

## חומרים שהוכנו על-ידי מורי הניסוי תש"ע להוראת "יסודות מדעי המחשב"

ניתן להשתמש בחומרים לצורך הוראה בלבד.  
לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי  
ללא קבלת אישור מראש מצוות הפיתוח

### אוסף מחלקות גרפיות

הסבר כללי

כתיבה ופיתוח:

ולרי פקר

## אוּסַף מַחֲלָקוֹת גֵּרְפִיּוֹת

הַקּוֹבֵץ *Games.dll* מְכִיל אוּסַף מַחֲלָקוֹת גֵּרְפִיּוֹת, הַמְאוּרְגָנוֹת בְּמִרְחָבֵי שְׂמוֹת (*namespace*).

הַטְּבֵלָה הַבָּאָה מְסַכֶּמֶת אֶת כָּל מִרְחָבֵי הַשְּׂמוֹת וְהַמַּחֲלָקוֹת שְׂקִימּוֹת בֵּהֶם:

הַמַּחֲלָקוֹת	תֵּיאוּר	שֵׁם הַמִּרְחָב
<i>Turtle</i>	מִרְחָב זֶה מְכִיל אֶת הַמַּחֲלָקָה הַגֵּרְפִיּוֹת לְצִיּוֹר בְּעִזְרַת 'צִב'.	<i>using Games.TurtleLib;</i>
<i>Die</i>	מִרְחָב זֶה מְכִיל אֶת הַמַּחֲלָקָה הַגֵּרְפִיּוֹת 'קוֹבִיָּה', בְּעֵלֶת 6 פְּאוֹת, כֹּאשֶׁר עַל הַפְּאוֹת מוֹפִיעִים הַמְסַפְרִים 1 עַד 6. הַמַּחֲלָקָה הַנְּחוּצָה לְסִימּוּלָצִיָּה שֶׁל "הַטֵּלֶת קוֹבִיָּה".	<i>using Games.DieLib;</i>
<i>Dreidle</i>	מִרְחָב זֶה מְכִיל אֶת הַמַּחֲלָקָה הַגֵּרְפִיּוֹת 'סְבִיבּוֹן', בְּעֵלֶת 4 פְּאוֹת, כֹּאשֶׁר עַל הַפְּאוֹת מוֹפִיעוֹת הָאוֹתִיוֹת נ ג ה ש. הַמַּחֲלָקָה הַנְּחוּצָה לְסִימּוּלָצִיָּה שֶׁל "הַגְּרֵלֶת סְבִיבּוֹן".	<i>using DreidleLib;</i>
<i>BrownFrog</i> <i>GreenFrog</i>	מִרְחָב זֶה מְכִיל אֶת שְׁנֵי הַמַּחֲלָקוֹת הַגֵּרְפִיּוֹת 'צִפְרָדַע יְרוּקָה' ו-'צִפְרָדַע כְּחוֹלָה', הַנְּחוּצוֹת לְסִימּוּלָצִיָּה שֶׁל פִּיתְרוֹן "בְּעִיַּת 4 הַצִּפְרָדַעִים" עַל לוּחַ בֶּן חֲמֵשׁ מִשְׁבָּצוֹת.	<i>using Games.FrogLib;</i>
<i>BrownFrog7</i> <i>GreenFrog7</i>	מִרְחָב זֶה מְכִיל אֶת שְׁנֵי הַמַּחֲלָקוֹת הַגֵּרְפִיּוֹת 'צִפְרָדַע יְרוּקָה' ו-'צִפְרָדַע כְּחוֹלָה', הַנְּחוּצוֹת לְסִימּוּלָצִיָּה שֶׁל פִּיתְרוֹן "בְּעִיַּת 6 הַצִּפְרָדַעִים" עַל לוּחַ בֶּן שֶׁבַע מִשְׁבָּצוֹת.	<i>using Games.FrogLib7;</i>
<i>Tower</i>	מִרְחָב זֶה מְכִיל אֶת הַמַּחֲלָקָה הַגֵּרְפִיּוֹת 'מוֹט' הַנְּחוּצָה לְסִימּוּלָצִיָּה שֶׁל הָעֵבֶרֶת הַדְּסִקִּיּוֹת מִמוֹט לְמוֹט שֶׁל הַמִּשְׁחָק "מַגְדְּלֵי הַנּוּי".	<i>using Games.TowerLib;</i>

## המחלקה *Tower*

המחלקה *Tower* מגדירה 'מוט' הנחוץ לסימולציה של העברת הדסקיות ממוט למוט של המשחק "מגדלי הנוי".

### הערה:

