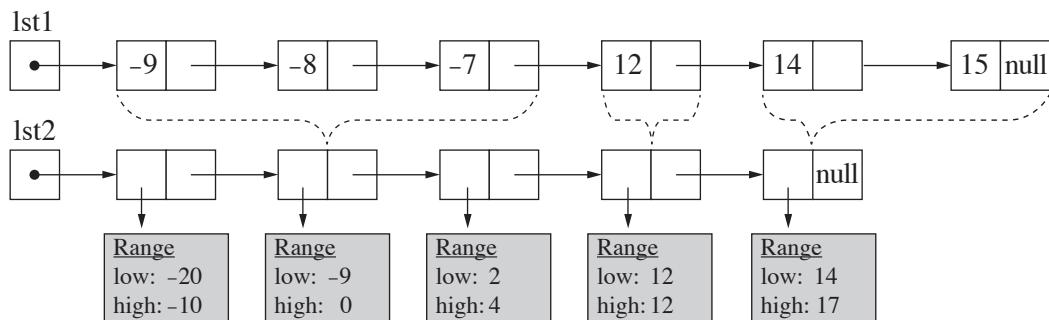
המספר high גדול או שווה ל- low ( $high \geq low$ )

הניחו שיש פעולות get/Get בuboar תכונות המחלקה.

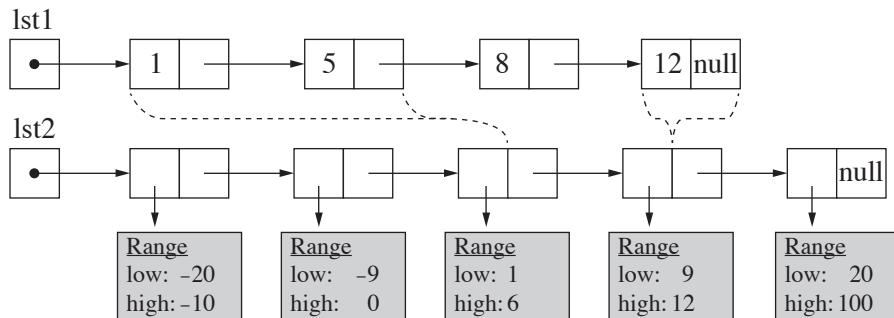
מספר כלשהו, x, ”מוכל” בעצם מטיפוס Range אם הוא נמצא בטוחה המספרים שבין low ובין high ( $x \geq low$ ).

שרשרת חוליות – lst1 מטיפוס שלם ”מוכלת” בשרשראת חוליות – lst2 מטיפוס Range אם בעבור כל מספר בשרשרת lst1 קיימת חוליה בשרשראת lst2 המכילה אותו.

דוגמה לשרשראת lst1 המוכלת בשרשראת lst2 :



דוגמה לשרשראת lst1 שאינה מוכלת בשרשראת lst2 :



הסביר: המספר 8 בשרשראת lst1 אינו ”מוכל” בשום חוליה בשרשראת lst2.

משמעות הפעולה החיצונית של הלן:

**Java** – public static boolean isIncluded (Node<Integer> lst1, Node<Range> lst2)

**C#** – public static bool IsIncluded (Node<int> lst1, Node<Range> lst2)

הפעולה ממחירה true אם "מוכלת" ב- lst2, אחרת היא ממחירה false. הפעולה חייבת לעבוד בסיבוכיות זמן ריצה של  $O(N)$ .

הערה: N הוא אורך שרשרת הארכיה יותר מבין שתי שרשרות.

הנחות:

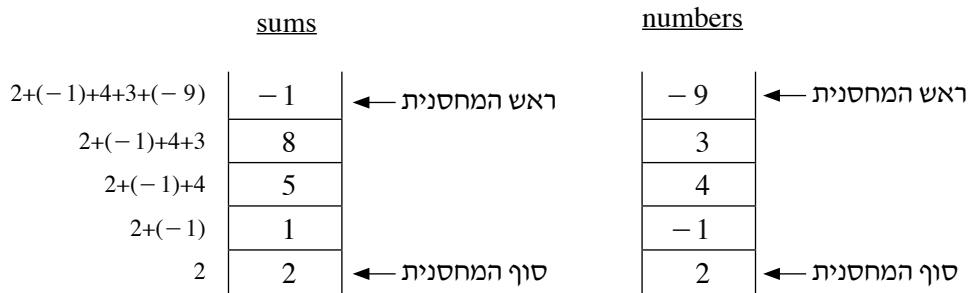
- lst1 ו- lst2 אינם null.
- בשרשראת lst2 כל העצמים מטיפוס Range אינם null.
- השרשרת lst1 ممונית בסדר עולה.
- השרשרת lst2 ممונית בסדר עולה, ככלומר, ערך ה- high של כל חוליה קטן מערך ה- low של החוליה הבאה אחריה בשרשראת (כפי שמצוין בדוגמאות בעמוד הקודם).

.5. נתונה המחלקה **TwoStack**, ולה שתי תכונות:

- numbers – מחסנית מטיפוס שלם
- sums – מחסנית מטיפוס שלם

היחס בין המחסנית sums למחסנית numbers הוא כמפורט להלן:  
 המספר בסוף המחסנית sums שווה למספר בסוף המחסנית numbers,  
 המספר השני מסוף המחסנית sums שווה לסכום שני המספרים האחרונים במחסנית numbers,  
 המספר השלישי מסוף המחסנית sums שווה לסכום שלושת המספרים האחרונים במחסנית numbers  
 וכן הלאה עד המספר בראש המחסנית sums, השווה לסכום כל המספרים במחסנית numbers.

דוגמיה:



a. משמעות הפעולה הפנימית של להלן:

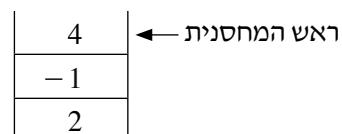
**Java** – public Stack<Integer> getNums (int x)

**C#** – public Stack<int> GetNums (int x)

הפעולה מקבלת מספר x השווה לאחד המספרים במחסנית sums, ומוחזירה מחסנית חדשה מטיפוס שלם, שבה מופיעים המספרים מן המחסנית numbers שסכום שווה למספר x.  
 הניחו שהמספר x קיים במחסנית sums ומופיע בה רק פעם אחת.

דוגמיה:

בעבור המחסניות שבדוגמה לעיל ו-  $x = 5$  (5 הוא המספר השלישי מלמטה במחסנית sums), הפעולה תחזיר את המחסנית של להלן:



הערות: – אפשר לשנות את המחסניות של המחלקה.

– אין חשיבות לסדר המספרים במחסנית המוחזרת.

. StackInt Stack<int>/Stack<Integer> Skola להגדירה –  
 אפשר להשתמש בכל אחת מהן, לבחירתכם.

**ב.** ממשו את הפעולה הפנימית של להלן:

**Java** – public void eraseNum (int x)

**C#** – public void EraseNum (int x)

הפעולה מוחקת את המספר x מן המחסנית numbers ומקנתת את המחסנית sums בהתאם.  
הניחו שהמספר x קיים במחסנית numbers והוא מופיע בה רק פעם אחת.  
הערה: יש לשמור על סדר המספרים שנשארו במחסנית numbers.

דוגמא:

בעבור המחסניות שלפניכם ו-  $x = 3$  :

<u>sums</u>	<u>numbers</u>
8	6
2	-3
5	3
2	2

לאחר הרצת הפעולה ייראו המחסניות כך:

<u>sums</u>	<u>numbers</u>
5	6
-1	-3
2	2

בשאלה זו אפשר להשתמש בפעולה החיצונית eraseFirst / EraseFirst שלהן בלי למשם אותה.

דוגמאות	תיאור הפעולה	כותרת הפעולה
, str = "hello"	- בעבור המחרוזת "hello" הפעולה תחזיר את המחרוזת "ello".	הפעולה מחזירה תת-מחרוזת של str, ללא התו הראשון. : <b>Java</b> public static String eraseFirst (String str)
, str = "temp"	- בעבור המחרוזת "temp" הפעולה תחזיר את המחרוזת "emp".	אם המחרוזת – str ריקה לפנוי זימון הפעולה, תהיה שגיאת. : <b>C#</b> public static string EraseFirst (string str)
, str = "m"	- בעבור המחרוזת "m" הפעולה תחזיר מחרוזת ריקה "".	
, str = ""	- בעבור המחרוזת הריקה "" תהיה שגיאת.	

שימוש את הפעולה החיצונית שלהן:

**Java** – public static boolean wordFromRoot (BinNode<Character> tree, String str)

**C#** – public static bool WordFromRoot (BinNode<char> tree, string str)

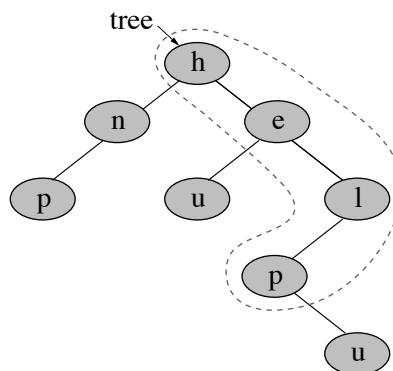
הפעולה מקבלת מחרוזת – str המכילה לפחות תו אחד, והפניה לעץ בינירי של תווים – tree שאינו null .

הפעולה תחזיר true אם קיים מסלול המתחילה בשורש העץ שבו בוצף התווים זהה למחרוזת – str . אחרת הפעולה תחזיר false .

הערה: אותן קטנה ואות גדולה אינן זהות זו לזו.

דוגמה:

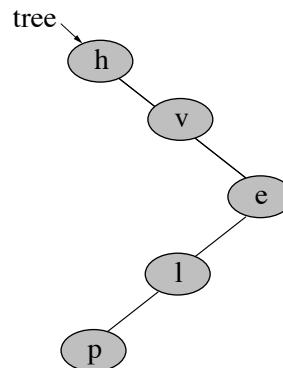
בעבור העץ הנתון והמחרוזת "help" הפעולה תחזיר true .



(שים לב: המשך השאלה בעמוד הבא.)

דוגמה:

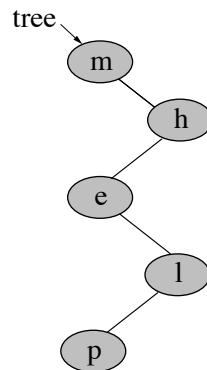
בעבור העץ הנתון והמחרוזת "help" הפעולה תחזיר false.



הסבר: לא קיים בעץ רצף תווים זהה למחרוזת "help".

דוגמה:

בעבור העץ הנתון והמחרוזת "help" הפעולה תחזיר false.



הסבר: אף על פי שקיימים בעץ רצף תווים זהה למחרוזת "help", הפעולה תחזיר false, כי הרצף אינו מתחילה בשורש העץ.

**שים לב:** אם אתה כותב ב-C#, ענה על שתיים מהשאלות 21-24.

. 21. נתונות ארבע המחלקות: Test , MultiOne , SingleOne , Basis .

עקוב בעזרת טבלת מעקב אחר הפעולה Main במחלקה Main , Test , ורשות את הפלט.

על הטעלה לכלול את ערכי המשתנים, ובמעבר כל עצם – את ערכי התוכנות שלו.

```
public class Basis
{
    protected int num1;
    public Basis ()
    {
    }
    public Basis (int n)
    {
        this. num1 = n;
    }
    public virtual void Print ()
    {
        Console.WriteLine (this.num1);
    }
}

public class SingleOne : Basis
{
    protected int num2;
    public SingleOne (int n1 , int n2) : base (n1)
    {
        this.num2 = n2;
    }
    public override void Print ()
    {
        base.Print ();
        Console.WriteLine (this.num2);
    }
}
```

(**שים לב:** המשך השאלה בעמוד הבא.)

/המשך בעמוד הבא/

+

+

```

public class MultiOne : Basis
{
    private int count = 0;
    private Basis[] arr;
    public MultiOne ()
    {
        this.arr = new Basis[5] ;
    }
    public override void Print ()
    {
        for (int i = 0 ; i < count ; i++)
            arr[i].Print ();
    }

    public void Add (Basis b)
    {
        arr[count] = b;
        count++;
    }
}

public class Test
{
    public static void Main (string [] args)
    {
        MultiOne container = new MultiOne ();
        SingleOne s1 = new SingleOne (11 , 35);
        container.Add (s1);
        s1 = new SingleOne (47 , 22);
        container.Add (s1);
        s1 = new SingleOne (8 , 17);
        container.Add (s1);
        MultiOne subContainer = new MultiOne ();
        s1 = new SingleOne (53 , 40);
        subContainer.Add (s1);
        s1 = new SingleOne (21 , 13);
        subContainer.Add (s1);
        s1 = new SingleOne (39 , 62);
        subContainer.Add (s1);
        container.Add (subContainer);
        container.Print ();
    }
}

```

+

+

+

+

.22. לפניך פרויקט ובו הממשקים IFirst , ISecond , IThird , AAA , BBB , ומחלקות Test ו CCC .

public interface IFirst

```
{
    public bool OpA (Object stam);
    public void OpB (int num);
}
```

public interface ISecond: IFirst

```
{
    public int OpC ();
}
```

public interface IThird

```
{
    public int OpD ();
}
```

public class AAA : IFirst

```
{
}
```

public class BBB : ISecond

```
{
}
```

public class CCC : ISecond , IThird

```
{
}
```

public class Test

```
{
    public static void Main (string[] args)
    {
    }
}
```

(שים לב: סעיפים השאלה בעמוד הבא).

+

+

א. רשום אילו פעולות יש למשב בכל אחת מן המחלקות CCC, BBB, AAA וסביר מודיע.

ב. לכל אחת מקבוצות ההוראות -iv שלפניך, קבע אם היא חוקית או אינה חוקית, ונמק את קביעתך.

ISecond s1 = new ISecond ();      i

BBB b1 = new BBB ();      ii

AAA a1 = new AAA ();      iii

IFirst f1 = al;

CCC c1 = new CCC ();      iv

IFirst f2 = cl;

ג. לפניך הדרישות - ii :

i הפעלת הפעולה OpB על עצם מן הטיפוס BBB.

ii המרת עצם מן הטיפוס CCC להיות עצם מן הטיפוס AAA.

לכל אחת מן הדרישות, קבע אם אפשר לבצע אותה על ידי כתיבת הוראה בפעולת Main.

אם אפשר לבצע אותה — כתוב את ההוראה המתאימה,

אם אי-אפשר לבצע אותה — נמק מודיע.

+

+

.23. המחלקה Point מדגישה את טיפוס הנתונים נקודה במשור. נקודה מתוארת במערכת

ציריים על ידי שתי הקואורדינטות שלה (x , y). x ו- y הם מספרים שלמים.

למחלקה Point התכונות האלה:

int x \*

int y \*

לפניך חלק ממשק המחלקה Point:

Point (int x , int y)	פעולה בונה המחזירה נקודה שהקואורדינטות שלה הן (x , y)
int GetX ()	פעולה המחזירה את ערך x של הנקודה
int GetY ()	פעולה המחזירה את ערך y של הנקודה

.המחלקה Line מדגישה את טיפוס הנתונים קטע במשור. קטע במשור מוגדר על ידי

שתי הנקודות הנמצאות בקצוות שלו.

למחלקה Line התכונות האלה:

Point point1 \*

Point point2 \*

לפניך חלק ממשק המחלקה Line:

Line (Point p1 , Point p2)	פעולה בונה המחזירה קטע הנמצא בין הנקודות p1 ו- p2
Point GetPoint1 ()	פעולה המחזירה את הנקודה point1 של הקטע
Point GetPoint2 ()	פעולה המחזירה את הנקודה point2 של הקטע

.המחלקה Drawing – סרטוט, היא אוסף של נקודות במשור ואוסף של קטעים במשור.

למחלקה Drawing התכונות האלה:

\* מערך של נקודות.

\* מערך של קטעים.

\* מספר הנקודות שיש בסרטוט.

\* מספר הקטעים שיש בסרטוט.

(שים לב: סעיפים השאלה בעמוד הבא.)

/המשך בעמוד 29/

מՔש ב-#C את הסעיפים א-ו במחלקה Drawing. השתמש במשקי המחלקות Point ו- Line, בלי למשוך אותן. אפשר להוסיף פעולות למשקים אלה. אם אתה מוסיף פעולות, عليك למשוך אותן.

**א.** כוורת המחלקה והתכונות שלה.

**ב.** פעולה בונה, מקבלת את מספר הנקודות шибולות להיות בסדרות – קה ואות מספר הקטיעים шибולים להיות בסדרות – אן.

הפעולה תבצע:

אתחול מערך הנקודות להיות בגודל קה,

אתחול מערך הקטיעים להיות בגודל אן,

אתחול מספר הנקודות שים בסדרות ל-0,

אתחול מספר הקטיעים שים בסדרות ל-0.

**ג.** פעולה המאזרת את נקודה הנמצאת במקום מ' בערך הנקודות של הסדרות. הנה שיש נקודה במקום מ' בערך.

**ד.** פעולה המוסיפה נקודה לסדרות.

הנה שבערך הנקודות יש מקום לנקודה.

**ה.** פעולה מקבלת נקודה בסדרות, ומחזירה את מספר הקטיעים שהנקודה נמצאת באחד מקצתיהם.

**ו.** פעולה מקבלת נקודה בסדרות, ומחזירה "אמת" – אם אין שם קטע שנקודה זו היא אחד מקצתיהם, ו"שקר" – אחרת.

.24 איגוד הבודרסל החליט למחשב את רישום הבודרסלנים במדינה. לצורך כך הוא הגדר

שלוש מחלקות: בודרסלן (Player), קבוצה (Team) ואיגוד הבודרסל (Union).

למחלקה Player התכונות האלה:

\* שם בודרסלן — string playerName —

\* גובה במטרים של בודרסלן — double height —

\* מספר נקודות שצבר בעונה זו — int points —

במחלקה Team הוגדרו הפעולות המאוחרות:

GetPlayerName () , GetHeight () , GetPoints ()

והפעולות הקובעות:

SetName (string name) , SetHeight (double height) , SetPoints (int points)

למחלקה Team התכונות האלה:

\* שם קבוצה — string teamName —

\* מספר השחקנים בקבוצה — int num —

\* מערך בגודל 20 של שחקני הקבוצה — Player[] playersArray —

במחלקה Team הוגדרו הפעולות המאוחרות:

GetTeamName () , GetNum () , GetPlayersArray ()

והפעולה הקובעת:

SetName (string name)

. א. מימוש ב-C# את המחלקה Union, המשתמשת במחלקות Player ו-Team.

באיגוד הבודרסל יש 30 קבוצות.

למחלקה Union התכונה: מערך של קבוצות בודרסל.

מיימוש המחלקה יכולול את התת-סעיפים i-ii שלפניך:

i כוורת המחלקה

ii הגדרת התכונה של המחלקה

(שים לב: המשך סעיף השאלה בעמוד הבא).

/המשך בעמוד 31/

+

+

**ב.** ממש ב-#C במחלקה Union את הפעולה שלפניך.

void PrintTeamsDetails ()	הפעולה מדפסה, בעבר כל קבוצה באיגוד הcadorsel, את שם הקבוצה ואת מספר השחקנים שבה.
---------------------------	--

**ג.** איגוד הcadorsel מעוניין לקבל נתונים על מספר הנקודות שצבר כל אחד מהשחקנים בעונה שחלפה, כדי לדעת כמה קלעים חלשים, כמה קלעים ממוצעים, וכמה קלעים מצטינים יש באיגוד.

קלע חלש הוא שחקן שצבר בעונה פחות מ-100 נקודות.

קלע ממוצע הוא שחקן שצבר בעונה בין 100 ל-200 נקודות (כולל).

קלע מצטיין הוא שחקן שצבר בעונה יותר מ-200 נקודות.

רשום את הפעולות שיש להוסיף לכל אחת משלוש המחלקות Team , Player ו- Union , כדי לקבל את המידע הנדרש. בכל אחת מהמחלקות עליך להוסיף לפחות פעולה אחת.

בעבור כל פעולה שהוספה, רשום את החתימה שלה ב-#C , את התיעוד שלה, ובאייזו מחלוקת יש למשם אותה.

הערה: אין צורך למשם את הפעולות.

## ב ה צ ל ח ה !

זכות היוצרים שומרה למדינת ישראל  
אין להעתיק או לפרסם אלא ברשות משרד החינוך התרבות והספורט

+

+

### תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב ב-#C, ענה על שתיים מהשאלות **21-24** (לכל שאלה – 25 נקודות).

**21.** באתר "הסקר השבועי" מתפרסם בכל שבוע סקר חדש ובו שאלה אחת. האתר שומר את 50 הסקרים האחרונים, כולל הסקר החדש. הנתונים בעבר כל סקר הם: תאריך פרסום שלו, השאלה הנשאלת בו, ארבע תשובות אפשריות לשאלת, ובמעבר כל תשובה – מספר המשתתפים שבחרו בה. משתנה יכול להשיב רק על השאלה בסקר האחרון שפורסם, ורשיי לבחור רק בתשובה אחת.

**a.** הגדר את המחלקות הנדרשות לצורכי מימוש האתר "הסקר בשבועי".

בעבור כל אחת מהמחלקות שאתה מגדיר:

- i כתוב מה היא מייצגת.
- ii כתוב את הכותרת שלה ב-#C.
- iii הגדר את התכונות שלה ב-#C, וכותב תיעוד לכל תכונה.

**b.** ניהול האתר "הסקר בשבועי" כולל:

- הוספת סקר חדש.
- הדפסת השאלה של הסקר החדש, והדפסת ארבע התשובות האפשרות עליה.
- קליטת תשובה שבחרו משתתף בסקר החדש, ועדכוון מספר המשתתפים שבחרו בתשובה זו.
- קליטת תאריך של פרסום סקר, ובמעבר שאלת הסקר שפורסמה בתאריך זה, הדפסת הנתונים האלה:
  - מספר המשתתפים שבחרו בכל אחת מהתשובות.
  - מה הייתה התשובה שנבחרה על ידי המספר הגדל ביותר של משתתפים.

הגדר את הפעולות הנדרשות לניהול האתר "הסקר בשבועי". הנח שפעולות get ו-set מוגדרות בעבר כל תכונה בכל אחת מהמחלקות שהגדרת בסעיף א.

בעבור כל אחת מהפעולות כתוב:

- i באיזו מחלוקת יש להגדיר אותה.
- ii את הכותרת שלה ב-#C.
- iii תיעוד שלה.

ג. ממש ב-C# פועלה המחזירה את מספר הסקרים שבם השתתפו יותר מ-1000 משתתפים. ציין באיזו מחלוקת מהמחלקות שהגדרת בסעיף א יש ממש פועלה זו. הנה שפועלות get ו- set מוגדרות בעבר כל תcona בכל אחת מהמחלקות שהגדרת בסעיף א.

.22. במוזאון "ישן וחדש" פותח פרויקט לרשום של המוצגים במוזאון.

לפניך הגדרה מחלקה של המחלקות ב-C#:

מוצר – צילום – פסל – ציור , Statue – Painting – Exhibit – Sculpture

public class **Exhibit**

{

protected string creator;	//	שם היוצר
protected string name;	//	שם המוצג
protected int year;	//	שנת היצירה
protected double height;	//	גובה המוצג
protected double width;	//	רוחב המוצג

    public Exhibit()

    {

        //...

    }

    public Exhibit(string creator, string name, int year, double height, double width)

    {

        //...

    }

    /\* הפעולה מחזירה את שטח התצוגה הדרוש להציג המוצג בתערוכה \*/

    public virtual double ExhibitionSpace()

    {

        return this.height \* this.width;

    }

}

public class **Painting : Exhibit**

{

private string style;	//	סגנון הציור
-----------------------	----	-------------

}

public class **Statue : Exhibit**

{

private string[] materials;	//	החומרים מהם בנוי הפסל
private string technique;	//	טכניקת הפיסול
private double depth;	//	עומק הפסל

    public override double ExhibitionSpace()

    {

        return this.height \* this.width \* this.depth;

    }

}

/המשך בעמוד הבא

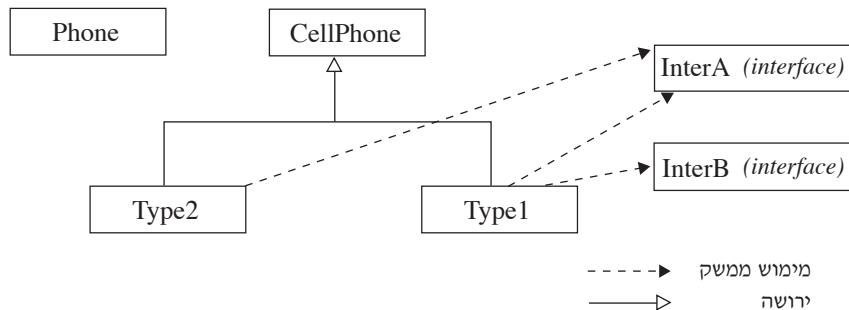
(שים לב: המשך השאלה בעמוד הבא.)

```
public class Photograph : Exhibit
```

```
{
    private string kind;           //    סוג הצילום
}
```

- א.** ממש ב-C# פועלה בונה במחלקה Painting, שתשתמש בפועלה הבונה המקבלת פרמטרים במחלקה Exhibit.
- ב.** ציין איזה מנגנון של תכונות מונחה עצמים באידי ביטוי בהגדרת הפעולה ExhibitionSpace() והפעולה Exhibit שבסמךיה Statue שבסמךיה ExhibitionSpace().
- ג.** האם אפשר למש את הפעולה Statue במחלקה Statue בעזרת הפעולה ExhibitionSpace() שבמחלקה Exhibit ? אם כן – משמש אותה; אם לא – נמק מדוע.
- ד.** המחלקה חדר – Room, מייצגת חדר במוזיאון. בחדר יכולים להיות עד 25 מוצגים. בחדר יכולים להיות ציורים, פסלים, צילומים.
- ממש ב-C# את המחלקה Room. מימוש המחלקה צריך לכלול את הסעיפים i-iii שלפניך:
- i כוורת המחלקה, והגדרת התכונות שלה.
  - ii פועלה המחזיר את מספר ציורים שנמצאים בחדר.
  - iii פועלה מקבלת מוצג ומוסיפה אותו לחדר. הנה שיש למוצג זה מקום בחדר. המחלקה תערוכה – Exhibition, מייצגת תערוכת אמנויות המתקיימת במוזיאון ב-10 חדרים הממוספרים מ-1 עד 10.
- ממש ב-C# את המחלקה Exhibition. מימוש המחלקה צריך לכלול את הסעיפים i-iii שלפניך:
- i כוורת המחלקה, והגדרת התכונות שלה.
  - ii פועלה מקבלת מספר חדר בין 1 ל-10, ומציירה את מספר ציורים שנמצאים בחדר זה.
  - iii פועלה מקבלת מספר חדר בין 1 ל-10, ומוצג, ומוסיפה את המוצג לחדר המתאים. הנה שיש למוצג זה מקום בחדר.

23. חברת טלפונים מפתחת פרויקט.  
התרשים שלפניך הוא תיאור חלקי של המחלקות שבפרויקט.



לחברה יש טלפונים קווים וטלפונים ניידים.  
המחלקה Phone מייצגת טלפון קוויי, והמחלקה CellPhone מייצגת טלפון נייד.  
יש טלפונים ניידים משני סוגים: Type2 , Type1  
לפניך מודיע על הפעולות שאפשר לבצע בטלפונים מכל אחד מהסוגים:

טלפון קווי:

— חיוג.

— קבלת שיחה.

טלפון נייד מסוג Type1 :

— חיוג.

— קבלת שיחה.

— ניהול זיכרון של מספרי טלפון: הוספת מספר, מחיקת מספר.

— צילום תמונה.

— הצגת תמונה.

— הצגת שעון.

— כיוון שעון.

(שים לב: המשך השאלה בעמוד הבא.)

## בטלפון נייד מסוג Type2 :

- חיוג.
- קבלת שיחה.
- ניהול זיכרון של מספרי טלפון: הוספה מספר, מחיקת מספר.
- הצגת שעון.
- כיוון שעון.

בממשק InterA מוגדרות הפעולות המטפלות בשעון: הצגת שעון, כיוון שעון.

בממשק InterB מוגדרות הפעולות המטפלות בתמונות: צילום תמונה, הצגת תמונה.

**א.** כדי לאפשר שימוש חוזר בקוד (code reuse), יש להוסיף לפרויקט מחלקה ובה טיפול בכל התכונות והפעולות המשותפות לטלפונים קווים ולטלפונים ניידים.  
בעבור מחלקה זו:

- i כתוב את כוורתה המחלקה ב-C#.
  - ii כתוב במילים את הפעולות שיש לכלול בה, ואילו תכונות יש להגדיר במחלקה כך שאפשר יהיה ממש פועלות אלה.
- ב.** העתק למחברתך את התרשים הנוכחי (בעמוד הקודם), והוסף לו את המחלקה שהגדרת בסעיף א.

צייר בתרשימים שבמחברתך את הקשרים החדשניים שנוצרו, והסביר אותם במילים.  
בנήחה שהוסיף לפרויקט את המחלקה שהגדרת בסעיף א,

בעבור בלאות מהמחלקות שבתרשים (לא כולל המשקדים, ולא כולל המחלקה שהגדרת בסעיף א):

- i כתוב את כוורתה המחלקה ב-C#.
  - ii כתוב במילים את הפעולות שיש לכלול בה, ואילו תכונות יש להגדיר במחלקה כך שאפשר יהיה ממש פועלות אלה.
- ג.** במחשב של נקודת השירות של החברה, מיצגים טלפונים מכל הסוגים, שנמצאים בתיקון. יש להוסיף לפרויקט את המחלקה הנדרשת לצורך ייצוג זה.
- כתב ב-C# את כוורתה המחלקה, ואת התכונות שלה, וכותב תיעוד לתכונות.

24. **לפניך פרויקט, ובו המחלקות MainApp , Derived3 , Derived2 , Derived1 , Base .**

```
public class Base
{
    protected static int[] arr;
    protected Base successor;

    public void SetSuccessor(Base b)
    {
        this.successor = b;
    }

    public static void SetArr(int[] arr)
    {
        Base.arr = arr;
    }

    public virtual void Opp()
    {
        Console.WriteLine ("Good Luck !");
    }
}

public class Derived1 : Base
{
    public override void Opp()
    {
        int sum = 0;
        for (int i = 0; i < Base.arr.Length; i++)
        {
            if (Base.arr[i] > 0)
            {
                sum += Base.arr[i];
            }
        }
        Console.WriteLine ("Sum :" + sum);
        if (this.successor != null)
            this.successor.Opp();
    }
}
```

(שים לב: המשך השאלה בעמוד הבא.)

```
public class Derived2 : Base
{
    public override void Opp()
    {
        int counter = 0;
        for (int i = 0; i < Base.arr.Length; i++)
        {
            if (Base.arr[i] <= 0)
            {
                counter++;
            }
        }
        Console.WriteLine ("Counter : " + counter);
        if (this.successor != null)
            this.successor.Opp();
    }
}

public class Derived3 : Base
{
    public override void Opp()
    {
        int counter = 0;
        for (int i = 0; i < Base.arr.Length; i++)
        {
            if (Base.arr[i] % 2 == 0)
            {
                counter++;
            }
            if (Base.arr[i] == 0)
                Console.WriteLine("*");
            else
                Console.WriteLine(Base.arr[i]);
        }
        Console.WriteLine("Counter : " + counter);
        if (this.successor != null)
            this.successor.Opp();
    }
}
```

```

public class MainApp
{
    public static void Main(string[] args)
    {
        Derived1 d1 = new Derived1();
        Derived2 d2 = new Derived2();
        Derived3 d3 = new Derived3();

        d1.SetSuccessor(d2);
        d2.SetSuccessor(d3);

        int[] arr = { 12, 13, -5, 0, -56, 34, 22, -30 };
        Base.SetArr(arr);
        d1.Opp();
    }
}

```

.א. כתוב מעקב אחר הפעולה Main במחלקה MainApp, וכותב את הפלט.

במעקב יש לכתוב את ערכי המשתנים, ובעבור כל עצם – את ערכי התכונות שלו.

.ב. מחליפים את הפעולה Main שבמחלקה MainApp בפעולה Main שלפניך:

```

public static void Main(string[] args)
{
    Derived1 d4 = new Derived1();
    d4.SetSuccessor(d4);
    int[] arr = { 12, 13, -5, 0, -56, 34, 22, -30 };
    Base.SetArr(arr);
    d4.Opp();
}

```

לפניך הטענה: ביצוע הפעולה Main אינו מסתiem.

קבע אם טענה זו נכונה או אינה נכונה, ונמק את קביעותך.

## בהצלחה!

זכות היוצרים שמורה למדינת ישראל  
אין להעתיק או לפרסם אלא ברשות משרד החינוך

תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב ב- C#, ענה על **שתיים** מהשאלות 24-21.  
**(לכל שאלה – 25 נקודות)**

21. לפניך חלק מפרויקט העוסק בכל תחבורת וכלל את המחלקות האלה:

מיצגת כלי תחבורה	המחלקה Vehicle
מיצגת רכבת שהיא כלי תחבורה	המחלקה Train
מיצגת סירה שהיא כלי תחבורה	המחלקה Boat
מיצגת מטוס שהוא כלי תחבורה	המחלקה Airplane
מיצגת חברה שיש לה כלי תחבורה מסווגים שונים	המחלקה TransportationCompany

public class **Vehicle**

1

```
private string type;           // מיקום (יבשה / מים / אוויר)  
private string way;           // סוג הדרך (כביש / מסילה / נهر / ...)  
private int maxSpeed;         // מהירות מקסימלית
```

```
public Vehicle(string type, string way, int maxSpeed)
```

{

```
this.type = type;  
this.way = way;  
this.maxSpeed = maxSpeed;
```

}

}

public class Train : Vehicle

{

private int numOfCarriages; // מספר הרכבות

```
public Train(int maxSpeed, int numOfCarriages) : base("land", "tracks",maxSpeed)  
{
```

```
    this.numOfCarriages = numOfCarriages;  
}
```

מגדילה ב- $n$  את מספר הרכבות // מס' הרכבות

{

```
        this.numOfCarriages = this.numOfCarriages + n;  
    }  
}
```

}

(שים לב: המשך השאלה בעמודים הבאים.).

```

public class Boat : Vehicle
{
    public Boat(string way, int maxSpeed) : base("water", way, maxSpeed)
    {
        }
}

public class Airplane : Vehicle
{
    private int maxHeight; // גובה טיסה מקסימלי
    public Airplane(int maxSpeed, int maxHeight) : base("sky", "air", maxSpeed)
    {
        this.maxHeight = maxHeight;
    }
}

public class TransportationCompany
{
    private Vehicle[] vehicles = new Vehicle[50]; // מערך כלי התחבורה בחברה
    private int counter = 0; // מספר כלי התחבורה שיש בפועל
    public TransportationCompany()
    {
    }

    public void AddVehicle (Vehicle v) // מוסיפה כלי תחבורה לערך כלי התחבורה של
    {                                       // החברה. הנח שיש מקום להוסיף כלי תחבורה.
        this.vehicles[counter] = v;
        this.counter++;
    }
}

```

(שים לב: סעיפי השאלה בעמוד הבא.)

**א.** ממש ב- C# מחלקה ראשית Program ובה פעולה ראשית, שתבצע את המשימות האלה:

i. בניית של עצם מטיפוס חברת של כלי תחבורה – **TransportationCompany**

הנקרא `company1`

ii. הוספה של סירה אחת ורכבת אחת לחברת `company1`

בחר לתוכנות ערכים כרצונך.

**ב.** במחלקה **TransportationCompany** הוגדרה הפעולה:

```
public void Display()
{
    for (int i=0; i<this.counter; i++)
    {
        Console.WriteLine((i+1) + ":" + this.vehicles[i]);
    }
}
```

משה ב- C# פועלות שיאפשרו ביצוע תקין של הפעולה `Display()`, כך שבמעבר כל כלי תחבורה יודפסו כל התוכנות שלו. הגדר את הפעולות באופן המתאים ביותר, לעקרונות של תוכנות מונחה עצמים (הכמזה – encapsulation, הורשה – inheritance, פולימורפיזם – polymorphism).  
בעבור כל פעולה שאתה ממשש, רשאי לאיזו מחלקה היא שייכת.  
אנן לשנות את הפעולה `Display()`.

**ג.** ממש ב- C# פעולה, שתתקבל מספר שלם `n` ותוסיף `n` קرونאות לכל הרכבות ששויות לחברת שיש לה כלי תחבורה מסוימים שונים. تعد את הפעולה, ורשום באיזו מחלקה יש להגדיר אותה. אנן לשנות את הפעולות הקיימות בפרויקט.

.22. בספרייה עירונית יש מאגר מידע על הספרים שבה. בעבור כל ספר נשמר הדירוג שהתקבל מקוראי הספר על מידת ההנאה שלהם ממנו. כאשר קורא מוחזיר בספרייה ספר הוא מקליד את הדירוג שלו, מספר שלם בין 0 ל-4, כאשר 4 מציין את מידת ההנאה הרבה ביותר. המידע על דירוג הספרים נשמר במאגר כך שאפשר לדעת כמה קוראים דירגו כל ספר בכלל אחת מחמש הדרגות האפשריות.

במאגר המידע יש מחלקת המיצגת **ספר** (Book) ומחלקת המיצגת **ספרייה** (Library). Library המציגת את המחלקות Book ו- :

<b>Book</b>		
private int code	//	קוד הספר
private string name	//	שם הספר
private string genre	//	סוג הספר (רומנים, מתח, ילדים ...)
private int numOfCopies	//	מספר העותקים של הספר שנמצאים
	//	עכשו בספרייה (לא מושאלים).
private int[] rating	//	מערך מונחים של דרגות ההנאה
	//	של הקוראים מהספר.
...	//	לכל תכונה מוגדרות פעולות Set ו- Get
public void IncNumOfCopies()	//	פעולה המגדילה ב- 1 את מספר העותקים של ספר שנמצאים עכשו בספרייה (לא מושאלים).
public double Score()	//	פעולה המחזיר ציון של ספר, המחשב על פי כל דרגות ההנאה שהספר קיבל.

<b>Library</b>		
private Book[] books	// //	מערך הספרים בספרייה. כל ספר מופיע במערך פעם אחד.

כאשר קורא מוחזיר בספרייה ספר, יש לבצע את המשימות האלה:

- קליטת הקוד של הספר לצורך זיהויו.
- עדכון של מספר העותקים של הספר שנמצאים עכשו בספרייה (לא מושאלים).
- קליטת הדירוג שנתן הקורא בספר, ועדכון מערך המונחים rating של הספר בהתאם.
- הדפסת הודעה שתזכיר אם בעקבות הדירוג של קורא זה, הציון של הספר עלה, ירד או לא השתנה.

משב- C#, במחלקות הנותנות, את הפעולות הנדרשות לטיפול בהזורה של ספר בספרייה. עליך להגדיר את הפעולות באופן המתאים ביותר לבניית המחלקות בפורוייקט. בעבור כל פעולה רשום את המחלקה שבה תוגדר, מה היא מקבלת, ומה היאמחזירה. הנה שלכל הפעולות הרשומות בדיאגרמות ה- UML יש מימוש במחלקות. הערת: אין צורך לבדוק את תקינות הקלט.

23. ברשות החנויות "גאדגיטון" יש 50 חנויות. הרשות מוכרת את המוצרים האלה:

נגני MP3 , נגני MP4 , אוזניות אלחוטיות.

הרשות זוקה לתוכנה שבמציאות יהיה אפשר לנחל את המלאי של כל אחת מהחנויות.

בעבור כל אחד מהמוצרים התוכנה צריכה לטפל בתנאים האלה:

- נגני MP3 : יצרן, דגם, מחיר, כמות מלאי, האם יש רדיו (כן / לא), האם יש רמקול פנימי (כן / לא)
- נגני MP4 : יצרן, דגם, מחיר, כמות מלאי, האם יש רדיו (כן / לא), האם יש רמקול פנימי (כן / לא), אורך מסך אוזניות אלחוטיות: יצרן, דגם, מחיר, כמות מלאי, טווח קליטה

כל אחד מהמוצרים האלה נמצא מלאי של כל אחת מהחנויות הרשות, ומכל מוצר יש דוגמים שונים בכמותות שונות.

הדרישות מהתוכנה נחלקות לשתי רמות: רמת החנות, רמת הרשות.

**הדרישות מהתוכנה ברמת החנות:**

- החזרת שווי המלאי שבחנות
- החזרת רשימת הדוגמים של מוצר, שהכמות שלהם מלאי החנות קטנה ממספר נדרש, limit,

**הדרישות מהתוכנה ברמת הרשות:**

- החזרת שווי המלאי שבכל הרשות
- החזרת רשימת הדוגמים של מוצר, שהכמות שלהם מלאי בכל הרשות קטנה ממספר נדרש, limit,

(שים לב: המשך השאלה בעמוד הבא.)

עליך לתכנן את המחלקות הנדרשות בעברור כתיבת התוכנה. התכנון של המחלקות צריך להיות באופן המתאים ביותר לעקרונות של תוכנות מונחה עצמים (הכמיסה – encapsulation , הירושה – inheritance , פולימורפיזם – polymorphism ). א. צייר את היררכיות המחלקות הנדרשות. לצורך השימוש בסימנים האלה:



- ב. בעברור כל מחלוקת שיכלلت בציור, הגדר את התוכנות והפעולות שלה.  
יש לכלול רק את הפעולות הנחוצות כדי לענות על הדרישות מהתוכנה שתוארו בתחילת השאלה.  
בעברור כל תוכנה, רשום את ההגדרה שלה ב- C# , ואת התיעוד שלה.  
בעברור כל פעולה, רשום את הכוורת שלה ב- C# , ורשום תיעוד הכלול מה היא מקבלת ומה היא מחזירה. אין צורך למש את הפעולה.  
אין צורך לרשום פעולות בונות, פעולות קבועות (פעולות Set) ופעולות אחזרות (פעולות Get) של התוכנות שאתה מגדר.

.24. לפניך פרויקט, ובו המחלקות Base , Derived1 , Derived2 , Derived3 . ובו המחלקות

```
public class Base
{
    protected int num;

    public Base (int n)
    {
        this.num = n;
    }

    protected virtual void DoSomeWork()
    {
        Console.WriteLine("num = " + this.num);
    }

    public void Run()
    {
        if (this.num%2 == 0)
            DoSomeWork();
    }
}

public class Derived1 : Base
{
    private int num1;

    public Derived1(int n , int n1) : base(n)
    {
        this.num1 = n1;
    }

    protected override void DoSomeWork()
    {
        Multiplication();
    }

    public void Multiplication()
    {
        Base.DoSomeWork();
        Console.WriteLine("num1 = " + this.num1);
        Console.WriteLine("num * num1 = " + this.num * this.num1);
    }
}
```

(שים לב: המשך השאלה בעמודים הבאים.).

```

public class Derived2 : Base
{
    private int num2;

    public Derived2(int n , int n2) : base(n)
    {
        this.num2 = n2;
    }

    protected override void DoSomeWork()
    {
        Division();
    }

    public void Division ()
    {
        Base.DoSomeWork();
        Console.WriteLine("num2 = " + this.num2);
        Console.WriteLine("num / num2 = " + this.num / this.num2);
    }
}

public class Derived3 : Base
{
    private int num3;

    public Derived3(int n, int n3) : base(n)
    {
        this.num3 = n3;
    }

    protected override void DoSomeWork()
    {
        Multiplication();
        Division();
    }

    public void Multiplication()
    {
        Base.DoSomeWork();
        Console.WriteLine("num3 = " + this.num3);
        Console.WriteLine("num * num3 = " + this.num * this.num3);
    }
}

```

(שים לב: המשך השאלה בעמוד הבא.)

```

public void Division()
{
    Base.DoSomeWork();
    Console.WriteLine("num3 = " + this.num3);
    Console.WriteLine("num / num3 = " + this.num / this.num3);
}
}

public class Program
{
    public static void Main(string[] args)
    {
        Base[] arr = new Base[5];
        arr[0] = new Derived1(12, 22);
        arr[1] = new Derived2(33, 44);
        arr[2] = new Derived3(54, 34);
        arr[3] = new Derived1(51, 72);
        arr[4] = new Derived2(58, 99);

        for (int i = 0; i < arr.Length; i++)
            arr[i].Run();

        if (arr[2] is Derived3)
        {
            arr[2].Run();
        }

        if (arr[3] is Derived2)
        {
            arr[3].Run();
        }
    }
}

```

כתב מעקב אחר הפעולה Main במחלקה Program, וכותב את הפלט.  
במעקב יש לכלול את ערכי המשתנים, ובabboר כל עצם — את ערכי התוכנות שלו.

## בצלחה!

זכות היוצרים שמורה למדינת ישראל  
אין להעתיק או לפרסם אלא ברשות משרד החינוך

**תכנות מונחה עצמים**

אם למדת מסלול זה ואתה כותב ב- C#, ענה על שתיים מהשאלות 21-24.  
(כל שאלת – 25 נקודות)

**.21. לפניך המחלקה מצולע – Polygon .**

חלק מתכונות המצלול והפעולות שלו מוגדים בגוף המחלקה.

```
public class Polygon
```

```
{
```

```
    public static int maxSides = 100;           // מספר צלעות מקסימלי
```

```
    private int[] values = new int[maxSides];    // מערך אורך הצלעות
```

```
    private int numSides = 0;                    // מספר צלעות במצולע
```

```
    /* פעולה בונה המחזירה מצולע ריק */
```

```
    public Polygon()
```

```
{}
```

```
    /* פעולה המחזיר את מספר הצלעות במצולע */
```

```
    public int GetNumSides()
```

```
{
```

```
    return this.numSides;
```

```
}
```

```
    /* פעולה המחזיר את מערך אורך הצלעות של המצלע */
```

```
    public int[] GetValues()
```

```
{
```

```
    return this.values;
```

```
}
```

```
    /* פעולה המתחלפת את maxSides */
```

```
    public static void SetMaxSides (int maxSides)
```

```
{
```

```
    Polygon.maxSides = maxSides;
```

```
}
```

```
    /* פעולה המקבלת מספר שלם וגדול מ-0 ומוסיפה למצולע צלע באירוע זה */
```

```
    /* הינה: יש מקום להוסיף צלע למערך אורך הצלעות */
```

```
    public void AddSide(int x)
```

```
{
```

```
    this.values[this.numSides] = x;
```

```
    this.numSides++;
```

```
}
```

```
    /* פעולה הקולטת מהמשתמש את מספר הצלעות שיש להוסify למצולע */
```

```
    /* ואת האורך של כל אחת מהן ומוסיפה את הצלעות למצולע. */
```

```
    /* הינה: יש מקום להוסיף את הצלעות למערך אורך הצלעות. */
```

```
    public void ReadToPolygon()
```

```
{.....}
```

```
}
```

```
/המשך בעמוד 41/
```

- א. מהי המשמעות של הגדרת התכונה `maxSides` כ- static ? הסבר ?
- ב. מה תהיה המשמעות של הגדרת התכונה `maxSides` אם היא תוגדר כ- static const ?
- ג. עליך למשב ב- C#, פעולה `Expand` המוסיפה למצולע צלעות של מצולע אחר, בכל אחד מהמקרים i-ii שלפניך:

i הפעולה `Expand` תוגדר כפעולה פנימית במחלקה `Polygon`.

ii הפעולה `Expand` תוגדר כפעולה סטטית חיצונית (לא במחלקה `Polygon`).

הערה: הנח כי במערך אורך הצלעות של המצולע יש די מקום להוספת כל הצלעות של המצולע الآخر.

ד. לפניך שلد של פעולה ראשית:

```
public static void Main(string[] args)
{
    Polygon p1 = new Polygon();
    p1.ReadToPolygon();
    Polygon p2 = new Polygon();
    p2.ReadToPolygon();
    (*) _____;
}
```

יש לרשום בשורה המסומנת ב-(\*), הוראה שתביא לכך שהצלעות של מצולע `p2` יתווסףו לצלעות של מצולע `p1`.

רשום במחברתך את ההוראה בעבור כל אחד מהמקרים i-iii :

i תוכן כדי שימוש בפעולת הפנימית `Expand` כפי שהוגדרה במחלקה `Polygon` לפי סעיף ג' .

ii תוכן כדי שימוש בפעולת הסטטית `Expand` כפי שהוגדרה בסעיף ג' אם היא מוגדרת בתוך המחלקה שבה נמצאת הפעולה `Main`.

iii תוכן כדי שימוש בפעולת הסטטית `Expand` כפי שהוגדרה בסעיף ג' אם היא מוגדרת במחלקה אחרת הנקראת `PolygonOperations`.

**22.** בוחנות למכירת אופניים אפשר לשלם באמצעות התשלום האלה: מזומנים, המחאה (ציק), כרטיס אשראי. את הסכום המשולם בהמחאות אפשר לחלק לכמה תשלוםים, התשלומים לא חייבים להיות שווים. תשלום בעבר כל קנייה יכול להתבצע באמצעות אחד או יותר, כך שילוב אמצעי התשלום ייתן את הסכום הנדרש. הנה כי הסכום לתשלום וכל אחד מהתשלומים הם מספרים שלמים.

לדוגמה:

- לפניך כמה שילובים אפשריים של אמצעי תשלום בעבר קנייה בסכום של 1000 ש"ח.
- תשלום את כל הסכום באמצעות תשלום אחד: מזומן או המחאה או כרטיס אשראי.
- תשלום 200 ש"ח במזומן ו- 800 ש"ח בכרטיס אשראי.
- תשלום 100 ש"ח במזומן, 500 ש"ח באמצעות שתי המחאות: הראשונה על סך 200 ש"ח והשנייה על סך 300 ש"ח, ו- 400 ש"ח בכרטיס אשראי.

הchnerות זוקקה לתוכנה כדי לנחל את תשלוםיו הקוניים.

בעבר כל קנייה, המידע המתקבל בוחנות הוא:

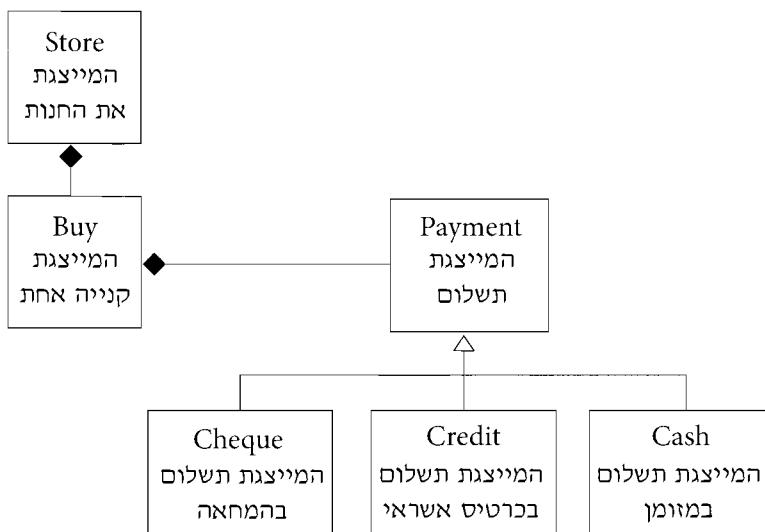
תאריך הקנייה, הסכום לתשלום בעבר הקנייה ופירוט השילוב של אמצעי התשלום.  
בעבר תשלום במזומן – הסכום לתשלום.  
בעבר תשלום בהמחאה – הסכום לתשלום, מספר ההמחאה, שם הבנק  
והתאריך הרשום על ההמחאה.

בעבר תשלום בכרטיס אשראי – הסכום לתשלום, מספר כרטיס האשראי, תוקף הכרטיס  
והתאריך שבו יחויב בעל כרטיס האשראי.

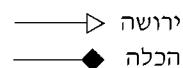
הפעולות הנדרשות מהתוכנה הן:

- קליטה ושמירה של המידע המתקבל בעבר כל קנייה.
- בדיקה שסכום כל התשלומים בעבר קנייה אחת שווה לסכום הדנייה – הדפסת קבלה בעבר הקנייה.
- בעבר תאריך מסוים, חישוב של סכום מחירי כל האופניים שנימרכו באותו תאריך.

לפניך תרשים של היררכיה המחלקות הנדרשות בעבר כתיבת התוכנה.



בתרשים מופיעים הסימנים הבאים:



בעבור כל מחלוקת בתרשימים, הגדר את התכונות ואת הפעולות שלה באופן המתאים ביותר, **לעקרונות של תכונות מונחה עצמים** (הכמסה — **encapsulation** , הירושה — **inheritance** , פולימורפיזם — **polymorphism** ).

יש לכלול רק את הפעולות הנוחות כדי לענות על הדרישות מהתוכנה שתווארו בשאלה.

אין צורך לרשום פעולות בוניות, פעולות קובעות (Set) ופעולות מאחרזרות (Get).

הנח שקיימת מחלוקת Date , המייצגת תאריך.

בעבור כל תוכנה, רשום את ההגדרה שלה בـ C# , ורשום את התיעוד שלה.

בעבור כל פעולה, רשום את הכוורת שלה בـ C# , ורשום תיעוד הכלול מה היא מקבלת ומה היאמחזירה. אין צורך למש את הפעולה.

.23. לפניך הממשקים: IPrintBinary, IPrintHtml

והמחלקות: .CreateReport, Page2, Page1, Page

```
public interface IPrintHtml
{
    public void CreateHtml();
}

public interface IPrintBinary
{
    public void CreateBinary();
}

public class Page : IPrintBinary
{
    protected int num;

    public Page()
    {}

    public Page(int n)
    {
        this.num = n;
    }

    public virtual void Print()
    {
        Console.WriteLine (this.num);
    }

    public virtual void CreateBinary()
    {}
}

public class Page1 : Page
{
    protected int num1;

    public Page1 (int n, int n1) : base(n)
    {
        this.num1 = n1;
    }
}
```

```
public override void Print()
{
    base.Print();
    Console.WriteLine (this.num1);
}

public override void CreateBinary()
{
    Console.WriteLine ("Binary Data: num = " + this.num + ", num1 = " + this.num1);
}

public class Page2 : Page, IPrintHtml
{
    protected int num2;

    public Page2(int n, int n2): base(n)
    {
        this.num2 = n2;
    }

    public override void Print()
    {
        base.Print();
        Console.WriteLine (this.num2);
    }

    public void CreateHtml()
    {
        Console.WriteLine ("Html Data: num = " + this.num + ", num2 = " + this.num2);
    }
}

public class CreateReport
{
    public static void CreateBinaryDoc(IPrintBinary doc)
    {
        Console.WriteLine("*** Binary Doc ***");
        doc.CreateBinary();
    }

    public static void CreateHtmlDoc(IPrintHtml doc)
    {
        Console.WriteLine("*** Html Doc ***");
        doc.CreateHtml();
    }
}
```

(שים לב: סעיפים השאלה בעמוד הבא.)

**א.** לפניך שלושה קטעים i-iii הכתובים ב-C#.

לכל אחד מהקטעים, קבע אם הוא חוקי או אינו חוקי. נמק את קביעותיך.

i Page1 doc1 = new Page1 (10 , 20);

CreateReport.CreateHtmlDoc (doc1);

ii Page doc2 = new Page2 (30 , 40);

CreateReport.CreateHtmlDoc (doc2);

iii IPrintBinary doc3 = new Page1 (50 , 60);

CreateReport.CreateBinaryDoc (doc3);

**ב.** הפעולה WriteHtmlDoc שלפניך הוספה למחלקה CreateReport.

לאחר ההוספה הקומפיילר הודיע על שנייה. הסבר מהי השניה, ותקן אותה.

```
public static void WriteHtmlDoc(Page doc)
```

```
{
```

```
    if (doc is Page2)
```

```
        doc.CreateHtml();
```

```
}
```

**ג.** כתוב את הפלט של קטע התכנית שלפניך:

Page2 doc1 = new Page2 (11 , 22);

CreateReport.CreateHtmlDoc (doc1);

doc1.Print();

Page doc2 = new Page1 (33 , 44);

CreateReport.CreateBinaryDoc (doc2);

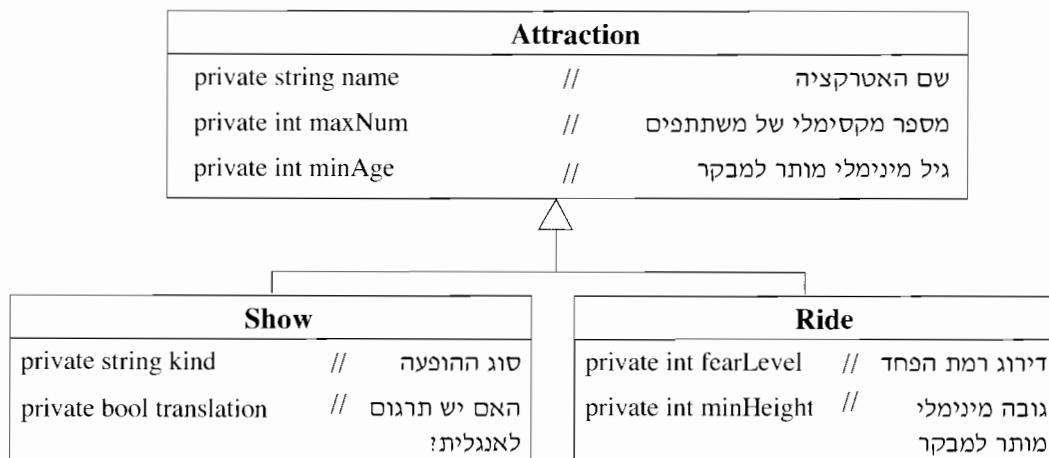
doc2.Print();

Page doc3 = new Page2 (55 , 66);

CreateReport.CreateBinaryDoc (doc3);

doc3.Print();

24. פארק שעשויים מציע למבקריו אטרקציות מסווגים שונים. חלק מהאטרקציות הן מתקנים כמו קורסלה או רכבת הרים) וחלק מהאטרקציות הן הופעות. נתונה דיאגרמת UML של חלק מהמחלקות במערכת הממוחשבת של הפארק. המחלקה **Ride** בעבר אטרקציה, המחלקה **Show** בעבר מתקן, והמחלקה **Attraction** בעבר הופעה. המחלקות **Show** ו- **Ride** יירושות מהמחלקה **Attraction**.



הנה שbulk את שלוש המחלקות מוגדרות:

- פעולות Set ו- Get בעבר כל התכונות.

פעולה בונה מקבלת פרמטרים לכל תכונה, ומתחילה את תוכנות העצם בהתאם.

a. יש להוסיף למערכת הממוחשבת מחלקה **Park** ובו התכונות:

שם הפארק, מערך האטרקציות בפארק.

ממש ב- C# את הסעיפים i-iii במחלקה **Park**

i כוורת המחלקה והתכונות שלה.

ii פעולה בונה, מקבלת את שם הפארק ואת מספר האטרקציות שהפארק יכול להכיל. הפעולה תאותל את שם הפארק ותאותל את מערך האטרקציות בפארק להיות בגודל שהתקבלה.

iii פעולה המדפיסה את שמות הופעות שיש להן תרגום לאנגלית.

b. במחלקה **Park** נתונה הפעולה AddAttraction מקבלת אטרקציה **Attraction**

ומוסיפה אותה למערך האטרקציות בפארק. כוורת הפעולה היא:

```
public void AddAttraction (Attraction a)
```

כתוב ב- C#, פעולה ראשית שתיצור עצם k מטיפוס **Park** בשם lunafun ויש בו

מקום ל- 30 אטרקציות. כמו כן, הפעולה הראשית תיצור 2 עצמים: אחד מטיפוס **Ride**

ואחד מטיפוס **Show**, ותוסיף אותם למערך האטרקציות של k.

בחר ערכים כרצונך לאותול 2 העצים.

## בהתכלחה!

זכות היוצרים שמורה למדיינית ישראל  
אין להעתיק או לפרסם אלא ברשות משרד החינוך

**תכנות מונחה עצמים**

אם למדת מסלול זה ואתה כותב ב- C#, ענה על שתיים מהשאלות 21-24.

לכל שאלה – 25 נקודות

21. בגין החיות "ZooZoo" בעלי החיים יודעים "לדבר". חיוט רעבות משמשות את קולו בדרישה לקבל אוכל. כאשר מטפל מגע להאכיל את החיות, החיות מקבלות את פניו בקריאות שמחה. החיות משמשות קול כשהן מסיימות לאכול. לפניך פרויקט המדמה את גן החיות (ZooZoo) , את בעלי החיים בגין החיות ואת "אמרות החכמה" שהם משמשים.
- . (Marine) או בעל חיים ימי (Animal) יכול להיות זוחל (Reptile) או תנין (Crocodile) או נחש (Snake) או דג זהב (GoldFish). בעל חיים ימי (Marine) יכול להיות דג זהב (GoldFish) . כל בעל חיים נשמרשמו (name). בעבור זוחל נשמר גם האורך שלו (len), ובעבור בעל חיים ימי נשמר גם עומק הצלילה המקסימלי שלו (depth). הפעולה Hungry מחזירה את דבריו של בעל חיים רעב. הפעולה Caretaker מחזירה את דבריו של בעל חיים המקבל את פניו המטפל שלו. הפעולה Satisfied מחזירה את דבריו של בעל חיים שסיים לאכול.

```
public class Animal
{
    private string name;           // שם בעל חיים
    public static int count = 0;    // מונה בעלי חיים

    public Animal(string name)
    {
        this.name = name;
        count++;
    }

    public string Hungry() { return this.name + " is hungry!"; }      // רעב
    public virtual string Caretaker() { return "Yammi"; }             // קבלת המטפלים
    public virtual string Satisfied() { return "Finish eating"; }       // סיום לאכול
}

public class Reptile: Animal
{
    private int len;               // אורך
    public Reptile(string name, int len): base(name)
    {
        this.len = len;
    }

    public override string Caretaker() {return "Crawling: " + base.Caretaker(); }
}
```

(שים לב: המשך השאלה בעמוד הבא.)

/המשך בעמוד 46/

```
public class Snake: Reptile
{
    public Snake(string name, int len): base (name, len) { }
    public override string Satisfied() { return "Tssss"; }
}

public class Crocodile: Reptile
{
    public Crocodile(int len): base("crocki" + count , len){ }
    public override string Satisfied() { return base.Satisfied() + " Whaamm"; }
}

public class Marine: Animal
{
    private double depth;           // עומק

    public Marine(string name, double depth): base(name)
    {
        this.depth = depth;
    }
    public override string Caretaker() { return "Swimming"; }
}

public class GoldFish: Marine
{
    public GoldFish(string name): base(name, 0.2) { }
    public override string Satisfied() { return "Bloop bloop"; }
}
```

```

public class ZooZoo
{
    private Animal[] animals;

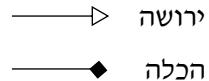
    public ZooZoo()
    {
        this.animals = new Animal[4];
        this.animals[0] = new Snake("snaki", 50);
        this.animals[1] = new Crocodile(78);
        this.animals[2] = new GoldFish("goldi");
        this.animals[3] = new Crocodile(103);
    }

    public void Print()
    {
        for (int i = 0; i < this.animals.Length; i++)
        {
            Console.WriteLine(this.animals[i].Hungry());
            Console.WriteLine(this.animals[i].Caretaker());
            Console.WriteLine(this.animals[i].Satisfied());
            Console.WriteLine("*****");
        }
    }
}

public class Test
{
    public static void Main(string [] args)
    {
        ZooZoo zoo = new ZooZoo();
        zoo.Print();
    }
}

```

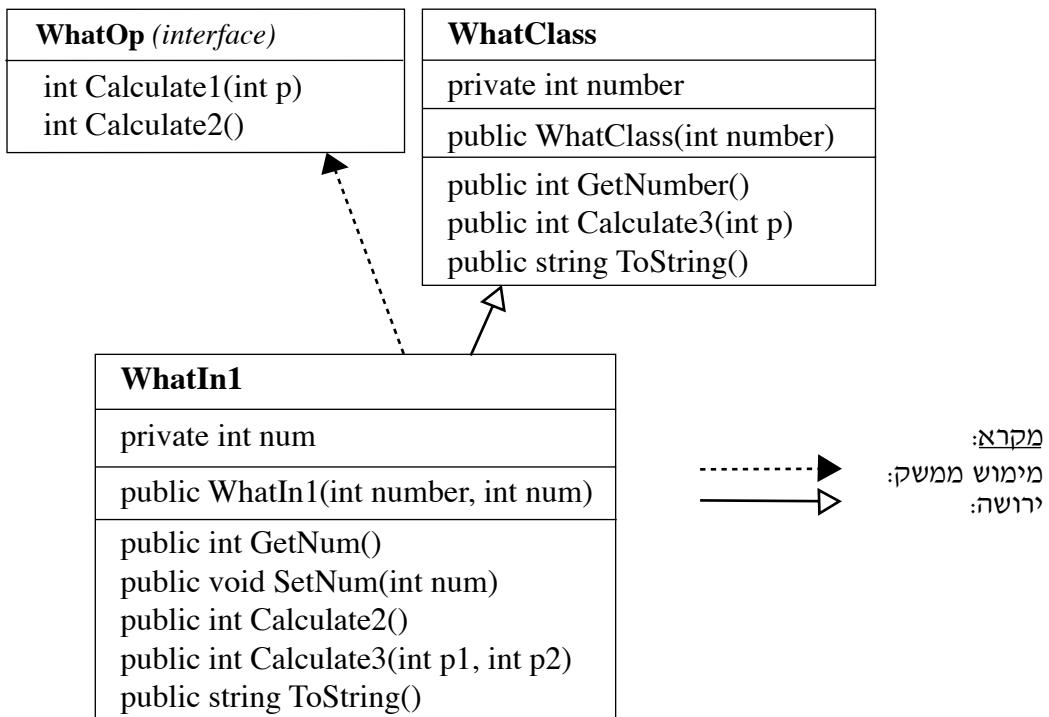
**א.** סרטט את היררכיות המחלקות המוגדרות בפרויקט. השתמש בסימנים האלה:



**ב.** כתוב מעקב אחר הפעולה Main שבמחלקה **Test**, וכתוב את הפלט.

במעקב יש לכלול את ערכי המשתנים, ובabboר כל עצם – את ערכי התכונות שלו.

## 22. לפניך UML חלקי:



**a.** אם מימוש הפעולה `Calculate1(int p)` במחלקה **WhatClass** במאפשר על דרישת המימוש שלה במחלקה **WhatIn1**? נמק את תשובתך.

**b.** מימוש הפעולה הבונה במחלקה **WhatIn1** הוא:

```

public WhatIn1(int number, int num)
{
    this.number = number;
    this.num = num;
}
  
```

האם הפעולה תקינה? אם כן – תאר את ביצוע הפעולה; אם לא – תקן את הפעולה (אין לשנות דבר פרט לפעולת עצמה). רשום במחברתך את הפעולה המתוקנת.

במחלקה **WhatIn1** מומשא הפעולה `Calculate2()`, המחזירה את הערך השם של

ממוחצע תכונות העצם:

```
public int Calculate2()
{
    return (int)((this.number + this.num)/2); }
```

האם הפעולה תקינה? אם כן – תאר את ביצוע הפעולה; אם לא – תקן את הפעולה (אין לשנות דבר פרט לפעולת עצמה). רשום במחברתך את הפעולה המתוקנת.

לפניך תיאור חלקו של המחלקה **WhatIn2** :

class <b>WhatIn2</b> : <b>WhatIn1</b>
private int sum
public WhatIn2(int number, int num, int sum)
public int Calculate3(int p1, int p2, int p3)

(1) האם אפשר להסתמך על הפעולה הבונה בירرت מחדל במקום להגדיר פעולה בונה במחלקה **WhatIn2**? נמק את תשובתך.

(2) הפעולה `Calculate3` מומשא בשלוש המחלקות **WhatIn1**, **WhatClass** ו- **WhatIn2** באופן זה:

class <b>WhatClass</b>	public int Calculate3(int p) {   return this.number * p;   }
class <b>WhatIn1</b>	public int Calculate3 (int p1, int p2) {   return this.Calculate3(p1) + this.num * p2 * p2;   }
class <b>WhatIn2</b>	public int Calculate3 (int p1, int p2, int p3) {   return this.Calculate3(p1, p2) + this.sum * p3 * p3 * p3;}

הנח כי העצם `obj` הוא מטיפוס **WhatIn2**, וערכי התכונות שלו הם:

`number = 1`

`num = 2`

`sum = 3`

לפניך הוראה כתובה בפעולת הראשית:

```
Console.WriteLine(obj.Calculate3(1000, 100, 10));
```

הראה את המעקב אחר ביצוע ההוראה. במעקב התייחס לזמן פעולה ולערכי תכונות העצם. רשום את הפלט המתקבל.

. 23. לפניך כמה עקרונות של תכנות מונחה עצמים:

- הapsulation — encapsulation
- העמסה — overloading
- ירושה — inheritance
- דרישת overriding
- רב-צורתיות — polymorphism

בשאלה זו תהיה התיחסות לחלק מהם.

. Stam, Davar: לפניך מימוש חלקי של שתי מחלקות:

```
public class Stam
{
    private char x;

    public Stam() { this.x = '*'; }
    public Stam(char c) { this.x = c; }
    public Stam GetStam() { return this; }
    public override string ToString() { return "x=" + this.x; }
    public bool IsSame1(Stam other)
    {
        התיחסות למימוש בהמשך השאלה //
    }
    public bool IsSame2(Stam other)
    {
        התיחסות למימוש בהמשך השאלה //
    }
    public void Same(Stam other)
    {
        if (this.IsMatch1(other))
            Console.WriteLine(this + " same1 as " + other);
        else
            Console.WriteLine(this + " not same1 as " + other);
        if (this.IsMatch2(other))
            Console.WriteLine(this + " same2 as " + other);
        else
            Console.WriteLine(this + " not same2 as " + other);
    }
    public void Print() { Console.WriteLine(this.ToString()); }
    public void Print(Stam other) { this.Same(other); }
}
```

```
public class Davar: Stam
{
    private int y;
    public Davar(): base() { this.y = 0; }
    public Davar(char c) : base(c) { this.y = 0; }
    public Davar(char c , int num): base(c) { this.y = num; }
    public override string ToString() { return "Davar: " + base.ToString(); }
}
```

א. במחלקה **Stam** הוגדרו שתי פועלות בונות.

(1) מהו העיקרון של תכנות מונחה עצמים ( מבין העקרונות שצוינו בתחילת השאלה )

האפשר זnat?

(2) כיצד בוחר המהדר (קומפיילר) איזו פעולה בונה להפעיל?

ב. איזה עיקרון של תכנות מונחה עצמים ( מבין העקרונות שצוינו בתחילת השאלה )

ממומש בפעולה () במחלקה **Davar** ? נמק.

(שים לב: סעיף ג של השאלה בעמוד הבא.)

/המשך בעמוד 52/

ג. נתונה המחלקה הראשית שלהן:

```
public class Program
{
    public static void Main(string[] args)
    {
        Stam[] s = new Stam[6];
        s[0] = new Stam();
        s[1] = new Davar();
        s[2] = new Stam('b');
        s[3] = new Davar('b');
        s[4] = new Davar('a', 0);
        s[5] = s[2].GetStam();

        for (int i = 0; i < s.Length; i++)
        {
            s[i].Print();
        }

        s[1].Print(s[0]);
        s[2].Print(s[5]);
        s[3].Print(s[4]);
    }
}
```

קטע 1

קטע 2

קטע 3

- (1) הצג את המערך s הנבנה בקטע 1 (במחלקה הראשית הנתונה).  
בעבור כל אחד מהעצמים רשום את ערכי התכונות שלו.
- (2) רשום את פלט הלולאה בקטע 2.
- (3) איזה עיקרון של תכונות מונחה עצמים ( מבין העקרונות שצויינו בתחילת השאלה )  
בא לידי ביטוי בקטע 2 ?
- (4) ממש את הפעולות IsSame2() , IsSame1() שבמחלקה Stam כך שהפלט  
המתתקבל מקטע 3, יהיה הפלט שלהן :

Davar: x = * same1 as x = *
Davar: x = * not same2 as x = *
x = b same1 as x = b
x = b same2 as x = b
Davar: x = b not same1 as Davar: x = a
Davar: x = b not same2 as Davar: x = a

24. במפעל לייצור נרות יש מקום ל- 12 פסי לייצור לכל היותר. בכל פס לייצור מיצרים דגם אחד

של נרות. לכל דגם יש כמה צבעים שאפשר לייצר מהם את הנרות מאותו הדגם. כל נר הוא

בצבע אחד בלבד מבין הצבעים בדגם שלו.

בעבור כל דגם של נרות נשמר המידע: שם הדגם, קוד הדגם, שמות הצבעים מהם הוא

מיוצר, כמה נרות שיש במלאי מכל אחד מהצבעים של הדגם, וכמה נרות המקסימלית

מדגם זה שאפשר לאחסן ( מכל הצבעים יחד).

דוגמה לדגם נרות:

— שם: "שורשנה".

— קוד: 11.

— שמות הצבעים מהם הוא מיוצר: אדום, לבן, צהוב.

— כמה נרות במלאי: 200 נרות אדומים, 350 נרות לבנים ו- 70 נרות צהובים.

— הכמות המקסימלית של הנרות מדגם זה ( מכל הצבעים יחד) שאפשר לאחסן:

.100,000.

במפעל נשמר מידע בנוגע לכל דגמי הנרות המיוצרים בו.

בכל פעם שרוצים לייצר נר מדגם מסוים יש להתייחס לתנאים האלה:

- צבע הנר ייבחר על פי כמה נרות במלאי: הצבע שכמות הנרות ממנו במלאי היא הקטנה ביותר מאותו דגם, הוא הצבע שייבחר לייצור הנר (אם יש יותר מצבע אחד כזה ייבחר אחד מהם).

- כמה הבחירה נתונה לבחירת המפעיל, אך אסור שכמות הנרות הכוללת מדגם מסוים (הנרות שייצרו והנרות שבמלאי) תעלה על הכמות המаксימלית מדגם זה שאפשר לאחסן.

מהנדס התוכנה של המפעל בנה פרוייקט ובו שלוש מחלקות:

<b>CandleKind</b>	מחלקה המייצגת דגם של נרות
<b>Factory</b>	מחלקה המייצגת את המפעל
<b>Run</b>	מחלקה ראשית

(שים לב: המשך השאלה בעמוד הבא.)

לפניך חלקים מן המחלקה **CandleKind**

<b>public class CandleKind</b>	
<b>private string name;</b>	שם הדגם //
<b>private int code;</b>	קוד הדגם //
<b>private string[] colors;</b>	מערך של מגוון צבעי הדגם //
<b>private int[] amounts;</b>	כמות במלאי מכל צבע (בהתאמה) //
<b>private int maxTotalAmount;</b>	מכסה מקסימלית לאחסון //
<b>public CandleKind(string name, int code, string[] colors, int maxTotalAmount)</b>	
{	
this.name = name;	
this.code = code;	
this.colors = colors;	
this.amounts = new int[colors.Length];	
for (int i = 0; i < amounts.Length; i++)	
{ this.amounts[i] = 0; }	
this.maxTotalAmount = maxTotalAmount;	
}	
<b>public void Update(string color, int amount)</b>	
	הפעולה <u>מנגדילה</u> בכמות amount את כמות הנרותצבע color שבמלאי //

א. כתוב הוראה / הוראות לבניית עצם בשם kind1 מן המחלקה **CandleKind** בעברוי

דגם חדש של נרות שמתחלים לייצר בפועל. הדגם החדש מיוצר בשלושה צבעים  
שוניים. בחר ערכים לתכונות כרצונך. רשום את ערכי תכונות העצם לאחר בנייתו.

ב. כתוב את כוורתת המחלקה **Factory** ואת התוכנה / התכונות שלה. כתוב תיעוד לכל  
תמונה.

ג. לפניך טבלה עם תיעוד של פעולות 1-3 שיש להוציא לפועל. לכל פעולה כתוב:  
i באיזו מחלקה (מבין שלוש המחלקות) מתאים להגדר אותה. הסבר את בחירתך.  
ii את הכוורות של הפעולה.

1	פעולה המחזירה צבע של נר מדגם נרות מסוים שיתחילו לייצר.
2	פעולה המחזירה את כמות הנרות שאפשר לייצר מדגם נרות מסוים.
3	פעולה המחזירה את הקוד של דגם הנר שבבورو ההפרש בין הכמות שיש מדגם זה במלאי לבין הכמות שאפשר לאחסן הוא הגדל ביותר. אם יש כמה דגמי נרות כאלה, יוחזר אחד מהם. אם הכמות במלאי של כל אחד מדגמי הנרות היא המקסימלית שאפשר לאחסן, יוחזר הקוד 999 (קוד לא קיים).

לפניך פעולה ראשית המומומשת במחלקה **Run**. הפעולה מנהלת את תהליכי הייצור. השלים את קטעי הקוד הממוספרים (1)-(4) על פי פירוט הדרישות המופיע אחרי הקוד. הנה כי הפעולות שהגדרת בסעיף ג' קיימות, ומוגדרות פועלות `Get`, `Set` לכל תוכינה במחלקה **CandleKind** הנזונה ובמחלקה **Factory** שכתבת בסעיף ב'. אם אתה מגדיר פעולות נוספות, רשאי באיזו מחלקה הן מוגדרות ומשמש אותן באופן מלא.

```
public static void Main(string [] args)
{
    Factory fty;
    //                                           מכאן והלאה הנה שקיים העצם fty עם ערכי
    //                                           תכונות המתארים את מצב המלאי הקיים בפעול
    int code = _____ (1)

    while (code != 999)
    {
        CandleKind ck;
        _____ (2)
        string colorToProduce;
        _____ (3)
        int amount;
        _____ (4)
        ck.Update(colorToProduce, amount);
        code = _____ (1)
    }
}
```

פירוט הדרישות:

- (1) זימון פעולה 3 מהטבלה שבסעיף ג'.
- (2) קטע קוד אשר יביא לכך שהמשתנה `ck` יוכל את דגם הנר בעל הקוד `code`.
- (3) קטע קוד אשר יביא לכך שהמשתנה `colorToProduce` יוכל את הצבע שיש לייצר מדגם הנר שהקוד שלו הוא `code`.
- (4) קטע קוד אשר יביא לכך שהמשתנה `amount` יוכל את כמות הנרות שאפשר לייצר מדגם הנר שהקוד שלו הוא `code`.

## בצלחה!

**תכנות מונחה עצמים**

אם למדת מסלול זה ואתה כותב ב- **C#**, ענה על **שתיים** מהשאלות 21-24.  
(כל שאלה – 25 נקודות)

.21. לפניך המחלקות **AA** ו-**BB**:

public class **AA**

{

    private string st;

    public AA() { this.st = "excellent"; }

    public AA(string st) { this.st = st; }

    public string GetSt() { return this.st; }

    public void SetSt (string st) { this.st = st; }

    public override string ToString() { return "st = " + this.st; }

}

public class **BB** : **AA**

{

    private int num;

    public BB() : base() { this.num = 1; }

    public BB(int num, string st) : base(st) { this.num = Math.Abs(num); }

    public int GetNum() { return this.num; }

    public void SetNum(int num) { this.num = num; }

    public override string ToString() { return base.ToString() + " num = " + this.num; }

}

.א. הגדר במחלקה **AA** פועלה בוליאנית הניתנת לדרישה, בשם (Object obj) המקבלת עצם obj מטיפוס **Object**. אם העצם obj הינו מטיפוס **AA** וגם תוכן המחרוזת st של obj זהה לתוכן המחרוזת st של העצם הנוכחי – הפועלה תחזיר true, אחרת – תחזיר false.

.ב. הגדר במחלקה **BB** פועלה הדורשת את הפועלה שהגדרת בסעיף א. אם העצם obj הינו מטיפוס **BB** וגם ערך התוכונה sum שלו זהה לערך התוכינה num של העצם הנוכחי – הפועלה תחזיר true, אחרת – תחזיר false.

ג. לפניך קטע מפעולה ראשית:

```
AA a = new AA("excellent");
BB b = new BB();
a = b;
if (a.IsLike(b)) Console.WriteLine(a);
```

האם קטע התוכנית תקין?

אם כן — מה יהיה פלט הקטע? רשוםஇயோ גרסה של הפעולה IsLike תופעל — זו של AA או זו של BB.  
אם לא — הסבר מהי השגיאה ומתי היא מתגללה: בזמן קומpileציה או בזמן ריצה.

ד. לפניך קטע מפעולה ראשית:

```
AA aa = new AA();
BB bb = new BB(2 , "excellent");
bb = aa;
if (bb.IsLike(aa)) Console.WriteLine(bb);
```

האם קטע התוכנית תקין?

אם כן — מה יהיה פלט הקטע? רשוםஇயோ גרסה של הפעולה IsLike תופעל — זו של AA או זו של BB.  
אם לא — הסבר מהי השגיאה ומתי היא מתגללה: בזמן קומpileציה או בזמן ריצה.

ה. כתוב פעלת חיצונית בשם LongString המתקבלת מערך של עצמים מטיפוס Object. הפעלת מחזירה מחרוזת המורכבת משרשור התכונה st של עצמים מטיפוס AA במערך, באופן זה:  
— אם לעצם יש רק התכונה st , תשורשר המחרוזות שבתכונה st פעם אחת.  
— אם לעצם יש גם התכונה sum , המחרוזות שבתכונה st תשורשר sum פעמיים.  
— אם אין במערך אף עצם מטיפוס AA , תוחזר מחרוזת ריקה.

לפניך פרויקט ובו המחלקות **A**, **B**, **D** .**OopTest**

```

public class B
{
    private static int numB = 0;
    private int m1;
    private int m2;

    public B(int m1, int m2)
    {
        this.m1 = m1;
        this.m2 = m2;
        numB++;
        Console.WriteLine("B(" + m1 + ", " + m2 + ") , #" + numB);
    }
}

public class D : B
{
    private static int numD = 0;
    private double d;

    public D(double d, int x) : base (x , x)
    {
        this.d = d;
        numD++;
        Console.WriteLine("D(" + d + ", " + x + ") , #" + numD);
    }

    public D(double d, int x, int y) : base(x , y)
    {
        this.d = d;
        numD++;
        Console.WriteLine("D(" + d + ", " + x + " , " + y + ") , #" + numD);
    }
}

```

המשך בעמוד /49

```

public class A
{
    private static int numA = 0;
    private A a;
    private B b;

    public A(A a, B b)
    {
        this.a = a;
        this.b = b;
        numA++;
        Console.WriteLine("A Constructor , #" + numA);
    }
}

public class OopTest
{
    public static void Main(string[] args)
    {
        B w1 = new B(2, 3);
        B w2 = new D(1.5, 6);
        B w3 = new D(2.3, 8, 9);
        A w4 = new A(null, w1);
        A w5 = new A(w4, w3);
    }
}

```

כתב מעקב אחר הפעולה Main במחלקה **OopTest** , וכותב את הפלט.  
במעקב יש ל כתוב את ערכי המשתנים, ובעבור כל עצם – את ערכי התכונות שלו.

.23. חברת המפיקת לומדות פיתחה לומדה העוסקת בסדרות של מספרים שלמים. המערך פותחה בשלבים.

עבור כל סדרה של מספרים מתייחסים אל:

- (1) האיבר הראשון בסדרה שמספרו הסידורי הוא 1.
- (2) האיבר שמספרו הסידורי בסדרה הוא 2.
- (3) הדפסת כל האיברים הראשונים בסדרה.

**בשלב הראשון** פותחו שתי המחלקות:

סדרה חשבונית (**ASeq**) – סדרה שבה הפרש בין כל איבר לקודמו הוא ערך קבוע.  
סדרה הנדסית (**GSeq**) – סדרה שבה המנה בין כל איבר לקודמו היא ערך קבוע.

להלן קוד המחלקות שפותחו בשלב הראשון:

```
public class ASeq
```

```
{
```

```
    private int first;
```

```
    private int difference;
```

```
    public ASeq(int first, int difference)
```

```
{
```

```
        this.first = first;
```

```
        this.difference = difference;
```

```
}
```

```
    public int TheNElement(int n)
```

```
{
```

```
        return this.first + (n - 1) * this.difference;
```

```
}
```

```

public void DisplayNElements(int n)
{
    Console.WriteLine("The sequence elements: ");
    for (int i = 0; i < n - 1; i++)
        Console.WriteLine(this.TheNElement(i + 1) + " , ");
    Console.WriteLine(this.TheNElement(n));
}

public class GSeq
{
    private int first;
    private int product;

    public GSeq(int first, int product)
    {
        this.first = first;
        this.product = product;
    }

    public int TheNElement(int n)
    {
        return this.first * (int)Math.Pow(this.product, n - 1);
    }

    public void DisplayNElements(int n)
    {
        Console.WriteLine("The sequence elements: ");
        for (int i = 0; i < n - 1; i++)
            Console.WriteLine(this.TheNElement(i + 1) + " , ");
        Console.WriteLine(this.TheNElement(n));
    }
}

```

א. עקוב אחר קטע התוכנית שלפניך. במקבץ הצג את העצם שנבנה, את התכונות שלו ואת הפלט.

```
ASeq aSeq = new ASeq(2 , 3);
Console.WriteLine(aSeq.TheNElement(4));
aSeq.DisplayNElements(5);
```

**בשלב השני** של הפיתוח הוחלט שמתאים לפתח מחלקה חדשה המותאמת סדרה קבועה (**Sequence**), כך שהמחלקות **ASeq** ו- **GSeq** יירשו מן המחלקה החדשה. בסדרה קבועה מוגדר ערך האיבר הראשון, וכל יתר האיברים זהים לאיבר הראשון.

ב. השלם את הפיתוח של שלב השני באופן המתאים ביותר לעקרונות של תכונות מונחה עצמים ובהתאם להנחיות (i)-(ii):

(i) ממש באופן מלא את מחלקת העל **Sequence**. המחלקה צריכה להתייחס אל:

- (1) האיבר הראשון בסדרה שמספרו הסידורי 1.
- (2) האיבר שמספרו הסידורי בסדרה הוא ז.
- (3) הדפסת ז האיברים הראשונים של הסדרה.

(ii) ממש מחדש את המחלקה **ASeq** כך שתירוש מן המחלקה **Sequence**.

**בשלב השלישי** של הפיתוח הוחלט להרחיב את הפרויקט שכולל את שלוש המחלקות שפותחו בשלב השני (**Sequence** , **ASeq** , **GSeq**), כך שבמעבר כל סדרה יהיה אפשר להפעיל פעולה המחשבת ומהזירה את סכום ז האיברים הראשונים של הסדרה. הנתן כי המחלקה **GSeq** מומשה מחדש כך שהיא יורשת מן המחלקה **Sequence**.

ג. מעבור כל אחת מן המחלקות כך שהיא יונחה עצמים. כתוב אם יש לעשות בה שינויים כך שהפרויקט יענה על דרישות הפיתוח של שלב השלישי באופן המתאים ביותר לעקרונות של תכונות מונחה עצמים. אם יש לעשות שינויים — פרט וממש אותם.

- בשלב הרביעי** של הפיתוח הוחלט לפתח פעולה סטטיט Check המקבלת ערך שלם או שני עצמים של סדרות: האחד מטיפוס **ASeq** והאחר מטיפוס **GSeq**.
- הפעולה מחשבת את סכום מה האיברים הראשונים בכל אחת משתי הסדרות ומחזירה:
- את התו 'A' – אם סכום מה האיברים הראשונים של הסדרה מטיפוס **ASeq** הוא גדול מבין שני הסכומים.
  - את התו 'G' – אם סכום מה האיברים הראשונים של הסדרה מטיפוס **GSeq** הוא גדול מבין שני הסכומים.
  - את התו 'E' – אם סכום מה האיברים הראשונים של שתי הסדרות שווה.
2. ממש את הפעולה הסטטיט Check על פי הדרישות שהוגדרו בשלב הרביעי של הפיתוח.

24. מרפאה וטורינרית של חיות מחמד מרכזת מידע על הווטרינרים העובדים במרפאה ועל חיות המחמד המטופלות בה. מספר הווטרינרים העובדים במרפאה הוא לכל היוטר 10 , ומספר חיות המחמד המטופלות במרפאה הוא לכל היוטר 500 .

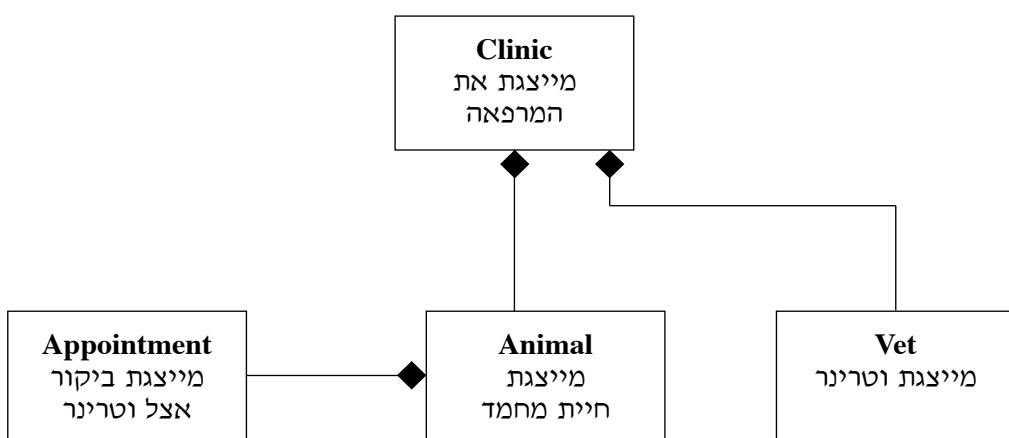
הकווים המוחים לניהול המידע הם:

- לכל וטורינר נשמר: מספר תעודת הזהות, שם, ותק בעבודה בשנים.
- לכל חיית מחמד נשמר: מספר הרישוי, שם, סוג (למשל: כלב, חתול או ארנב), גיל, פירוט של עד 50 הביקורים האחוריים במרפאה, מספר הביקורים השמורים.
- בכל ביקור במרפאה נשמר: מספר תעודת הזהות של הווטרינר שטיפול בחיות המחמד בבייקור זה, ומחזורות של תווים המכילה קודים של טיפולים שחיהת המחמד קיבלה באותו ביקור. קוד של טיפול הוא אות גדולה מתוך כל אותיות ה- ABC .

חלק מהפעולות שהמערכת יכולה לבצע הן:

- הפקט דוח של כל חיית המחמד מסווג מסוים (למשל כל הארנבים). לכל חייה הדוח יכלול את מספר הרישוי שלו, את שמה ואת גילו.
- בתחלת כל שנה: עדכון הגיל של כל חיית מחמד, קלומר הגדלת הגיל ב-1, ועדכון הוותק של כל וטורינר, קלומר הגדלת הוותק ב-1 .
- החזרת שם של וטורינר על פי מספר תעודת הזהות שלו.

לפניך תרשימים של היררכיית המחלקות הנדרשות בעבר ניהול המידע של המרפאה:



הסימן —————♦ בתרשימים מייצג הכללה.

**א.** בעבר כל מחלקה בתרשימים, הגדר את התוכנות ואת הפעולות שלה. עליך להגדיר רק את הפעולות הנחוצות כדי לענות על הדרישות שתוארו בפתחה לשאלת (שבעמוד 54) ובתרשים.

הנח נתוננות פעולות בונות, המקבלות פרמטר לכל תוכנה, ופעולות מאחזורת (פעולות Get), ואין צורך לכתוב אותן. לא נתונות פעולות קובעות (פעולות Set). בעבר כל תוכנה – רשום את ההגדרה שלה ב- C#, ורשום את התיעוד שלה. בעבר כל פעולה – רשום את הcotורת שלה ב- C#, ורשום תיעוד הכלול מה היא מקבלת ומה היא מחזירה. אין צורך למש את הפעולה. **ב.** הנח כי המערכת פותחה בהתאם לתוכנו שהציג בסעיף א, וכל הפעולות הנתונות ואלו שהגדרת ממושכות.

במחלקה **Clinic** מוסיפים את הפעולה:

```
public void AddAppointment (Animal p , string t , Vet v)
```

המקבלת חיית מוחמד k , מחרוזת t של קודים של טיפולים שקיבלה החיה בבדיקה הנוכחי ואת הווטרינר שטיפול בה. הפעולה מוסיפה את הביקור לחיה המוחמד. ממש באופן מלא את הפעולה.

הנח שמספר הביקורים הקודמים של החיה במרפאה קטן מ- 50 .

אם נוסף על הפעולות הנתונות ועל הפעולות שהגדרת בסעיף א אתה משתמש בפעולות אחרות, עליך למש אותן באופן מלא ולציין בעבר כל פעולה באיזו מחלקה יש למש אותה.

## בהצלחה!

זכות היוצרים שמורה למדינת ישראל  
אין להעתיק או לפרסם אלא ברשות משרד החינוך

**תכנות מונחה עצמים**

אם למדת מסלול זה ואתה כותב ב- **C#**, ענה על **שתיים** מהשאלות 21-24.  
(כל שאלה – 25 נקודות)

```
public class A
{
    private int myVal;
    public A (int val) {myVal = val;}
    public int F () {return 1;}
}

public class B : A
{
    private double x;
    public bool ValidCode() {return x > 8.0;}
}
```

.21 נתון:

**א.** לפני ארבעה היגדים-i-zo. קבוע לכל אחד מהם אם הוא נכון או אינו נכון, ונמק את קביעתן.

i המחלקה A ירושת את הפעולה `ValidCode` מהמחלקה B.

ii המחלקה B ירושת את כל התכונות ואת כל הפעולות של המחלקה A.

iii המחלקה B יכולה למסור שירות לתכונות של המחלקה A.

iv המחלקה A יכולה למסור לתכונה x של המחלקה B.

**ב.** ממש במחלקה B פועלה בונה שמקבלת כפרמטר מספר שלם ומספר ממשי (בסדר זה), ומאתחלת את התכונות בהתאם.

**ג.** לפני שתי הגדרות מהתכנית הראשית:

A code = new B(127 , 1.4);

A num = new A(613);

בעבור כל אחת מההוראות-i-zo שלפני, קבוע אם היא תקינה או אינה תקינה.

אם אינה תקינה, נמק את קביעטר וכותב אם זו שגיאת ריצה או שגיאת הידור (קומPILEZIA).

i `bool myBool = code.ValidCode();`

ii `bool myBool = num.ValidCode();`

iii `bool myBool = (B) code.ValidCode();`

iv `bool myBool = (B) num.ValidCode();`

**ד.** מעוניינים לדעת בזמן הריצה כמה פעמים הפעולה F הופעלת מעצם מסוג A שאינו B  
וכמה פעמים היא הופעלת מעצם מסוג B.

אם אפשר לקבל מידע זה – הוסף הוראה או הוראות שנדרשות כדי לקבל את המידע.

העתק למחברתך את המחלקה או המחלקות שבחן הוספה הוראה / הוראות.

אם אי-אפשר לקבל את המידע – הסבר מדוע.  
המשך בעמוד 36/

לפניך פרויקט ובו המחלקות A ו B .Run 22

```

public class A
{
    private int n;
    private char ch;

    public A() { n = 2; ch = 'G'; }
    public A(int n) { this .n = n; ch = 'M'; }
    public A(int n , char ch) { this.n = n; this.ch = ch; }
    public A(A other) { n = other.n; ch = other.ch; }
    public int GetN() { return n; }
    public char GetCh() { return ch; }
    public virtual void Inc() { n++; ch++; }

    public override string ToString()
    {
        string s = "";
        for (int i = 0; i < n; i++)
            s = s + ch;
        return s;
    }
}

public class B : A
{
    private A a;

    public B() : base() { a = new A(); }
    public B(int n) : base(n) { a = new A(); }
    public B(A other) : base() { a = new A(other); }
    public B(A other , int n) : base (other){ a = new A(n); }
    public override void Inc() { a.Inc(); }
}

```

```

public A MakeA()
{
    return new A(One(this.a.GetN(), this.GetN()), One(this.a.GetCh(), this.GetCh()));
}
private int One(int n, int m)
{
    if(n > m)
        return n;
    return m;
}
private char One(char ch1, char ch2)
{
    if (ch1 < ch2)
        return ch1;
    return ch2;
}
public override string ToString() { return a.ToString(); }
}

public class Run
{
    public static void Main()
    {
        A a1 = new A(4, 'E');
        A a2 = new A(3);
        B b1 = new B(a1);
        a1.Inc();
        Console.WriteLine(a1);
        Console.WriteLine(a2);
        Console.WriteLine(b1);
        B b2 = new B(b1, 1);
        Console.WriteLine(b2);
        A a3 = b2.MakeA();
        Console.WriteLine(a3);
    }
}

```

כתב מעקב אחר הפעולה Main, במחלקה Run, וכותב את הפלט.

במעקב יש לכתוב את ערכי המשתנים, ובעבור כל עצם — את ערכי התכונות שלו.

.23. לספרייה ציבורית יש אתר וירטואלי.

באתר יש 3 סוגים פריטים:

- **ספרים:** כל ספר כולל קוד פריט, שם הכותר, שם המחבר, מספר הממליצים.
- **ספר לימוד:** כל ספר לימוד כולל קוד פריט, שם הכותר, שם המחבר, האם מאושר על ידי משרד החינוך, מספר הממליצים.
- **תוכנות לומדה:** כל תוכנת לומדה כוללת קוד פריט, שם הכותר, שם המקצוע, האם מאושר על ידי משרד החינוך, מספר הממליצים.

בספרים ובספרי לימוד שם הכותר הוא שם הספר.

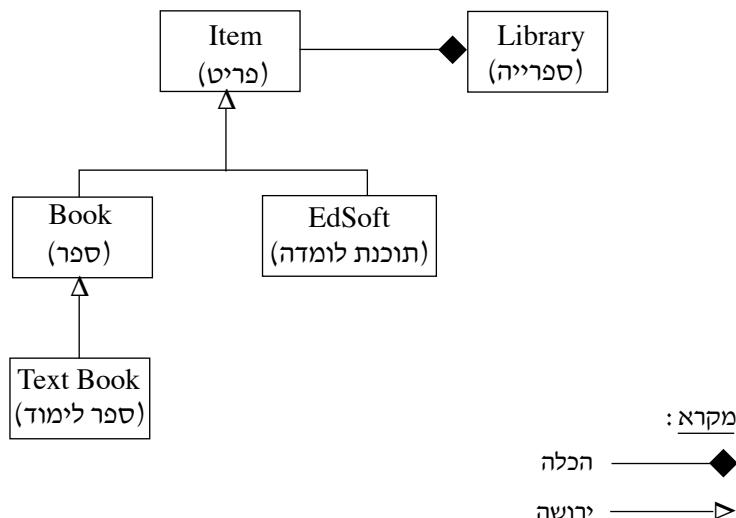
בתוכנות לומדה שם הכותר הוא שם התוכנה.

המנויים יכולים להתחבר לאתר הספרייה דרך האינטרנט.

באתר אפשר לבצע את הפעולות האלה:

- חיפוש לפי מספר הממליצים — מחזיר מערך של קודים של פריטים שיש להם מספר הממליצים המבוקש.
- קריأت ספר
- הפעלת תוכנת לומדה
- המלצה על כותר

לפניר תרשימים של היררכיות המחלקות בעברו האתר הוירטואלי של הספרייה הציבורית:



בנוסך על המחלקות המתוארות בתרשימים, נתונים 3 ממשקים:

interface IReadable { void Read(); }	קריאה // }
interface IRankable { void Rank(); }	המליצה על כותר // }
interface IApprovable { bool IsApproved(); }	האם מאושר על ידי משרד החינוך // }

**א.** העתק למחברתך את תרשימים היררכיות המחלקות, והוסף לו את הממשקים במקומות המתאים ביותר לפי העקרונות של תכניות מונחה עצמים.

השתמש בסימן זה **→ — — —** לציון מיימוש ממשק.

**ב.** לכל מחלקה כתוב:

את הכותרת שלה ב- C#, את התכונות ואת הפעולות שלה.

בעבור כל תכונה כתוב את הגדרה שלה ב- C#, ואת התיעוד שלה.

בעבור כל פעולה כתוב:

את הכותרת שלה ב- C# ותיעוד הכולל מה היא מקבלת ומה היא מחזירה. אין צורך לממש את הפעולה.

אין צורך לכתוב פעולות בונוט, פעולות קובעות (פעולות Set) ופעולות מאחזרות (פעולות Get).

.24 במרכז לאمنות מציעים שלושה סוגים סדניות: סדניות חרוזים, ציור על עץ, ציור על بد.

במרכז מתקיימות סדניות שנתיות, סדניות של 10 פגישות וסדניות חד-פעמיות.

הסדניות נערכות בבודק וברב. לכל סוג סדנה מחיר אחר. המרכז מספק את החומריים הנדרשים לכל סדנה בלי תשלום נוספים.

הסדניות מוגבלות למספר מקסימלי של משתתפים.

משתתף יכול להירשם ליותר מסדנה אחת.

במערכת ניהול הסדניות נשמרים לכל סדנה הפרטים האלה: קוד הסדנה, שם הסדנה, סוג הסדנה (חרוזים, ציור על עץ, ציור על بد), משך הסדנה (שנתית, 10 פגישות, חד-פעמית), מועד הסדנה (בוקר, ערב), מחיר הסדנה למשתתף, מספר מקסימלי של משתתפים, ורישמת המשתתפים בסדנה.

לכל משתתף נשמרים שמו ומספר הטלפון שלו.

במערכת ניהול הסדניות אפשר בין היתר לבצע את הפעולות האלה:

(1) לרשום משתתף לסדנה.

(2) לבדוק אם יש מקום בסדנה.

(3) לחשב את סך כל ההכנסות מסדנה אחת.

**א.** סרטוט את היררכיות המחלקות בעברו ניהול הסדניות. בסרטוט כתוב לכל מחלקה את שם המחלקה, התכונות שלה, הכותרות של הפעולות, ותיעוד של התכונות והפעולות. סמן בסרטוט את הקשרים בין המחלקות.

הנח כי לכל תכונה הוגדרו פעולות Set/Get, ולכל מחלקה הוגדרה פעולה בונה. אין צורך לרשום פעולות אלה.

**ב.** בשל עליית המחירים במשק הוחלט כי בחילק מהסדניות לא יחולקו החומריים חינם. המשתתפים יידרשו לשלם עבור החומריים, על פי רישימת חומריים אשר תפורט בעת הרישום לסדנה. לכל חומר נשמר שמו ומהירות שלו בעבר הנסיבות הנדרשות לסדנה.

מחיר הסדנה יכולות מחולאות מהבסיס של הסדנה ואת עלות החומריים הנדרשים לסדנה.

(1) הוסף מחלקות לסרטוט שרטוט בסעיף א' (בלי להוסיף או לగורע תכונות ופעולות במחלקות שהוגדרו בסעיף א'), כך שתיארים למערכת ניהול הסדניות לאחר השינוי,

באופן המתאים ביותר לעקרונות של תכונות מונחה עצמים.

לכל מחלקה שהוספה כתוב ב- C את התכונות שלה, הפעולות שלה ותיעוד התכונות והפעולות.

אין צורך לכתוב פעולות בונות, פעולות קובעת (Set), ופעולות מאחזורת (Get).

(2) ממש ב- # C את הפעולות שכתבת בסעיף (1).

## בצלחה!

תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב ב- C#, ענה על שתיים מהשאלות 21-24.  
(כל שאלה – 25 נקודות)

- .21. א. לפניך חלק מהממשק של המחלקה Node, ובו ארבע פעולות.

Node(T x)	הפעולה בונה חוליה. הערך של החוליה הוא x, ואין לה חוליה עוקבת.
Node(T x, Node<T> next)	הפעולה בונה חוליה. הערך של החוליה הוא x, והחוליה העוקבת לה היא next. ערךו של next יכול להיות null.
Node<T> GetNext()	הפעולה מחזירה את החוליה העוקבת. אם אין חוליה עוקבת, הפעולה מחזירה null.
string ToString()	הפעולה מחזירה מחרוזת המתארת את החוליה.

- (1) מהו המנגנון המאפשר שתי פעולהות ששם זהה באוטה מחלקה?  
(2) האם אפשר להוסיף לממשק המחלקה Node פעולה בונה שלישית?  
אם כן – כתוב כותרת מתאימה לפעולה, אם לא – הסבר מדוע.

(שים לב: סעיף ב בעמוד הבא.)

. Node<T> הירושות את המחלקה Node2<T> , Node1<T> נתונות המחלקות .

```
public class Node1<T> : Node<T>
{
    public Node1(T x) : base(x) { }
    public Node1(T x, Node<T> n):base(x,n) { }
    public override string ToString() { return "$" + base.ToString() + "$"; }
}

public class Node2<T> : Node<T>
{
    public Node2(T x) : base(x) { }
    public Node2(T x, Node<T> n):base(x,n) { }
    public override string ToString() { return "#" + base.ToString() + "#"; }
}
```

נתונה הפעולה Main במחלקה . Test

בפעולה Main חסרים חלקו הוראות בשורות המסומנות (v)-(i).

העתק שורות אלה למחברתך והשלם אותן, כך שפלט הפעולה Main יהיה (משמאל לימין)

1\$2\$#3#\$4\$5

```
public class Test
{
    public static void Main ()
    {
        Node<int> nd = _____ (i) _____ ;
        nd = _____ (ii) _____ ;
        nd = _____ (iii) _____ ;
        nd = _____ (iv) _____ ;
        nd = _____ (v) _____ ;
        while (nd != null)
        {
            Console.WriteLine(nd);
            nd = nd.GetNext();
        }
    }
}
```

.22. בשאלת זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

**א. לפניך הגדרות:**

```
interface IOne { ... }
interface ITwo { ... }
interface IThree : IOne { ... }
class C1 : IOne { ... }
class C2 : ITwo { ... }
class C3 : IThree { ... }
class C4 : C3 , ITwo { ... }
```

לפניך שМОונה קטעי תכנית, i-viii . קבע אם כל אחד מהם תקין או אינו תקין.  
אם אינו תקין — נמק את קביעתך, וקבע אם יש שגיאת הידור (קומפלציה) או שגיאת ריצה.

- i     ITwo a = new ITwo();
- ii    ITwo b = new C2();
- iii   C3 c = new C4();
- iv    C2 d = new C4();
- v     C4 e = new C3();
- vi    C4 f = (C4)(new C3());
- vii   IOne g1 = new C1();
- C4 g2 = new C4();  
g1 = g2;
- viii IOne h1 = new C4();  
ITwo h2 = new C2();  
h2 = h1;

(שים לב: סעיף ב בעמוד הבא.)

.ב. אין קשר לסעיף א.)

נתונות שתי מחלקות A, B, ונתונה הפעולה Main במחלקה Test.

```
public class A
{
    public A(int k) { Console.WriteLine(k + " "); }

public class B : A
{
    public B(int k)
    {
        A a = new A(10);
        Console.WriteLine(k + " ");
    }
}

public class Test
{
    public static void Main() {   A a = new B(6);   }
}
```

המחלקה B אינה עוברת קומPILEציה.

i הסבר מהי השגיאה.

ii תקן את השגיאה שהסבורה בתת-סעיף i כך שפלט הפעולה Main יהיה (משמאל לימין) 6, מבלתי להוציא פקודות הדפסה או לשנות פקודות הדפסה קיימות, ומבלתי לשנות את המחלקה Test.

iii העתק למחברתך את המחלקה / המחלקות שתיקנת.

iv תקן את השגיאה שהסבורה בתת-סעיף i כך שפלט הפעולה Main יהיה 6 10 6, מבלתי להוציא פקודות הדפסה או לשנות פקודות הדפסה קיימות, ומבלתי לשנות את המחלקה Test.

v העתק למחברתך את המחלקה / המחלקות שתיקנת.

. בשאלת זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

א. לפניך פרויקט שבו המחלקות AAA ו Run BBB , .

```
public class AAA
{
    protected int x;
    public AAA(int k) { x = k; }
    public override string ToString() { return "H"; }
}

public class BBB: AAA
{
    public BBB(int k): base(k) { }
    public override string ToString() { return "T"; }
    public AAA ConvertToAAA() { return new AAA(x); }
}

public class Run
{
    public static void Main()
    {
        AAA s1 = new AAA(10);
        AAA s2 = new BBB(20);
        BBB s3 = new BBB(30);
        Console.WriteLine(s1 + " ");
        Console.WriteLine(s2 + " ");
        Console.WriteLine(s3 + " ");
        Console.WriteLine((AAA)s3 + " ");
        Console.WriteLine(s3.ConvertToAAA());
    }
}
```

כתוב מעקב אחר הפעולה Main במחלקה Run , וכותוב את הפלט.  
במעקב יש לכתוב את ערכי המשתנים, ובabboר כל עצם — את ערכי התכונות שלו.

(שים לב: סעיף ב בעמוד הבא.)

(ב.) אין קשר לסעיף א).

לפניך פרויקט ובו המחלקות A ו B . Test

public class A

{

```
    public int x;
    public A() : this(1)      { }
    public A(int k)          { this.x = k*10; }
    public virtual int M()   { return x; }
```

}

public class B : A

{

```
    public int x;
    public B () : this(2)     { }
    public B(int k) : base(k + 1) { this.x = base.x + 1; }
    public override int M()  { return x; }
```

}

public class Test

{

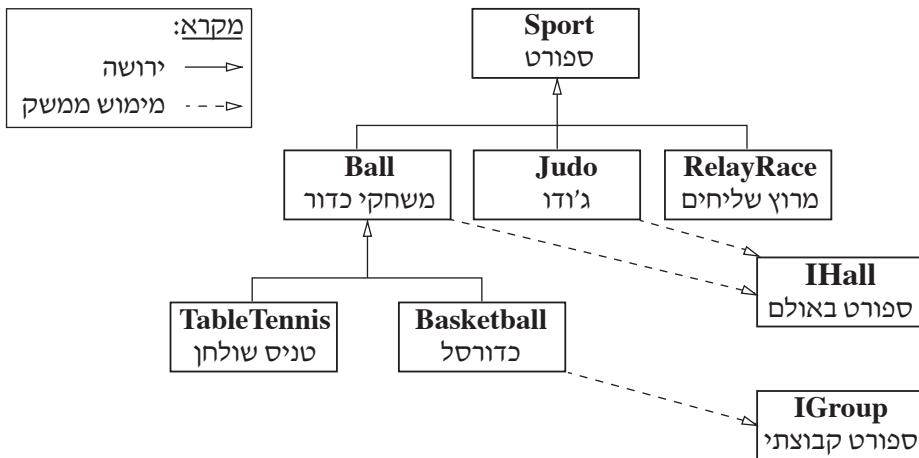
```
    public static void Main()
    {
        A a = new A();
        Console.WriteLine(a.x);
        Console.WriteLine(a.M());
        B b = new B();
        Console.WriteLine(b.x);
        Console.WriteLine(b.M());
        a = new B();
        Console.WriteLine(a.x);
        Console.WriteLine(a.M());
        b = (B)a;
        Console.WriteLine(b.x);
        Console.WriteLine(b.M());
    }
}
```

}

כתב מעקב אחר הפעולה Main במחלקה Test , וכתוב את הפלט.

במעקב יש לכתוב את ערכי המשתנים, ובעבור כל עצם — את ערכי התכונות שלו.

לפניך תרשימים של היררכיות מחלקות בעבור המחלקות **Sport** – ספורט, **Ball** – משחק כי כדור, **TableTennis** – טניס שולחן, **BasketBall** – כדורסל, **Judo** – ג'ודו, **RelayRace** – מרוץ שליחים, **IHall** – ספורט באולם ו- **IGroup** – ספורט קבוצתי.



א. (1) לפניך ארבע הגדרות, i-iv. קבע אם כל אחת מהן נכונה או אינה נכונה.

אם אינה נכונה, נמק את קביעתך.

- Sport s1 = new Ball();
- Sport s2 = new Basketball();
- Judo b2 = new Sport();
- Basketball b3 = new Sport();

לפניך שני היגדים, i-ii. קבע אם כל אחד מהן נכון או לא נכון. נמק את קביעתך.

- אפשר להגדיר במחלקה **Ball** משתנים מסווג private שאינם מוגדרים במחלקה **Sport** או במחלקה **Judo**.
- אם במחלקה **Sport** יש הגדרה של משתנה בשם title, בהכרח המחלקות **Judo** ו- **Ball** יורשוות אותו, אך אין יכולות לgesht אליו יישירות.

ב. כתובコード ב- C# לכל מחלקה ולכל ממשק שבתרשים.

כתב ב- C# פועלה שחתימתה public static int Count(Sport[] sp), שתחזיר את מספר העצמים במערך sp הממשים את שני הממשקים.

## בצלחה!

זכות היוצרים שמורה למדיינית ישראל  
אין להעתיק או לפרסם אלא ברשות משרד החינוך

**תכנות מונחה עצמים**

אם למדת מסלול זה ואתה כותב ב- C#, ענה על שתיים מהשאלות 21-24, (כל שאלה – 25 נקודות)

.21      בשאלת זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שנייהם.

א.      לפניך שתי מחלקות, A, B :

```
public class A
{
    public A() {
        Console.WriteLine( "A" );
    }
    public virtual void One() {
        Console.WriteLine( "One of A" );
    }
    public virtual void Two() {
        One();
    }
}
public class B : A
{
    public B() {
        Console.WriteLine( "B" );
    }
    public override void One() {
        Console.WriteLine( "One of B" );
    }
    public override void Two() {
        Console.WriteLine( "Two of B" );
    }
    public void Three() {
        base.Two();
    }
}
```

(שים לב: המשך סעיף א בעמוד הבא.)

לפניך שבעה קטיעי תכנית, i-vii. לכל אחד מהם קבוע אם הוא תקין או אינו תקין.

אם הקטע תקין — כתוב מה הפלט שיתקבל מהרצת קטע התכנית.

אם הקטע אינו תקין — נמק את קביעתך, וכותב אם יש שגיאת הידור (קומפילציה) או שגיאת זמן ריצה.

- i      A a1 = new A();  
a1 . Two();  
( (A) a1 ) . Two();
- ii     A a2 = new A();  
( (B) a2 ) . Two();
- iii    A a3 = new B();  
a3 . Two();  
( (B) a3 ) . Three();
- iv    B b1 = new A();  
b1 . Two();
- v    B b2 = new B();  
b2 . Two();  
( (A) b2 ) . Two();
- vi   B b3 = new B();  
( (A) b3 ) . Three();
- vii   B b5 = new B();  
( (B) b5 ) . Three();

(שים לב: סעיף ב של השאלה בעמוד הבא.)

/המשך בעמוד 48/

.ב. (אין קשר לסעיף א.)

לפניך חתימה של המחלקות AAA, BBB וכוכנותיהן:

```
public class AAA
{
    private int n ;
    private double x ;
}

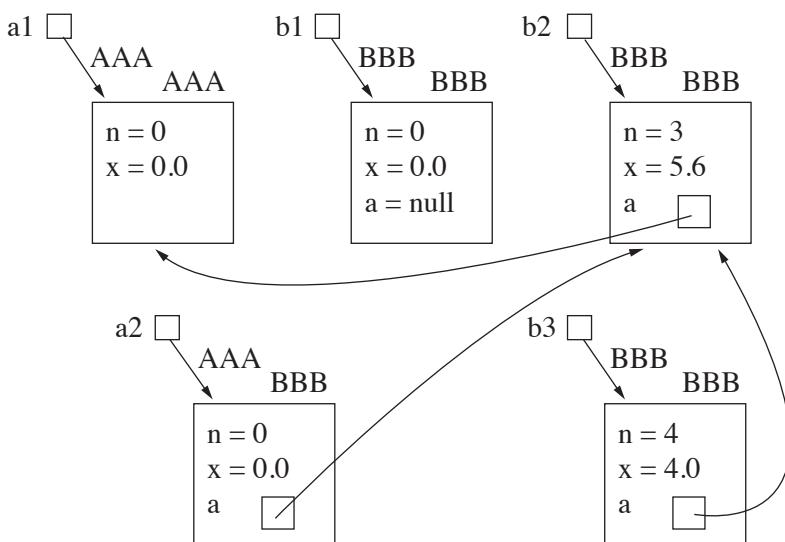
public class BBB : AAA
```

```
{    private AAA a;
```

לפניך המחלקה Test, ובה פעולה ראשית:

```
public class Test
{
    public static void Main()
    {
        AAA a1 = new AAA();
        BBB b1 = new BBB();
        BBB b2 = new BBB(3, 5.6, a1);
        AAA a2 = new BBB(b2);
        BBB b3 = new BBB(4, b2);
    }
}
```

הרצת הפעולה הראשית Main יוצרה את העצמים הבאים:



ממש ב- C# במחלקות BBB, AAA את הפעולות הבוניות הנדרשות להרצת הפעולה Main כך שיתקבלו העצמים המתוארים.  
המשך בעמוד 49/

.22. בשאלת זו שני סעיפים, א-ב, שאין קשר ביניהם. ענה על שניהם.

א. לפניך המחלקות C, D, UseCD. רשום את הפלט של הפעולה הראשית.

```

public class C {
    public C() {
        Console.WriteLine("C");
    }
    public C(int n) {
        for (int i = 0; i < n; i++)
            new C();
        Console.WriteLine("CC");
    }
}
public class D : C {
    public D() : base() {
        Console.WriteLine("D");
    }
    public D(int n) : base(n) {
        for (int i = 0; i < n; i++)
            new D();
    }
}
public class UseCD {
    public static void Main() {
        Console.WriteLine("****");
        C c1 = new C(3);
        Console.WriteLine("---");
        D d1 = new D(1);
        Console.WriteLine("###");
        D d2 = new D(4);
    }
}

```

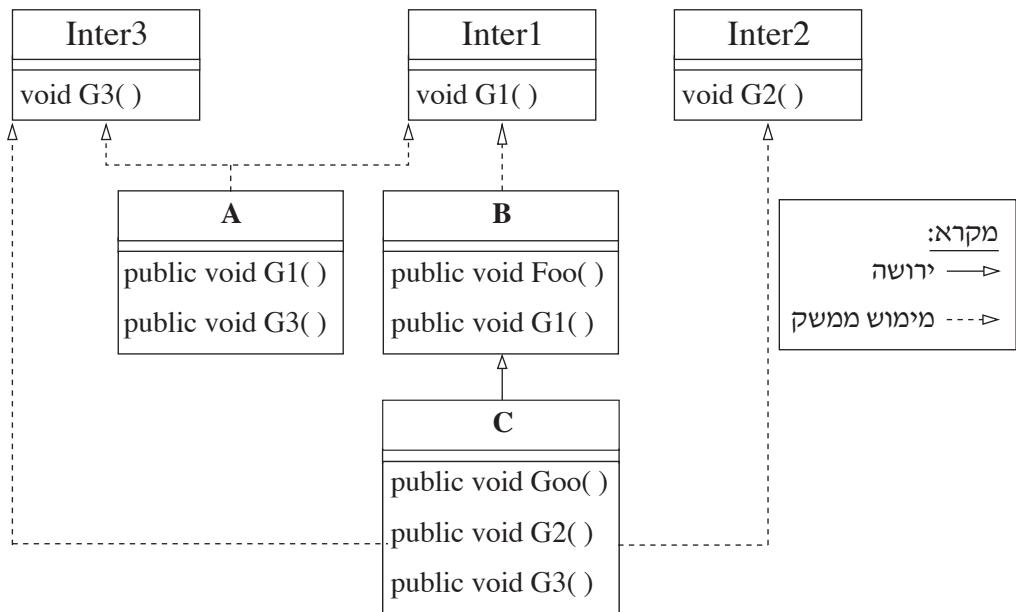
**ב.** (אין קשר לסעיף א.).

לפניך המחלקות AA, BB, Run. Run היא ראשית במחלקה Run. עקוב אחר הפעולה הראשית, ורשות את הפולט המתקבל. במקביל יש לכתוב את ערכי המשתנים, את ערכי תכונות המחלקה (תכונות סטטיות), ובמעבר כל עצם יש לכתוב את ערכי התכונות שלו.

```
public class AA
{
    private int num1 = 0;
    private int num2 = 0;
    private static int count = 0;
    public AA(int num)
    {
        num1 = num;
        num2 = num;
        count++;
        Console.WriteLine("AA ctor1");
    }
    public AA(int num1, int num2)
    {
        this.num1 = num1;
        this.num2 = num2;
        count++;
        Console.WriteLine("AA ctor2");
    }
    public virtual int Sum()
    {
        return num1 + num2;
    }
    public static int GetCount()
    {
        return count;
    }
}
```

```
public class BB : AA
{
    private int num3 = 0;
    public BB(int num)
        :base (num)
    {
        num3 = num;
        Console.WriteLine("BB ctor1");
    }
    public BB (int num1, int num2, int num3)
        :base(num1, num2)
    {
        this.num3 = num3;
        Console.WriteLine("BB ctor2");
    }
    public override int Sum()
    {
        return base.Sum() + num3;
    }
    public void SetNum3(int num)
    {
        num3 = num;
    }
}
```

```
public class Run
{
    public static void Main()
    {
        AA f1 = new AA(10);
        Console.WriteLine("sum1 = " + f1.Sum());
        Console.WriteLine("count = " + AA.GetCount());
        AA f2 = new AA(10, 20);
        Console.WriteLine("count = " + AA.GetCount());
        BB s1 = new BB(1);
        AA f3 = new BB(2);
        Console.WriteLine("count = " + AA.GetCount());
        f2 = s1;
        Console.WriteLine("sum = " + f2.Sum());
        s1.SetNum3(2);
        Console.WriteLine("sum = " + s1.Sum());
        Console.WriteLine("sum = " + f2.Sum());
    }
}
```



**א.** השלים את שלוש השורות החסרות כך שקטוע התכנית שלפניך יעבד בלי שגיאות הידור (קומPILEציה) ו/או זמן ריצה.  
העתק למחברתך את השורות שהשלמות.

```

_____ x = new A();
_____ y = new B();
_____ z = new C();

x = y;
y = new C();
y.Foo();
z.G3();
((C) z).Goo();
z = new A();
  
```

**ב.** כתוב ב- C# פעליה שתתקבל מערך חד-ממדי מטיפוס .Inter1

הפעולה תעבור על איברי המערך, ותקרו לפעליה () G3() אם היא קיימת בעבור העצם. אם הפעולה () G3() אינה קיימת – הפעולה שכתבה תקרה לפעליה () G1().

. type3 , type2 , type1 .<sup>24</sup> חברת מסוימת מייצרת שלושה סוגי טלפונים סלולריים:

טלפון מסוג type1 אפשר לבצע את הפעולות האלה:

—	חיוֹג —	Dial
—	קבלה שיחה —	RecCall
—	שליחת הודעה —	SendMsg
—	קבלת הודעה —	RecMsg

טלפון מסוג type2 אפשר לבצע את הפעולות האלה:

—	חיוֹג —	Dial
—	קבלה שיחה —	RecCall
—	שליחת הודעה —	SendMsg
—	קבלת הודעה —	RecMsg
—	צילום תמונה —	TakePic
—	תצוגת תמונה —	ShowPic

טלפון מסוג type3 אפשר לבצע את הפעולות האלה:

—	חיוֹג —	Dial
—	קבלה שיחה —	RecCall
—	צילום תמונה —	TakePic
—	תצוגת תמונה —	ShowPic
—	התקנת אפליקציות —	InstallApp

בכל אחד מסוגי הטלפונים:

נשמר מספר דקוטה השיחה שנעשו מן הטלפון.

אפשר לAPS את מספר דקוטה השיחה — . Reset

אפשר להחזיר חשבון טלפון עדכני — Bill , המוחשב כר: מספר דקוטה השיחה × 1 ש"ח.

טלפונים שיש בהם אפשרות שלוחה הודעות, החזרת חשבון הטלפון העדכני היא:

מספר דקוטה השיחה × 1 ש"ח + מספר ההודעות שנשלחו × 0.5 ש"ח.

טלפונים האלה פועלות האיפוס מאפסת את מספר דקוטה השיחה וכן את מספר ההודעות שנשלחו.

(שים לב: המשך השאלה בעמוד הבא.)

.א.

בעבור הטלפונים הסולריים המתוארים לעיל, סרטט את היררכיות המחלקות והמשקדים, באופן המתאים ביותר לעקרונות של תכונות מונחה עצמים (הכמה inheritance – הורשה encapsulation – סגירתם, polymorphism – משתנים). סמן בסרטוט את הקשרים בין המחלקות, ואת הקשרים בין המחלקות ובין המשקדים.

.ב.

אין צורך לכלול בסרטוט תכונות ופעולות. לכל אחת מן המחלקות שבסרטוט כתוב ב- C# את כותרת המחלקה ואת התכונות שלה, וכותרות לכל הפעולות.

.ג.

כל אחד מן המשקדים שבסרטוט כתוב ב- C# את הכותרת שלו, וכותרות לכל הפעולות. אין צורך לכתוב את הפעולות Get ו- Set ופעולות בונות. ממש ב- C# את הפעולה המחשבה את החשבון העדכני בכל אחת מן המחלקות שהיא מופיעה בהן.

## בצלחה!

זכות היוצרים שורזה למדינת ישראל  
אין להעתיק או לפרסם אלא ברשות משרד החינוך

תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב ב- #C, ענה על שתיים מהשאלות 21-24.

(כל שאלת — 25 נקודות)

.21. בשאלת זו שלושה סעיפים, א-ג, שאין קשר ביניהם. ענה על שלושתם.

א. לפניך קטע קוד בפעולה ראשית במחלקה Ran:

C c = new A();

B b1 = (B) (new A());

B b2 = new D();

A a = new D();

(1) הנה כי A, B, C, D הן מחלקות ונתון כי לכל אחת מהן מחלקה יורשת אמת לכל היותר.

כתב שתי אפשרויות להיררכיות המחלקות כך שקטע הקוד יהיה תקין.

(2) נתון גם כי הפעולה () F מוגדרת במחלקה B בלבד, כמו כן נתון כי בתכנית הראשית נכללות גם ההוראות האלה:

c.F();

b1.F();

קבע באיזו מבחן שתי האפשרויות להיררכיות המחלקות שכותבת בתת-סעיף א (1) ההוראות האלה תקיןות.

העתק את היררכיות המחלקות הזו, והסביר מדוע בהיררכיה האחורה ההוראות אין תקיןות.

ב. לפניך קטע תוכנית בפעולה ראשית במחלקה Program:

CC c = new AA();

BB b1 = (BB) (new AA(2));

קבע מי מבחן AA, BB, CC לא יכול להיות ממשק. הסבר מדוע.

(שים לב: סעיף ג של השאלה בעמוד הבא.)

/המשך בעמוד 42/

לפניך המחלקות A, B והמחלקה Run ובה הפעולה הראשית Main. רשום את הפלט של הפעולה הראשית.

```
public class A
{
    public A()
    {
        F();
    }
    public virtual void F()
    {
        Console.WriteLine("Class A");
    }
}

public class B : A
{
    public B(): base() { }
    public override void F()
    {
        Console.WriteLine("Class B");
    }
}

public Class Run
{
    public static void Main()
    {
        A a = new B();
    }
}
```

לפניך המחלקות Run , CC , BB , AA והמחלקה Run ובה פעולה ראשית.

```
public class AA
{
    protected int i;
    public AA(int i)
    {
        this.i = i;
    }
}
```

```
public class BB : AA
{
    public BB(int i) : base(i + 1) { }
    public virtual bool What(Object other)
    {
        return ((other != null) &&
                (other is BB) &&
                (this.i == ((BB)other).i));
    }
    public virtual bool What(BB other, int k)
    {
        return ((other != null) &&
                (this.i - k == ((BB)other).i));
    }
}
```

```
public class CC : BB
{
    public CC(int i) : base(i) { }
```

```
public class DD : BB
{
    public DD(int i) : base(i + 1) { }
    public override bool What(BB other, int k)
    {
        return base.What(other, 1);
    }
}
```

```
public class Run
{
    public static void Main()
    {
        AA a = new AA(1);
        BB b = new BB(1);
        CC c = new CC(1);
        DD d = new DD(1);
        BB b1 = new DD(1);
        AA c1 = new CC(1);
        //***
    }
}
```

(שים לב: המשך השאלה בעמוד הבא).

- א. סרטט את היררכיית המחלקות AA, BB, CC, DD.
- ב. עקוב אחר הפעולה הראשית.
- במיעקב יש לכתוב את ערכי המשתנים, ובעבור כל עצם יש לכתוב את ערכי התכונות שלו.
- ג. לפניך הוראות 1-iii.

i       $a = c;$   
 ii      $b = a;$   
 iii     $c = ((CC) b1);$

בעבור כל אחת מן ההוראות 1-iii בצע:

- כתוב את ההוראה במקומות \*\*\*// בפעולה הראשית של המחלקה Run.
- קבע אם ההוראה תקינה או אינה תקינה.

אם ההוראה תקינה — הצג את השינויים שנוצרו בעצמים בעקבות ההוראה.

אם ההוראה אינה תקינה — נמק את קביעתך, וכותב אם השגיאה היא שגיאת הידור (קומפילציה) או שגיאת זמן ויצחה.

ד. לפניך הוראות 1-iv.

- i      Console.WriteLine(b1.What(b));  
 ii     Console.WriteLine(b1.What(b, 1));  
 iii    Console.WriteLine(((CC) c1).What(c));  
 iv    Console.WriteLine(d.What(a));

בעבור כל אחת מן ההוראות 1-iv. בצע:

- כתוב את ההוראה במקומות \*\*\*// בפעולה הראשית של המחלקה Run.
- כתוב מה יהיה הפלט בעקבות כל אחת מן ההוראות.

- .23. לפניך הממשק IThing, המחלקות A , B , C , D , ופולה ראשית במחלקה Run .  
 א. עקב אחר הפעולה הראשית, וכותב את הפלט המתkeletal.

במשך יש לכתוב את ערכי המשתנים, ובמעבר כל עצם את ערכי התוכנות שלו.

```
public interface IThing
```

```
{
```

```
    int Value();
```

```
}
```

```
public class A : IThing
```

```
{
```

```
    private int a;
```

```
    public A(int a)
```

```
{
```

```
        this.a = a;
```

```
}
```

```
    public int Value()
```

```
{
```

```
        return this.a;
```

```
}
```

```
}
```

```
public class B : IThing
```

```
{
```

```
    protected IThing a , b;
```

```
    public B(IThing a , IThing b)
```

```
{
```

```
        this.a = a;
```

```
        this.b = b;
```

```
}
```

```
    public virtual int Value()
```

```
{
```

```
        return this.a.Value() + this.b.Value();
```

```
}
```

```
}
```

(שים לב: המשך הסעיף בעמוד הבא.)

/המשך בעמוד 46/

```
public class C : B
{
    public C(IThing a, IThing b) : base(a, b) { }
    public override int Value()
    {
        return this.a.Value() * this.b.Value();
    }
}
```

```
public class D : IThing
{
    private IThing[] things;
    private int limit;
    public D (IThing[] things, int limit)
    {
        this.things = things;
        this.limit = limit;
    }
    public int Value()
    {
        int val = 0;
        for (int i = 0; i < limit; i++)
        {
            val += this.things[i].Value();
        }
        return val;
    }
}
```

(שים לב: המשך הסעיף בעמוד הבא.)

```

public class Run
{
    public static void Main()
    {
        IThing[] things = new IThing[5];
        things[0] = new A(2);
        things[1] = new B(things[0] , things[0]);
        things[2] = new C(things[0] , things[1]);
        things[3] = things[2];
        things[4] = new D(things , 4);      // (*) 
        for(int i = 0; i < things.Length; i++)
        {
            Console.WriteLine(things[i].Value());
        }
    }
}

```

ב. הסבר בקצרה מה תהיה ההשפעה על ריצת התוכנית אם בשורה המסומנת ב- (\*) יחליפו את הוראה הקיימת בהוראה שלפניך:

things[4] = new D(things , 5);

24.

החברה "צעועים זה הם" מעוניינת למחשב את מאגר הצעועים במחסן המרכזי שלו.

כדי לעשות זאת הוגדרו שתי מחלקות בעבר שני סוגי צעועים:

מחלקה **Doll** בעבר בובות, והמחלקה **Car** בעבר מכוניות צעצוע.  
במחלקה **Doll** חמש תכונות ושתי פעולות.

התכונות: שם הבובה, מחיר בסיס של הבובה, צבע שער, מספר האביזרים המצורפים, מחיר לאביזר  
(לכל האביזרים מחיר זהה).

הפעולות:

(i) החזרת מחיר הבובה לצרכן. מחיר הבובה לצרכן הוא מחיר הבסיס שלה, בתוספת מחיר  
האביזרים המצורפים אליה (המוחשב על פי מספר האביזרים כפול המחיר לאביזר).

(ii) עדכון מחיר הבסיס של בובה על ידי הגדלו ב- ק אחוזים. ק הוא מספר ממשי המתkeletal  
כפרמטר.

במחלקה **Car** ארבע תכונות ושתי פעולות.

התכונות: שם מכונית הצעצוע, מחיר הבסיס של המכונית, סיווג גודל המכונית, צבע המכונית.

התכוונה "סיווג גודל המכונית" מיוצגת עליידי מספר שלם: 0 – מכונית קטנה;  
1 – מכונית בינונית; 2 – מכונית גדולה.

הפעולות:

(i) החזרת מחיר המכונית לצרכן, כויהר המכונית לצרכן נקבע על פי גודלה.  
מחיר לצרכן של מכונית קטנה הוא מחיר הבסיס,

מחיר לצרכן של מכונית בינונית הוא מחיר בסיס + 15 ש"ח,

וממחיר לצרכן של מכונית גדולה הוא מחיר הבסיס + 30 ש"ח.

(ii) עדכון מחיר הבסיס של מכונית על ידי הגדלו ב- ק אחוזים. ק הוא מספר ממשי  
המתkeletal כפרמטר.

(שים לב: המשך השאלה בעמוד הבא.)

לפניך תרשימים המחלקות של מאגר הצעצועים הממוחשב:

<b>Doll</b>
— string name
— double basePrice
— string color
— int accNums
— double accPrice
+ double Price()
+ void UpdatePrice(double percent)
+ Doll(string name , double basePrice , string color , int accNum , double accPrice)

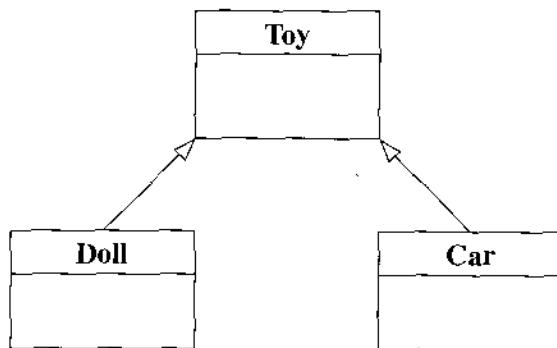
<b>Car</b>
— string name
— double basePrice
— string color
— int size
+ double Price()
+ void UpdatePrice(double percent)
+ Car (string name , double basePrice , string color , int size)

מקרה:	
private	— מציין
public	+ מציין

(שים לב: סעיפים השאלה בעמוד הבא.)

/המשך בעמוד 50/

א. הוגדרה מחלקה חדשה **Toy** שהמחלקות **Doll** ו- **Car** יירושות ממנה.  
 לפניך שلد של תרשימים מחלקות בעבור המחלקות **Toy**, **Doll**, **Toy** ו- **Car**. העתק את התרשימים למחברתך וכותב בו לכל מחלקה את התכונות ואת כוורות הפעולות, באופן המתאים ביותר לעקרונות תכנון מונחה עצמים (הכמסה – encapsulation –, הורשה – inheritance –, פולימורפיזם – polymorphism –).  
 כתוב תיעוד לכל תכונה.



- בסעיפים ב-ג שלפניו התייחס לתרשימים המחלקות שכתבת בסעיף א.  
**ב.** ממש ב-# C את הפעולה **Price**, בכל המחלקות שהיא מופיעה בהן.  
**ג.** כתוב ב-# C פעולה ראשית שתבצע את שתי המשימות (1) ו- (2) שלפניו.  
 (1) יצירת עצם אחד מטיפוס **Doll**, ועצם אחד מטיפוס **Car** (עם ערכים כרצונך).  
 (2) חישוב והדפסה של סכום המחיריהם לצרךן של שני הצעדים (בובה ומכונית).

## בהצלחה!

**תבניות מונחה עצמים**

אם למדת מסלול זה ואתה כותב ב- #C, ענה על אחת מהשאלות 15-16. (25 נקודות)

15. בשאלת זו שני סעיפים א-ב שאין קשר ביניהם. ענה על שניהם.

א. נתנות חמיש מחלקות Run, Z, Y, X, B. התח סעיפים (1)-(4) שלפניך מתייחסים למחלקות אלה. התת-סעיפים אינם קשורים זה לזה. ענה על כלם.

(1) בכל אחת מהמחלקות B, X, Y, Z הוגדרה הפעולה { } Foo()

ובמחלקה Run הוגדרה הפעולה

האם תתקבל שגיאת הידור (קומPILEציה)? אם כן – הסבר מדוע.

(2) הנה שהמחלקות X, Y, Z יירושות את המחלקה B.

בכל אחת מהמחלקות X, Y, Z הוגדרה הפעולה { } Foo()

ובמחלקה Run הוגדרה הפעולה

האם תתקבל שגיאת הידור (קומPILEציה)? אם כן – הסבר מדוע.

(3) הנה שהמחלקות X, Y, Z יירושות את המחלקה B.

בכל אחת מהמחלקות B, X, Y, Z הוגדרה הפעולה { } Foo()

ובמחלקה Run הוגדרה הפעולה

האם תתקבל שגיאת הידור (קומPILEציה)? אם כן – הסבר מדוע.

(4) הנה שהמחלקות X, Y, Z יירושות את המחלקה B.

בכל אחת מהמחלקות B, X, Y, Z הוגדרה הפעולה

public virtual void Foo() {}

ובמחלקה Run הוגדרה הפעולה

האם תתקבל שגיאת הידור (קומPILEציה)? אם כן – הסבר מדוע.

**(שים לב: סעיף ב של השאלה בעמוד הבא.)**

ב. (אין קשר לשיעין א.)

לפניך המחלקה **Singer** שיורשת את המחלקה **Artist** והממשק **IPerform** שהמחלקה

**Singer** מממשת.

interface IPerform

{ void Act();

int Train();

} public class Singer : Artist , IPerform

{ private int hits;

public Singer(string name, double sal) : base(sal, name)

{

    this.hits = 5;

}

public Singer(double sal, int hits) : base(sal, "Singer Name")

{

    this.hits = hits;

}

public Singer(int hits) : base(6532.6, "Some", "One")

{

    this.hits = hits;

}

public double Value() { return this.hits \* this.Price(); }

public int GetNum() { return Artist.num; }

public override double Calc(double d) { return d \* base.Calc(10.2); }

public override void Print() { base.Print(); Console.WriteLine("Singer"); }

public void Act() { Console.WriteLine("I am singing"); }

}

כתב ב- C# את כוורת המחלקה **Artist**, את התכונות ואת הפעולות המתיחסות מהמחלקה

**Artist** והמשק **IPerform** הנתונים. אין צורך למש את הפעולות של המחלקה **Singer**

אין לשנות את המחלקה **Singer** ואת המשק

לפניך המחלקות A, B, C. 16

שלשות הסעיפים א-ג מתייחסים למחלקות האלה אך איןם קשורים זה לזה. ענה על שלושתם.

```

public class A {
    private int x;
    public A() { this.x = 0; }
    public A(int x) { this.x = x; }
    public virtual int GetX() { return x; }
    public void DoubleX() { this.x = 2 * GetX(); }
    public virtual void TripleX() { this.x = 3 * GetX(); }
    public virtual void Sub() { this.x = x - 1; }
    public void Calc() { Sub(); }
    public override string ToString() { return "xA=" + this.x; }
}

public class B:A {
    private int x;
    public B() : base() { this.x = 1; }
    public B(int x) : base(x) { this.x = -x; }
    public B(int xA, int xB) : base(xA) { this.x = xB; }
    public override int GetX() { return x; }
    public int BaseX() { return base.GetX(); }
    public void TenTimesX() { this.x = 10 * GetX(); }
    public override void TripleX() { this.x = 3 * GetX(); }
    public override void Sub() { this.x = x - 2; }
    public override string ToString() { return base.ToString() + "xB=" + this.x + ";" }
}

```

(שים לב: סעיפי השאלה בעמודים הבאים.)

/המשר בעמוד 30/

**א.** לפניך סדרת הוראות שהפלט עבורה צריך להיות:  
 $xA=1$   $xB=20$ ;  $xA=1$   $xB=20$ ;  $xA=1$   $xB=20$ ;  $xA=1$   $xB=20$ ;

בסדרת הhorאות נפלת שגיאת היור (קומפיילציה). תכן את השגיאה כדי שיתקבל הפלט הנכון.

```
A a1 = new B(1, 20);
Object obj = a1;
B b1 = a1;
A a2 = a1;
Console.WriteLine(a1+" "+obj+" "+a2+" "+b1);
```

**ב.** לפניך סדרת הוראות, הצג את העצמים הנוצרים, וכלל עצם הצג את ערכיו התוכנות שלו.

```
A aa = new B(3, 10);
aa.Sub();
Object[] ar = new Object[6];
ar[0] = new A();
ar[1] = new A(5);
ar[2] = new B();
ar[3] = new B(5);
ar[4] = new B(2, 4);
ar[5] = aa;
((A)ar[3]).TripleX();
((B)ar[4]).TenTimesX();
```

/המשך בעמוד 31/

לפניך פועלה ראשית.

```
public static void Main()
{
    A a1 = new A(1);
    A a2 = new B(2, 99);
    /**
}
```

לפניך הקטעים (i)-(vi).

- (i) a2.DoubleX();
 Console.WriteLine(a2);
- (ii) a2.TenTimesX();
 Console.WriteLine(a2.TenTimesX());
- (iii) if (a2 is B)
 {
 a2.TenTimesX();
 Console.WriteLine(a2);
 }
- (iv) ((B)a1).TenTimesX();
 Console.WriteLine(a1);
- (v) a2.Calc();
 Console.WriteLine(a2);
- (vi) B bb = (B)a2;
 Console.WriteLine(bb.BaseX());

בעבור כל אחד מן הקטעים (i)-(vi) בערך:

כתבו את הקטע במקום \*\*\* // בפעולה הראשית.

קבע אם הקטע תקין או אינו תקין.

אם הקטע תקין — כתבו את הפלט המתkeletal בעקבות ביצועו.

אם הקטע אינו תקין — כתבו אם השגיאה היא שגיאת הידור (קומpileציה) או

שגיאת זמן ריצה.

**בצלהחה!**

אתם הצעירים שמוחה למדינת ישראל  
אין להעתיק או לפרסום אלא ברשות משרד החינוך

## תכנות מונחה עצמאים

אם למדת מסלול זה ואתה כותב ב- C#, ענה על אחת מהשאלות 15-16. (25 נקודות)

15. לפני חלק מפרויקט העוסק בכל תחבורה וכלול את המחלקות האליה:

מייצגת כלי תחבורה	המחלקה
מייצגת רכבת שהיא כלי תחבורה	Train
מייצגת סירה שהיא כלי תחבורה	Boat
מייצגת מטוס שהוא כלי תחבורה	Airplane
מייצגת חברת שיש לה כל תחבורה מוסגים שונים	TransportationCompany

## Public class Vehicle

```

private string type;           // מיקום (יבשה / מים / אויר)
private string way;           // סוג הדרך (כביש / מסילה / נהר / ...)
private int maxSpeed;         // מהירות מקסימלית

public Vehicle(string type, string way, int maxSpeed)
{
    this.type = type;
    this.way = way;
    this.maxSpeed = maxSpeed
}

```

## public class Train : Vehicle

```

private int numOfCarriages;     // מספר الكرונות
public Train(int maxSpeed, int numOfCarriages) : base("land", "tracks", maxSpeed)
{
    this.numOfCarriages = numOfCarriages;
}

public void IncNumOfCarriages(int n) // מגדילה ב- n את מספר الكرונות ברכבת
{
    this.numOfCarriages = this.numOfCarriages + n;
}

```

(שים לב: המשך השאלה בעמודים הבאים.)

**public class Boat : Vehicle**

```
public Boat(string way, int maxSpeed) : base("water", way, maxSpeed)
```

{

}

}

**public class Airplane : Vehicle**

```
private int maxHeight; // גובה טיסה מקסימלן
```

```
public Airplane(int maxSpeed, int maxHeight) : base("sky", "air", maxSpeed)
```

{

}

}

**public class TransportationCompany**

```
private Vehicle[] vehicles = new Vehicle[50]; // מערך כל התחבורה בחברה
```

```
private int counter = 0; // מספר כל התחבורה שיש בפועל
```

```
public TransportationCompany()
```

{

}

מוסיפה כל תחבורה לערך כל התחבורה של // מערך כל התחבורה בחברה.

החברה. הנה שיש מקום להוסיף כלי תחבורה. //

```
this.vehicles[counter] = v;
```

```
this.counter++;
```

}

}

(שים לב: המשך השאלה בעמוד הבא.)

א. ממש ב- C# מחלקה דואשית Program ובה פעולה ראשית, שתבצע את המשימות האלה:

בנייה של עצם מטיפוס חברת כל תחבורה – TransportationCompany

i. הנקרא company1

ii. הוספה של סירה אחת ורכבת אחת לחברת company1

בחר לתוכנות ערכיים כרצונך.

ב. במחלקה TransportationCompany הוגדרה הפעולה:

```
public void Display()
{
    for (int i=0; i<this.counter; i++)
    {
        Console.WriteLine((i+1) + ":" + this.vehicles[i]);
    }
}
```

ממש ב- C# פועלות שיאפשרו ביצוע תקין של הפעולה Display(), כך שבמעבר כל כל

חברה יודפסו כל התוכנות שלו. הגדר את הפעולות באופן המתאים ביותר לעקרונות של

תבניות מונחה עצמים (הכמסה – encapsulation, הורשה – inheritance,

פולימורפיזם – polymorphism).

בעבור כל פעולה שאתה ממש, רשום לאיזו מחלקה היא שייכת.

אין לשנות את הפעולה Display()

ג. ממש ב- C# פעולה, שתקבל מספר שלם n ותוסיף n קرونות לכל הרכבות. ששייכות

לחברת שיש לה כל תחבורה מסווגים שונים. تعد את הפעולה, ורשום באיזו מחלקה יש

להגדר אותה. אין לשנות את הפעולות הקיימות בפרויקט.

## לפניך המחלקות AA ו BB 16

public class AA

{

private string st;

public AA()

{ this.st = "excellent"; }

public AA(string st)

{ this.st = st; }

public string GetSt()

{ return this.st; }

public void SetSt(string st)

{ this.st = st; }

public override string ToString() { return "st = " + this.st; }

}

public class BB : AA

{

private int num;

public BB() : base()

{ this.num = 1; }

public BB(int num, string st) : base(st) { this.num = Math.Abs(num); }

public int GetNum()

{ return this.num; }

public void SetNum(int num)

{ this.num = num; }

public override string ToString()

{ return base.ToString() + " num = " + this.num; }

}

.א. הגדר במחלקה AA פעולה בוליאנית הניתנת לדרישה, בשם IsLike (Object obj), המתקבלת עצם obj מטיפוס AA, הינו מטיפוס Object. אם העצם obj הוא מטיפוס AA וגם תוכן המחרוזות st של obj זהה לתוכן המחרוזות st של העצם הנוכחי – הפעולה תחזיר true, אחרת – false.

.ב. הגדר במחלקה BB פעולה הדורשת את הפעולה שהגדרת בסעיף א.

אם העצם obj הינו מטיפוס BB וגם ערך התכונה num שלו זהה לערך התכונה num של העצם הנוכחי – הפעולה תחזיר true, אחרת – false.

(שים לב: המשך השאלה בעמוד הבא)

## ג. לפני קטע מפעולה ראשית:

```

AA a = new AA("excellent");
BB b = new BB();
a = b;
if (a.IsLike(b)) Console.WriteLine(a);
    
```

האם קטע התוכנית תקין?

אם כן — מה יהיה פלט הקטע? רשום איזו גרסה של הפעולה IsLike – זו של AA או זו של BB.

אם לא – הסבר מהי השגיאה ומתי היא תתגלת: בזמן קומpileציה או בזמן ריצה.

## ד. לפני קטע מפעולה ראשית:

```

AA aa = new AA();
BB bb = new BB(2, "excellent");
bb = aa;
if (bb.IsLike(aa)) Console.WriteLine(bb);
    
```

האם קטע התוכנית תקין?

אם כן — מה יהיה פלט הקטע? רשום איזוגרסה של הפעולה IsLike – זו של BB או זו של AA.

אם לא – הסבר מהי השגיאה ומתי היא תתגלת: בזמן קומpileציה או בזמן ריצה.

ה. כתוב פעולה חיצונית בשם LongString Object מקבלת מערך של עצמים מטיפוס AA

הפעולהמחזירה מהירות המרכיבת מרששו הוכונה st של עצמים מטיפוס AA

במערך, באופן זה:

אם לעצם יש בק הוכונה st, תשורר המחרוזות שבתוכנה st פעם אחת.

אם לעצם יש גם הגופה mut, המחרוזות שבתוכנה st תשורר mut פעמיים.

אם אין במערך אף עצם מטיפוס AA, תוחזר מחרוזות ריקה.

## בצלחה!

זכות היוצרים שמורה למדיינט ישראל  
אין להעתיק או לפחס אלא בשותפות משרד החינוך

תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב ב- C#, עם על אתת מן השאלות 15-16. (25 נקודות)

.15 לפניך הממשקים IOne, ITwo, IThree, IFour והמחלקות Gamma, Beta, Alpha .Program ו-

```
interface IOne {
    bool FirstA(Object x);
    void FirstB(int num);
}
```

```
interface ITwo {
    int Second();
}
```

```
interface IThree {
    int Third();
}
interface IFour {
    int Fourth();
}
```

```
public class Alpha : IOne {}
public class Beta : ITwo, IFour {}
public class Gamma : ITwo, IThree {}
public class Program {
    public static void Main() {}
}
```

A. כתוב את שמות הפעולות שיש למש בכל אתת מן המחלקות Gamma, Beta, Alpha ו- Program והסביר מדוע.

הנה שבסכט אחת מן המחלקות מומשו הפעולות שתכתבת בסעיף א.

ב. אם ההגדודה public class Omega : Ifour, Beta { } תקינה?

אם היא איינה תקינה — הסבר מדוע.

ג. לכל אחד מון הקטועים i-iv שלפניך, אם כתובו אותו בפועלה Main — קבוע אם הפעולה Main שתתקבל היא תקינה או איינו תקינה. נמק את קביעותך.

- i ITwo x1 = new ITwo();
- ii Beta b = new Beta();
- iii Alpha a = new Alpha();  
IOne x2 = a;
- iv Gamma c = new Gamma();  
IOne x3 = c;

ד. כל אחת מן הדרישות i-iii שלפניך קבוע אם אפשר לבצע אותה באמצעות כתיבת הוראה או הוראות בפועלה Main. אם אפשר — כתוב את הוראה המתאימה או את ההוראות המתאימות. אם אין אפשר — הסבר מדוע.

- i הפעלת הפעולה Second על עצם מטיפוס Beta .
- ii המרת עצם מטיפוס Gamma שייהי עצם מטיפוס Alpha .

16. ב שאלה זו ארבעה סעיפים, א-ד. אין קשר בין הסעיפים. עלייך לענות על כל הסעיפים.

- א. נתונת המחלקה A והמחלקה B היורשת מ- A .  
 ההוראה; A a1 = new B(); עברת ההידור (קומפילציה) ורצה בצורה תקינה.  
 בעבור כל אחד מן המשפטים -iii: שלפניך, קבע אם המשפט נכון או אינו נכון: הסביר את  
 קביעותיך.

i אי-אפשר לכתוב את ההוראה Object obj = a1 ; מכיוון שדרישה המורה מפורשת (casting) .

ii אי-אפשר לכתוב את ההוראה A a2 = a1 ; מכיוון שדרישה המורה מפורשת (casting)

iii אי-אפשר לכתוב את ההוראה B b1 = a1 ; מכיוון שדרישה המורה מפורשת (casting)

ב. נתונה הפעלה Program Main במחלקה Main :

```
public static void Main(string[] args) {  

    C c = new A();  

    B b1 = (B) (new A());  

    B b2 = new D();  

    A a = new D();  

}
```

קבע איזו מן האפשרויות -v שלפניך מתחייבת קשר בין המחלקות, לפי הפעולה Main הנתונה.

i C יורשת מ- A , B יורשת מ- A , D יורשת מ- A .

ii A יורשת מ- C , B יורשת מ- A , D יורשת מ- A .

iii C יורשת מ- A , B יורשת מ- A , D יורשת מ- B .

iv A יורשת מ- C , C יורשת מ- B , D יורשת מ- A .

v A יורשת מ- C , B יורשת מ- A , D יורשת מ- B , C יורשת מ- B .

## ג. נתונות הגדרות המחלקות City School :

```
public class City{ (1) int students; }
public class School : City {
    void Poll() { Console.WriteLine("There are "+ students + " in the school"); }
```

כדי שלא תהיה שגיאת הידור (קומפילציה) אפשר להשלים את חלק השורה המסומן (1) באמצעות אחת מן האפשרויות שלפניך. בחר באפשרות המתאימה והעתק אותה למחברתך.

- .protected , public , או
- .private public או
- .protected public או
- .protected public
- .private public
- .private public

## ד. לפניך הגדרה של המחלקה Quest :

```
public class Quest {
    private int num;
    private static int count = 0;

    public Quest()
    {
        count++;
        num = count;
    }

    public void PrintNow()
    {
        Console.WriteLine(num+" "+count);
    }
}
```

קבע כמה עצמים (אובייקטים) יש ליצור מן המחלקה Quest, ועל איזה מהם יש להפעיל את הפעולה PrintNow, כדי שהפלט יהיה 35.

**בהתכלחה!**

זכות היוצרים שמוצה למריית ישראל  
אין להעתיק או לפרסם אלא ברשות משרד החינוך

**תכנות מונחה עצמים**

אם למדת מסלול זה ואתה כותב ב- C#, ענה על אחת מן השאלות 15-16. (25 נקודות)  
**15.** לפניך פרויקט שהוגדרו בו המחלקות האלה: A, B, C, D, E. בפרויקט מומשה פעולה () בשתי מחלקות.  
 קטע קוד שלפניך תקין.

```
A a1 = new A();
A e1 = new E();
E c1 = new C();
C b1 = new B();
C d1 = new D();
```

נתון:

ההוראה; B d2=new D() גורמת לשגיאת הידור (קומפילציה).  
 הההוראה; a1 F() גורמת לשגיאת הידור (קומפילציה).  
 . "bye-bye" ((E)e1).F() תקינה ומדפסה ".  
 הההוראה; ((B)b1).F() תקינה ומדפסה "hello".  
 הההוראה; (D)b1).F() גורמת לשגיאת זמן ריצה.

- a.** צייר עץ ירושה של כל המחלקות, וציין בailo שתי מחלקות מומשה הפעולה () F().  
**b.** נתונה המחלקה Z :

```
public class Z
{
    public void G(){}
}
```

(1) הוסף את המחלקה Z לעץ הירושה שציירת בסעיף א כך שההוראה שלפניך תהיה תקינה:

Z x = new A();

(2) קטע הקוד שלפניך תקין.

```
A a2 = new A();
Z z1 = new Z();
Z a3 = new A();
```

לכל אחת מן ההוראות -v שלפניך, ציין אם היא תקינה או אינה תקינה.

אם הההוראה אינה תקינה, נמק מדוע.

הסעיפים -v **איןם תלויים** זה בזה:

- i a2 = z1;
- ii a3 = z1;
- iii ((A)z1).G();
- iv a2.G();
- v ((A)a3).G();

## ג. נתונה המחלקה Y :

```
public class Y
{
    public void M(){}
}
```

- (1) הוסף את המחלקה Y לעצם הירושה שצוייר בסעיף א כר שההוראה שלפניך תהיה תקינה  
(לא התייחסות לנתונים בסעיף ב).

C f = new Y();

- (2) קטע הקוד שלפניך תיקין.

A a2 = new A();

Y y1 = new Y();

C y2 = new Y();

עובד בלאחת מן ההוראות ז-ז שלפניך, ציין אם היא תקינה או אינה תקינה.

אם ההוראה אינה תקינה, נמק מדוע.

הסעיפים ז-ז איןם תלויים זה זהה.

i a2 = y1;

ii y2 = y1;

iii ((A)y1).M();

iv y2.M();

v ((A)y2).M();

16. תשלום מיסי ארנונה (Property tax) הוא תשלום לרשותות המקומיות בעבר שטח המגורים והקרקע הנוסף שבבעלות התושבים. מחיר הארץ הוא 10 ש"ח למ"ר של שטח המיועד למגורים ור. 0.5 ש"ח למ"ר של קרקע נוספת. לדוגמה, תושב שבבעלותו שטח למגורים בגודל 100 מ"ר ושטח קרקע בגודל 500 מ"ר משלם מיסי ארנונה בסכום של  $100 \cdot 10 + 500 \cdot 0.5 = 1250$ .

#### התושבים חולקו לשלווש קבועות:

- תושב עיר משלם מיסי ארנונה עבור שטח המגורים והקרקע שבבעלותו, ומתקבל מענק בסך 250 ש"ח לתשלום הארץ.
  - תושב עיר ותיכון – תושב עיר בן 60 ומעלה – משלם מיסי ארנונה כפי שמשלם תושב עיר, ונוסף לכך זכאי להנחה בהתאם לגיל: 1% הנחה עבור כל שנת חיים לאחר גיל 60. ההנחה מחושבת לאחר הפקחת המענק מהסכום המקורי.
  - תושב כפר משלם ארנונה עבור שטח המגורים והקרקע שבבעלותו וזכאי להנחה של 10%. תושב כפר בן 60 ומוללה אינו זכאי להנחה התלויה בגיל.
- אם לאחר הפקחת הנחה והמענק הסכום לתשלום מיסי הארץ הוא מספר שלילי, התושב פטור מהתשלום (משלם 0 ש"ח).
- וחולט לפתח תוכנה לחישוב מיסי ארנונה ולגביהם מההתושבים, ולשם כך הוגדר הממשק שפניך.

interface IData

{

```
string GetName(); //מחזירה שם של תושב
double GetPropertyTax(); //מחזירה גובה של מיס ארנונה
```

}

בסעיפים א-ב שלפניך עליך להגדיר את המחלקות לפי עקרונות של תוכנות מונחה עצמים בהתאם לדרישות:

- את הנתונים לחישוב מס הארץ (תשלום עבור שטח של דירה במ"ר, תשלום עבור שטח של קרקע במ"ר, שיעור הנחה באחוזים, סכום המענק בש"ח) הגדר פעמי אחת בלבד קבועים.
- את כל התכונות שאינן קבועות הגדר כפרטיות (private).

א. הגדר מחלקה Resident (המייצגת תושב), המימוש את הממשק IData, וכותב בה את תכונות המחלקה (אין צורך לכתוב פעולה בונה).

ב. – הגדר שלוש מחלקות כל אחת מקבוצות התושבים: CityResident – תושב עיר, VillageResident – תושב כפר, תושב עיר ותיכון.

כתב את תכונות המחלקות ואת הפעולה GetPropertyTax() שתאפשר לחשב את תשלום מס הארץ לכל אחת שלוש קבוצות התושבים (אין צורך לכתוב פעולה בונה).

/המשך בעמוד 27/

ג. נתון גוף פעולה המקביל לערך a של נתונים התושבים ומדפיס עבור כל תושב עיר ותיק את שמו ואת סכום הארנונה שהוא נדרש לשלם.

```
for ( int i = 0; i < a.Length; i++)
{
    if (a[i] is SeniorCityResident)
    {
        Console.WriteLine (a[i].GetName() + " " + a[i].GetPropertyTax());
    }
}
```

עבור כל אחת מכותרונות הפעולה (1)-(3) שלפניך, ציין אם הפעולה תקינה או אינה תקינה. אם הפעולה אינה תקינה — נמק מדוע, תקן את הפעולה, והעתיק את הפעולה המתוקנת למחברתך.

- (1) public static void Print(Object [] a)
- (2) public static void Print(IData [] a)
- (3) public static void Print(Resident [] a)

### **בצלחה!**

זכות רישיונים שמורות למדינת ישראל  
אין להעתיק או לפרסם אלא ברשות משרד החינוך

**C# – תכונות מונחה עצמים ב-**

אם למדת מסלול זה ואתה כותב ב- C#, ענה על אחת מן השאלות 15-16 נקודות (25)

15. נתונות המחלקות: A, B, Driver

```
public class A {
    private int val ;

    public A() {
        this.val = 1 ;
    }

    public A(int val) {
        this.val = val ;
    }

    public int GetVal () {
        return this.val ;
    }

    public virtual bool Equals (Object other) {
        Console.WriteLine ("AObject") ;
        if (other is A)
            return (this.val == ((A) other).val) ;
        return false ;
    }
}
```

```
public class B : A {
    private string st ;

    public B () {
        this.st = "B" ;
    }

    public B (string st, int val)
        : base (val) {
        this.st = st ;
    }

    public string GetSt () {
        return this.st ;
    }

    public override bool Equals (Object other) {
        Console.WriteLine ("BObject") ;
        if (other is B)
            return (this.st.Equals (((B) other).st)
                && this.GetVal () == ((B) other).GetVal ()) ;
        return false ;
    }

    public bool Equals (A other) {
        Console.WriteLine ("BA") ;
        if (other is B)
            return (this.st.Equals (((B) other).st)
                && this.GetVal () == ((B) other).GetVal ()) ;
        return false ;
    }

    public bool Equals (B other) {
        Console.WriteLine ("BB") ;
        return (this.st.Equals (other.st)
            && this.GetVal () == other.GetVal ()) ;
    }
}
```

```

public class Driver {
    public static void Main (string[] args) {
        A a1 = new A () ;
        A a2 = new A (5) ;
        A ab = new B () ;
        B b1 = new B ("B", 1) ;
        B b2 = new B ("B", 5) ;
        // ***
    }
}

```

**א.** ציר עצמים שנוצרו בפועלה Main. בעבר כל עץ عليك לציין את ערכי התכונות שלו.

**ב.** לפניו שמונה הוראות, 1-8.

- הצב כל אחת מן ההוראות במקום //\*\*\*// בפועלה Main של המחלקה Driver.
- כתוב מה יהיה הפלט בעקבות הצבת כל אחת מן ההוראות.

1. if (a1.Equals (b1)) Console.WriteLine (1) ;
2. if (b1.Equals (a1)) Console.WriteLine (2) ;
3. if (a1.Equals (ab)) Console.WriteLine (3) ;
4. if (ab.Equals (a1)) Console.WriteLine (4) ;
5. if (b1.Equals (ab)) Console.WriteLine (5) ;
6. if (ab.Equals (b1)) Console.WriteLine (6) ;
7. if (a1.Equals (a2)) Console.WriteLine (7) ;
8. if (b1.Equals (b2)) Console.WriteLine (8) ;

```

public class First
{
    public static int count=0 ;
    private string str ;

    public First (string str) {
        count ++ ;
        this.str = str ;
    }

    public First (First g) {
        count ++ ;
        this.str = "Copy" + g.str ;
    }

    public virtual void SetStr (string str) {
        this.str = str ;
    }

    public virtual void Print () {
        Console.WriteLine ("First" + this.str ) ;
    }
}

```

```

public class Second : First {
    private First f ;

    public Second (First fx, First fy)
    : base (fx) {
        this.f = fy ;
    }

    public override void SetStr (string str) {
        base.SetStr (str) ;
        this.f = new First (str) ;
    }

    public override void Print () {
        Console.WriteLine ("Second") ;
        base.Print () ;
        this.f.Print () ;
    }
}

```

(שים לב: המשך השאלה בעמוד הבא.)

```

public class Third {
    private int curr ;
    private First[] arr ;

    public Third (int count) {
        curr = 0 ;
        arr = new First [count] ;
    }

    public void Print () {
        for (int i = 0 ; i < curr ; i++)
            arr [i].Print () ;
    }

    public void Add (First s) {
        if (curr < arr.Length) {
            arr [curr] = s ;
            curr++ ; }
    }
}

```

```

public class Driver {
    public static void Main (string[] args) {
        First f1 = new First ("One") ;
        First f2 = new First ("Two") ;
        First f3 = new First (f2) ;
        Second s1 = new Second (f1, f3) ;
        Third t = new Third (First.count) ;
        t.Add (f1) ;
        t.Add (f2) ;
        t.Add (s1) ;
        Console.WriteLine (First.count) ;
        t.Print() ;
        Console.WriteLine ("-----") ;
        f1.SetStr ("Five") ;
        Console.WriteLine (First.count) ;
        t.Print () ;
    }
}

```

**א.** ציר מרים הידדיה בין המחלקות First , Second , Third . יש לסמן ירושה באמצעות החץ →

והכליה באמצעות הסימן ♦

**ב.** עקוב אחר ביצוע הפעולה Main של המחלקה : Driver

(1) הצג את התוכן של כל העצמים שנוצרו.

(2) רשום את הפלט שהתקבל.

## בצלחה!

זכות היוצרים שומרה למדינת ישראל  
אין להעתיק או למסס אלא ברשות משרד החינוך

**תכנות מונחה עצמים בשפת C#**

אם למדת מסלול זה ואתה כותב בשפת **C#**, ענה על אחת מן השאלות 15-16 (25 נקודות)

.15. נתונות המחלקות: **A, B, C, D, E**.

```
public class A
{
    protected int x ;
    public A ()
    {
        this.x = 9 ;
        Console.WriteLine ( "A. x = " +this.x ) ;
    }
    public A ( int x )
    {
        this.x = x ;
        Console.WriteLine ( "A. x = " + this.x ) ;
    }
    public int GetX () { return this.x ; }
    public virtual int Foo () { return this.x ; }
}

public class B : A
{
    public B () : base () {}
    public B ( int x ) : base ( x ) {}
    public override int Foo () { return this.x +1 ; }
}

public class C : B
{
    public C () : base () {}
    public C ( int x ) : base ( x ) {}
    public override int Foo () { return this.x +2 ; }
    public virtual int Bar () { return this.x ; }
}
```

```
public class D : C
{
    public D () : base ()
    {
        this.x ++ ;
        Console.WriteLine ( "D. x = " + this.x ) ;
    }
    public D ( int x ) : base ( x )
    {
        Console.WriteLine ( "D. x = " + this.x ) ;
    }
    public D ( int x, int y ) : base ()
    {
        this.x = this.x + x + y;
        Console.WriteLine ( "D. x = " + this.x ) ;
    }
    public override int Foo () { return this.x - 1 ; }

}

public class E : C
{
    public E () : base () {}
    public override int Bar () { return this.x +1 ; }
}
```

(שים לב: המשך השאלה בעמוד הבא.)

.————► א. סרטט תרשימים היררכיה בין המחלקות A , B , C , D , E . יש לסמן ירושה ממחלקה באמצעות החץ

ב. לפני כוורת הפעולה:  
public static int GetType ( Object m )

הפעולה מחזירה 1 אם m מטיפוס A , 2 אם m מטיפוס B , 3 אם m מטיפוס C , 4 אם m מטיפוס D  
ר- 5 אם m מטיפוס E .

כמיש את הפעולה.

הדרך: כדי לבדוק את סוג העצם יש להיעזר בפעולות GetX , Bar , Foo

. A , B , C , D , E , Object , ו- as ובפעולות של המחלקה אין לשנות את המחלקות הנח ש- m שייר לאחת מן המחלקות E , B , C , D , A ו-ain null .

ג. לפני המחלקה : Tester

```
public class Tester
{
    public static void Main ( string [] args )
    {
        A a1 = new B () ;
        A a2 = new E () ;
        A a3 = new D () ;
        A a4 = new D ( 5 ) ;
        A a5 = new D ( 3, 7 ) ;
    }
}
```

כתב את הפלט של הפעולה Main .

16. בחברת התקשורות "אוֹזְנִים לְעַתִּיד" יש 4 סוגי עובדים. כל עובד שירק لسוג אחד בלבד.

**עובד רגיל** — עובד רגיל בחברה.

**חבר ועדי** — עובד לחבר בוועד העובדים.

**טכני** — עובד שאחראי על כמה מחשבים בחברה (כל טכני אחראי על מספר שונה של מחשבים).

**אחראי פרויקט** — עובד שמנהל פרויקטים מיוחדים בחברה. הוא אחראי על 15 עובדים ובهم 10 עובדים רגילים, חבר ועדי וטכנאים.

החלטות מתקובלות באמצעות הצבעה בהשתתפות כל עובדי החברה. הצבעה נערכת באמצעות נקודות. לכל עובד בחברה יש מספר נקודות שונה, לפי סוג:

לפניך טבלה ובה נתונים בנוגע למספר הנקודות שיש לכל עובד לפי סוג:

סוג העובד	מספר הנקודות
עובד רגיל	מקבל 4 נקודות עבור היוטו עובד בחברה, ועוד נס"ן על כך הוא מקבל עוד נקודה עבור כל שנת ותק. <b>דוגמה:</b> לעובד רגיל שעבוד 5 שנים בחברה יש <b>9</b> נקודות: 4 נקודות עבור היוטו עובד בחברה ועוד 5 נקודות — נקודה עבור כל שנת ותק בחברה.
חבר ועדי	מקבל פי 2 ממספר הנקודות שעבוד רגיל מקבל, ועוד נס"ן על כך הוא מקבל 2 נקודות עבור כל שנה שהוא חבר בוועד. <b>דוגמה:</b> לחבר ועדי שעבוד 5 שנים בחברה ומתוכן הוא חבר בוועד 3 שנים יש <b>24</b> נקודות: 18 נקודות — פי 2 מעובד רגיל שעבוד 5 שנים בחברה ועוד 6 נקודות — 2 נקודות עבור כל שנה שהוא חבר בוועד.
טכנאי	מקבל נקודות כמו עובד רגיל, ועוד נס"ן על כך הוא מקבל נקודה עבור כל מחשב שבאחריותו. <b>דוגמה:</b> לטכני שעבוד 5 שנים בחברה ואחראי על 4 מחשבים יש <b>13</b> נקודות: 9 נקודות — כפי שמקבל עובד רגיל שעבוד 5 שנים בחברה ועוד 4 נקודות — נקודה עבור כל מחשב שבאחריותו.
אחראי פרויקט	מקבל נקודות כמו עובד רגיל, ועוד נס"ן על כך הוא מקבל מספר נקודות השווה לסך הנקודות של כל 15 העובדים שעלייהם הוא אחראי. <b>דוגמה:</b> לאחראי פרויקט שעבוד 5 שנים בחברה, סך כל הנקודות של 15 העובדים שבאחריותו הוא 80, יש <b>89</b> נקודות: 9 נקודות — כפי שמקבל עובד רגיל שעבוד 5 שנים בחברה ועוד 80 נקודות — סך כל הנקודות של 15 העובדים שבאחריותו.

(שים לב: המשך השאלה בעמוד הבא.)

בחברה ווצים לבנות מערכת ממוחשבת שתציג את מספר הנקודות של כל עובד בחברה. לשם כך הגדרו 4 מחלקות: **Employee** – עובד רגיל, **UnionMember** – חבר ועד, **Technician** – טכני, **Supervisor** – אחראי פרויקט. כל עובד שייר למחלקה אחת בלבד.

את הפעולות ש滥בניך יש לפטור באופן המתאים ביותר לעקרונות תכונות מונחה עצמים.

**א.** סרטט תרשימים הייררכיה בין המחלקות. יש לסמן ירושה ממחילה באמצעות החץ → והכללה באמצעות הסימן ◆ .

**ב.** עבור כל אחת מן המחלקות כתוב את כוורתה המחלקה, את התכונות שלה ואת הפעולה () המחזירה את מספר הנקודות בהצעה. אין צורך לכתוב פעולה בונה ופעולות ו- Set GetScore .

הנחה: הנתונים הקבועים של מספר הנקודות אינם חלק מתכונות המחלקות, ויש להשתמש בהם רק בכתיבת הפעולה GetScore .

**ג.** הoulתת הצעה לשנות את שם החברה ל"תקשות עתידית". כל העובדים הרגילים והטכנאים תמכו בהצעה וכל חברי הוועד והאחראים לפרויקטים התנגדו לה. רק אם סך הנקודות بعد ההצעה גדול ממספר הנקודות נגד ההצעה – ההצעה מתתקבלת, אחרת היא אינה מתתקבלת.

נתונה כוורתה פעולה מקבלת מערך של כל העובדים ומחזירה true אם ההצעה התקבלה, אחרת היא מחזירה false .

```
public static bool IsAccepted (Object [] arr)
```

ממש את הפעולה.

## בצלחה!

**תכנות מונחה עצמים**

אם למדת מסלול זה ואתה כותב בשפט #C, ענה על אחת מן השאלות 15-16. (25 נקודות)

15. במועדון כושר "השלום" אפשר להתאמן בכל מתקני הכוח. נוסף על כך אפשר להתאמן אימון מיוחד שבו אפשר להשתמש בכל מתקני הכוחם בילויו של אמן אישי לזמן מוגדר. כדי לשמר מידע על המתרחש במועדון נבנו המחלקות שלפניך:

. Management – אימון – Trainer – Client – SpecialT – Training , הנהלה – Client –珂仇 – SpecialT – אימון – Trainer – Management

התכונות של המחלקות הן בהתאם לדרישות המידע שלפניך:

**המידע ששומרים על מאמן:**

- name — שם המאמן, מטיפוס מחרוזת.
- wid — מספר עובד, מטיפוס שלם.

**המידע ששומרים על אימון:**

- sumt — מספר המתקנים שבהם השתמש הלקוח באימון, מטיפוס שלם.

**המידע ששומרים על אימון מיוחד:**

- sumt — מספר המתקנים שבהם השתמש הלקוח באימון מיוחד, מטיפוס שלם.
- trainer — המאמן שילוה את הלקוח באימון, מטיפוס Trainer .
- time — מספר הדקות של האימון, מטיפוס שלם. המספר גדול מ- 0 ואינו מוגבל ל- 60 .

**המידע ששומרים על לקוחות המועדון:**

- id — מספר תעודת זהות, מטיפוס מחרוזת.
- name — שם הלקוח, מטיפוס מחרוזת.
- visits — כל האימונים שהיו ללקוח, במערך מטיפוס Training .

**המידע ששומרים בעבור הנהלה:**

- staff — כל המאמנים העובדים במועדון, במערך מטיפוס Trainer .
- clients — כל הלקוחות של המועדון, במערך מטיפוס Client .

(שים לב: המשך השאלה בעמוד הבא.)

על הסעיפים שלפניך יש לענות באופן המתאים ביותר לעקרונות תכנות מונחה עצמים.

א. סרטט תרשימים היררכיה בין המחלקות.

יש לסמן ירושא בעורת החץ:

יש לסמן הכלה באמצעות הסימן:

ב. כתוב את כוורות ותכנות המחלקה.

הנה שיש פעולה בונה ופעולות Get ו- Set בכל המחלקות.

ג. כתוב פעולה פנימית במחלקה **Client**, המחזירה את סכום כל הdkות שלקוות התאמן אימון מיוחד.

הפעולה תחזיר 0 אם הלוקוח אף פעם לא התאמן אימון מיוחד.

ד. כתוב פעולה פנימית במחלקה **Management**, המחזירה את סך כל הלקוחות שהתאמנו אימון מיוחד

לפחות פעם אחת.

16. בוחנות הכלבו "הכול לבית" מוכרים מוצרי מזון שאינם דורשים אחסון בקירור, מוצרי מזון הדורשים אחסון בקירור ומוצרי אלקטרוני.

כדי לנחל את מלאי המוצרים שבחנות נבנו המחלקות האלה:

— מוצר, **FoodItem** — מוצר מזון, **FoodRefrigerated** — מוצר מזון הדורש אחסון בקירור, **ItemDate** — מוצר אלקטרוני,

. **ItemDate** היא מחלוקת שמייצגת תאריך ויש בה 3 תכונות מטיפוס מספר שלם: יום day, חודש month, שנה year.

התכוונות של המחלוקות הן בהתאם לדרישות המידע שלפניך:

המידע ששמורים בנוגע לכל אחד מן המוצרים בחנות:

- name — שם המוצר, מטיפוס מחרוזת.
- catalogNumber — מספר קטלוגי, מטיפוס מחרוזת.
- quantity — מספר פריטים מן המוצר במלאי, מטיפוס שלם.
- minQuantity — המספר המינימלי של הפריטים מן המוצר שיש להחזיק במלאי, מטיפוס שלם.

המידע הנוסף ששמורים בנוגע לכל מוצר המזון:

- . **ItemDate** — תאריך התפוגה של המוצר, מטיפוס .

המידע הנוסף ששמורים בנוגע למוצר המזון הדורשים אחסון בקירור:

- minTemperature — הטמפרטורה המינימלית הנדרשת לאחסן מוצר המזון, מטיפוס שלם.
- maxTemperature — הטמפרטורה המקסימלית האפשרית לאחסן מוצר המזון, מטיפוס שלם.

המידע הנוסף ששמורים בנוגע למוצר אלקטרוני:

- . **ItemDate** guaranteeDate — תאריך סוף אחריות של הייצור על המוצר, מטיפוס .

הערה: מוצרים המזון שאינם דורשים אחסון בקירור ומוצר אלקטרוני נשמרים בכל טמפרטורה.

אפשר להניח שכל המחלוקות יש פועלות בונות ופועלות Get ו Set.

(שים לב: המשך השאלה בעמוד הבא.)

**א.** סרטט תרשימים היררכיה בין המחלקות:

**Item, ElectronicItem, FoodItem, FoodRefrigerated, ItemDate.**

- יש לסמן ירושה באמצעות החץ:
- ◆ יש לסמן הכללה באמצעות הסימן:

**ב.** מימוש את הפעולה הפנימית () IsMissing. הפעולה תחזיר true בעבור המוצר אם מספר הפריטים במלאי

כמפורט ממספר הפריטים המינימלי הנדרש במלאי, אחרת היא תחזיר false.

יש לציין באיזו מחלוקת נמצאת הפעולה.

**ג.** במחלקה Tester הוגדרה הפעולה:

```
public class Tester {
    public static bool CanBeStored (Item item, int temp) {
        return item.CanBeStored (temp);
    }
}
```

הפעולהמחזירה true אם אפשר לשמור את המוצר item בטמפרטורה temp, אחרת היא מחזירה false. הוסף פעולות נדרשות למחלוקת כדי שהפעולה תבצע את הנדרש.

ציין בעבור כל פעולה באיזו מחלוקת היא נוספת.

**בצלחה!**

**C#** **תכנות מונחה עצמים בשפת**

שאלות 16-17 מייעדות לכוטבים בשפת C#.

16. כדי לנוהל את האירועים שלה בניתה חברת "המשך" את המחלקה **Date** – תאריך, שלה שלוש תכונות מטיפוס שלם: **יום** – day , **חודש** – month ,  **שנה** – year .

נוסף על כן, לצורך ניהול האירועים בניתה המחלקה **Event** – אירוע.

לחברה יש שלושה סוגי אירועים: פגישות,שיחות טלפון ומשימות. כדי לשמר את המידע שבכל סוג אירוע, בניתה החברה את המחלקות האלה: פגישה – **Meeting** , שיחת טלפון – **PhoneCall** , משימה – **Task** . התכונות של המחלקות הן בהתאם לדרישות המידע שלפנינו:

**המידע ששומרים על פגישה – Meeting**

**Date** – מטיפוס date

– שעת התחלה, מטיפוס שלם hour

arrNames – שמות המשתתפים בפגישה, במבנה מטיפוס מחוوظת (גודל המערך הוא לפי מספר המשתתפים)

duration –משך הפגישה בדקות, מטיפוס שלם

location – מקום הפגישה, מטיפוס מחוوظת

**המידע ששומרים על שיחת טלפון – PhoneCall**

**Date** – מטיפוס date

– שעת התחלה, מטיפוס שלם hour

phoneNumber – מספר הטלפון שמחייגים, מטיפוס מחוوظת

name – שם האדם שאליו מתקשרים, מטיפוס מחוوظת

**המידע ששומרים על משימה – Task**

**Date** – מטיפוס date

– שעת התחלה, מטיפוס שלם hour

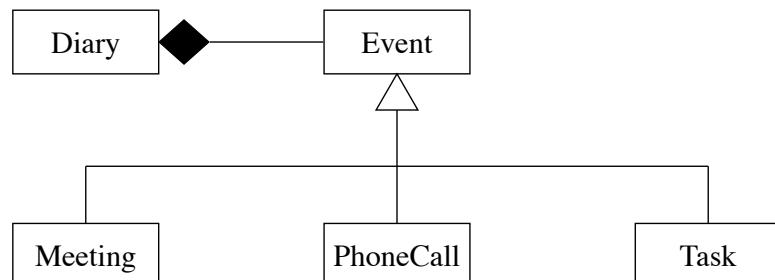
title – כותרת המשימה, מטיפוס מחוوظת

**המידע ששומרים על אירוע – Event**

אותן דרישות המידע המפורטות בסוגי אירועים שלילי וצרכיות להישמר במחלקה **Event** בהתאם לעקרונות תכונות מונחה עצמים.

נוסף על המחלקות האמורות, בניתה המחלקה **Diary** – יומן, השומרת עד 1,000 אירועים – **Event** . במבנה – arr גודל 1,000 מטיפוס **Event**

לפניר תרשימים חלקו של המחלקות:



על הסעיפים שלפניך יש לענות לפי עקרונות של תכניות מונחה עצמים.

- A.** העתק את התרשים למחברתך והוסף את המחלקה **Date** לתרשים. כתוב את כוותרת המחלקה אירוע –  
ואם התרומות שלב

הנ"ח שבסחר המחלקות כתובות התכווית המתאימות, ושבכל המחלקות יש פעולה בוגרת ופערות Get ר. Set .

- ב.** כתוב במחלקה **Diary** פעולה בשם AllCalls מקבלת עצם — date מטיפוס **Date**, ומחזירה מערך מטיפוס **PhoneCall** בגודל 100. המכיל את כל שיחות הטלפון בתחום — date — .

שים לב:

— הנק שקיימת הפעולה שלפניך במחלקה : Date

public bool Same (Date other)

הפעולה מחזירה true אם התאריך הנוכחי זהה ל- other , אחרת היא מחזירה false . אפשר להשתמש בפעולת בילוי למשוך אותה.

— הנח שיש בדיקת 100 שיחות טלפון בתאריך — date .

הנץ כי בעת הקרייה לפעולה זו, מערך האירועים — arr במקלחת Diary מלא בעצמים (ללא null).

ג. לפניך כוורתת הפעולה הפנימית – Match במחלקה

```
public virtual bool Match (string name)
```

זימון הפעולה יחזיר true אם האירוע (Event) הוא פגישה (Meeting) שבה משתתף אדם ששמו זהה ל- name או אם האירוע הוא שיחת טלפון (PhoneCall) עם אדם ששמו זהה ל- name . אחרת יוחזר false.

חמש את היטולו, וווסף פעולות לאחלהות המחייבות כדי לברא את הודרשו

אינו בעבור כל פעולה באיזו מחלוקת היא מוסכת.

הערה: בסעיף זה, אין להשתמש בפעולות `is` ו-`as` ובפעולות של המחלקה `Object`.

/המשר בעמוד 30/

```
public class A {
    public A() { Console.WriteLine("ctor A"); }
    public virtual void Foo(int x) {
        Console.WriteLine("A Foo int " + x);
    }
    public void Foo(double y) {
        Console.WriteLine("A Foo double " + y);
    }
    public void Bar(int x) {
        Console.WriteLine("A Bar " + x);
        Foo(x);
    }
}
```

```
public class B : A {
    public B() { Console.WriteLine("ctor B"); }
    public override void Foo(int x) {
        Console.WriteLine("B Foo int " + x);
    }
    public void Bar() {
        Console.WriteLine("B Bar");
        Foo(2);
    }
    public void Another(int x) {
        Console.WriteLine("B Another " + x);
        base.Foo(x);
        Foo(2.0 * x);
    }
}
```

```
public class Tester {
    public static void Main (string [] args)
    {
    }
}
```

**שים לב: המשך השאלה בעמוד הבא.**

לפניך 15 קטעי קוד. בחר עשרה מהם.

הצב כל אחד מקטעי הקוד שבחורת בפועלה Main במחלקה Tester, כתוב את מספר הקטע במחברתך וציין אם הקוד תקין או אינו תקין. אם הקוד תקין — כתוב את הפלט, ואם הוא אינו תקין — הסבר את השגיאת.  
קטעי הקוד אינם קשורים זה לזה.

<b>1.</b> <code>A a = new A(); a.Foo(2);</code>	<b>2.</b> <code>A a = new A(); ((B)a).Foo(3);</code>	<b>3.</b> <code>A a = new A(); B b = a; b.Foo(2);</code>
<b>4.</b> <code>A x = new B(); x.Foo(2);</code>	<b>5.</b> <code>A x = new B(); x.Bar();</code>	<b>6.</b> <code>A a = new A(); a.Bar(3);</code>
<b>7.</b> <code>A a = new A(); a.Bar();</code>	<b>8.</b> <code>B b = new B(); b.Bar();</code>	<b>9.</b> <code>B b = new B(); b.Bar(3);</code>
<b>10.</b> <code>B b = new B(); b.Foo(2); b.Foo(2.0);</code>	<b>11.</b> <code>A a = new A(); a.Foo(2); a.Foo(2.0);</code>	<b>12.</b> <code>B b = new A(); b.Another(2);</code>
<b>13.</b> <code>A a = new A(); a.Another(2);</code>	<b>14.</b> <code>B x = new B(); x.Another(2);</code>	<b>15.</b> <code>A x = new B(); x.Another(2);</code>

## בצלחה!

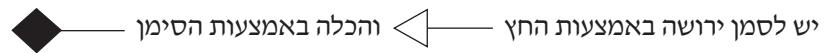
**תכנות מונחה עצמים**

השאלות 16-17 מיועדות לכוכבים בשפת C# .

.16 לפניך המחלקות A, B, C, D .

<pre>public class A {     public static int count = 0;     protected int num;      public A() {         this.num = 3;         count++;     }      public int GetNum() {         return this.num;     }      public void SetNum(int num) {         this.num = num;     }      public void Foo(int num) {         this.num = num;         num++;     }      public bool IsEqual(A other) {         return (this.num == other.num);     }      public virtual void Func() {         Console.WriteLine("I am A");     } }</pre>	<pre>public class B : A {     public override void Func() {         Console.WriteLine("I am B");     }      public bool IsEqual(B other) {         return (this == other);     } }  public class C : A { }  public class D {     private A a;      public D(A a) {         this.a = a;     }      public void Func() {         (this.a).Func();     }      public void IsB() {         Console.WriteLine((this.a) is B);     } }</pre>
---	--

. א. צייר מפת היררכיה בין המחלקות D, C, B, A.



. ב. צייר את העצמים שנוצרו בפועלה Main במחלקה Tester שלפניך, וכותב את הפלט.

```
public class Tester {
    public static void Main(string[] args) {
        A y1 = new A();
        A y2 = new A();
        y1.Foo(y1.GetNum());
        Console.WriteLine(y1.AreEqual(y2));
        B y3 = new B();
        A y4 = new B();
        Console.WriteLine(y3.AreEqual(y4));
        Console.WriteLine(y4.AreEqual(y3));
        B y5 = y3;
        y5.SetNum(0);
        Console.WriteLine(y3.AreEqual(y5));
        C y6 = new C();
        Console.WriteLine(((B)y4)..AreEqual(y6));
        D d1 = new D(y4);
        D d2 = new D(y6);
        Console.WriteLine(B.count);
        d1.Func();
        d2.Func();
        d1.IsB();
        d2.IsB();
    }
}
```

.17. בשאלת זו שני סעיפים, א-ב, שאין קשר ביניהם. עליך לענות על שניהם.

א. נתונות המחלקות האלה:

```
public class C {
    public virtual void Dolt(D d) { Console.WriteLine("a"); }
}

public class D : C {
    public override void Dolt(D d) { Console.WriteLine("b"); }
    public void Dolt(C c) { Console.WriteLine("x"); }
}

public class CDTest {
    public void Main (string[] args) {
        C cd = new D();
        D dd = (D) cd;
        *****
    }
}
```

(1) בעבר כל אחת מן ההוראות 1-3 שלפניך, ציין אם הוראה תקינה אם היא תחליף את השורה \*\*\*\*

בפעולה Main . אם כן, רשום את הפלט, ואם לא, נמק מדוע הוראה אינה תקינה.

dd.Dolt(cd); .1

cd.Dolt(dd); .2

cd.Dolt(cd); .3

(2) למחלקה C נוספה הפעולה שלפניך:

```
public void Dolt (Object o) {
    Console.WriteLine("o");
}
```

(\*) בעבר כל אחת מן ההוראות 1-3 שלפניך, ציין אם הוראה תקינה אם היא תחליף את השורה \*\*\*\*

בפעולה Main . אם כן, רשום את הפלט, ואם לא, נמק מדוע הוראה אינה תקינה.

dd.Dolt(cd); .1

cd.Dolt(dd); .2

cd.Dolt(cd); .3

(שים לב: המשך השאלה בעמוד הבא.)

.ב. נתונות המחלקות AA, BB.

<pre>public class AA {     public AA() { Console.WriteLine ("In AA"); }     public AA(int n) : this() {         Console.WriteLine ("In AA");         Console.WriteLine (n * 2);     } }</pre>	<pre>public class BB : AA {     public BB() { Console.WriteLine ("In BB"); }     public BB(double k) {         Console.WriteLine ("In BB" + (int)k);     }     public BB(int k) : base(k * 2) {         for (int i = 1; i &lt; k; i++)             Console.WriteLine ("In BB");     } }</pre>
---	---

לפניך המחלקה Run :

```
public class Run {
    public static void Main (string[] args) {
        ****
    }
}
```

במחלקה Run בפועלה הראשית Main המקום המסומן ב-\*\*\*\* מייצג שורת קוד כלשהי.  
לפניך שלושה פלטים שאין קשור ביניהם. בעבר כל פלט כתוב את שורת הקוד שمفיקה אותו.

(1) In AA

In BB

(2) In AA

In BB7

(3) In AA

In AA

12

In BB

In BB

הערה: עליך להתבסס על הפעולות הבוניות המוגדרות במחלקות AA ו- BB. אין להוסיף פעולות והדפסות.

**בצלחה!**

זכות היוצרים שמורה למדינת ישראל  
אין להעתיק או לפרסם אלא ברשות משרד החינוך

## תכנות מונחה עצמים בשפת #C

אם למדתם מסלול זה ואתם כותבים בשפת C#, יש לענות על אחת מן השאלות 16–17 (25 נקודות).

16. בעל חנות כלבו מעוניין למכור את מוצריו גם באינטרנט (אונליין). בשלב הראשון החליט בעל החנות למכור באופן רק חולצות ומכנסיים.

לשם כך נבנו המחלקות האלה:

– המחלקה **Clothing**, המייצגת בגד

– המחלקה **Shirt**, המייצגת חולצה

– המחלקה **Pants**, המייצגת זוג מכנסיים

```
public class Clothing {
    private int id;           // מספר זיהוי של המוצר
    private string fabric;   // סוג הבד
    private string color;    // צבע הבגד
    private double price;    // מחיר
}

public class Shirt : Clothing {
    private string size;     // (S, M, L, XL, XXL) מידת
}

public class Pants : Clothing {
    private int length;      // אורך באינץ'ים
    private int width;       // רוחב באינץ'ים
}
```

הנicho שיש פועלות Get ו- Set בעבר התכונות בכל המחלקות.

A. המחלקה **Cart** מייצגת סל קניות. למחלקה שלוש תכונות:

name – שם הקונה

arr – מערך המכיל גם חולצות וגם מכנסיים

current – מספר המוצרים הנוכחי בקנייה

(1) כתבו את כוורת המחלקה Cart ואת התכונות שלה.

(2) סרטטו תרשימים הייררכיה המתאר את הקשר בין המחלקות: Clothing , Shirts , Pants , Cart

יש לסמן ירושה באמצעות החץ  והכלא באמצעות הסימן .

(שים לב: המשך השאלה בעמוד הבא.)

**ב.** בשלב השני החליט בעל הchnoot למכור באופן מקוון גם ספרים.

לפניהם המחלקה **Book** המייצגת ספר:

```
public class Book {
    private int id;           // מס' זיהוי של המוצר
    private string bookName;  // שם הספר
    private string author;   // שם הסופר
    private double price;    // מחיר
}
```

הניחו שיש פועלות Get ו- Set בעבר תכונות המחלקה.

כדי שהמערך arr במחלקה Cart יוכל להכיל גם ספרים, נעשו הפעולות האלה:

- הוגדרה המחלקה – **Product** – מוצר.

- טיפוס המשתנה arr במחלקה Cart שונה ל- **Product**.

- נכתבו במחלקות המתאימות (כולל במחלקה Book) יחסיו הורשה חדשים בין ובין המחלקה **Product**.

- חלק מן התכונות שהופיעו עד עכשוו בכמה מן המחלקות האחרות (כולל המחלקה Book) הועברו

למחלקה **Product**, בהתאם לעקרונות תכונות מונחה עצמים (מעבר לתכונות אלו אין תכונות נוספות

במחלקה **Product**).

**(1)** סרטטו תרשימים הייררכיה מעודכן המתאר את הקשר בין המחלקות:

. Clothing , Shirt , Pants , Book , Product , Cart

יש לסמן ירושה באמצעות החץ ─────────── ◆ ─────────── והכליה באמצעות הסימן ─────────── .

**(2)** כתבו את כוורת המחלקה **Product** ותכונתייה.

**(3)** כתבו מחדש את המחלקות שחל בהן שינוי בעקבות הוספה של המחלקה **Product**.

שים לב: יש להתייחס גם ליחסו הורשה חדשים וגם לשינוי שחל בתכונות המחלקה.

**ג.** בשלב האחרון, החליט בעל הchnoot להכריז על מוצר זמני: רכישה מקוונת כל המוצרים יימכרו בהנחה של 10%.

ובעבר רכישת בגדים (חולצות ומכנסיים) תהיה הנחה נוספת של 10 שקלים (הניחו שהמחיר נשאר חיובי לאחר ההנחה).

נתונה הפעולה הפנימית במחלקה **Product**:

```
public virtual double GetDiscountPrice()
{
    זימון הפעולה מחזיר את מחיר המוצר (חולצת/מכנסיים/ספר) לאחר ההנחה – בהתאם לכל מוצר.
    מימוש את הפעולה במחלקה Product, והוסיפו פעולה/ות במחלקה/ות המתאימה/ות כדי לבצע את הנדרש.
    ציינו בעבר כל פעולה שהוספה באיזו מחלקה היא נוספת.
    אין להשתמש בפעולות is ו- as ובפעולות המחלקה Object ואין לשנות את תכונות המחלקות.
```

.17 נתונות שלוש מחלקות – Mammal (יונק), Antelope (אנטילופה) ו- Beaver (בונה) :

```
public class Mammal {
    protected int weight;

    public Mammal (int w) {
        weight = w;
    }

    public int GetWeight () {
        return weight;
    }

    public virtual bool IsSame (Mammal other) {
        Console.WriteLine ("In Mammal");
        return (this == other);
    }
}

public class Antelope : Mammal {
    public Antelope (int w) : base (w) {}

    public bool IsSame(Antelope other) {
        Console.WriteLine ("In Antelope");
        return ((other != null) && (this.weight == other.weight));
    }
}

public class Beaver : Mammal {
    public Beaver (int w) : base (w) {}

    public override bool IsSame (Mammal other) {
        Console.WriteLine ("In Beaver");
        return ((other != null) && (other is Beaver) && (this.weight == ((Beaver)other).weight));
    }
}
```

(שימוש ל-**המשך** השאלה בעמוד הבא.)

```

public class Program {
    public static void Main (string[] args) {
        Antelope a1 = new Antelope (10);
        Object a2 = new Antelope (10);
        Beaver b1 = new Beaver (10);
        Mammal b2 = new Beaver (10);
        *****
    }
}

```

הציבו כל אחת מן השורות 1–10 שלහלן בפעולה Main , במקומות שמסומן לעיל בכוכביות \*\*\*.

כתבו במחברת את מספר השורה וציינו אם הקוד תקין או לא תקין. אם הקוד תקין – כתבו את הפלט, ואם הוא לא תקין – הסבירו מדוע.

1. Console.WriteLine (a1.weight);
2. Console.WriteLine (((Beaver)a2).GetWeight());
3. Console.WriteLine (a1.Equals (a2));
4. Console.WriteLine (a2.Equals (a1));
5. Console.WriteLine (b1.Equals (b2));
6. Console.WriteLine (b2.Equals (b1));
7. Console.WriteLine (a1.Equals ((Beaver)b2));
8. Console.WriteLine (a1.Equals ((Antelope)a2));
9. Console.WriteLine (b1.Equals ((Antelope)a2));
10. Console.WriteLine (b1.Equals ((Beaver)a2));

## בצלחה!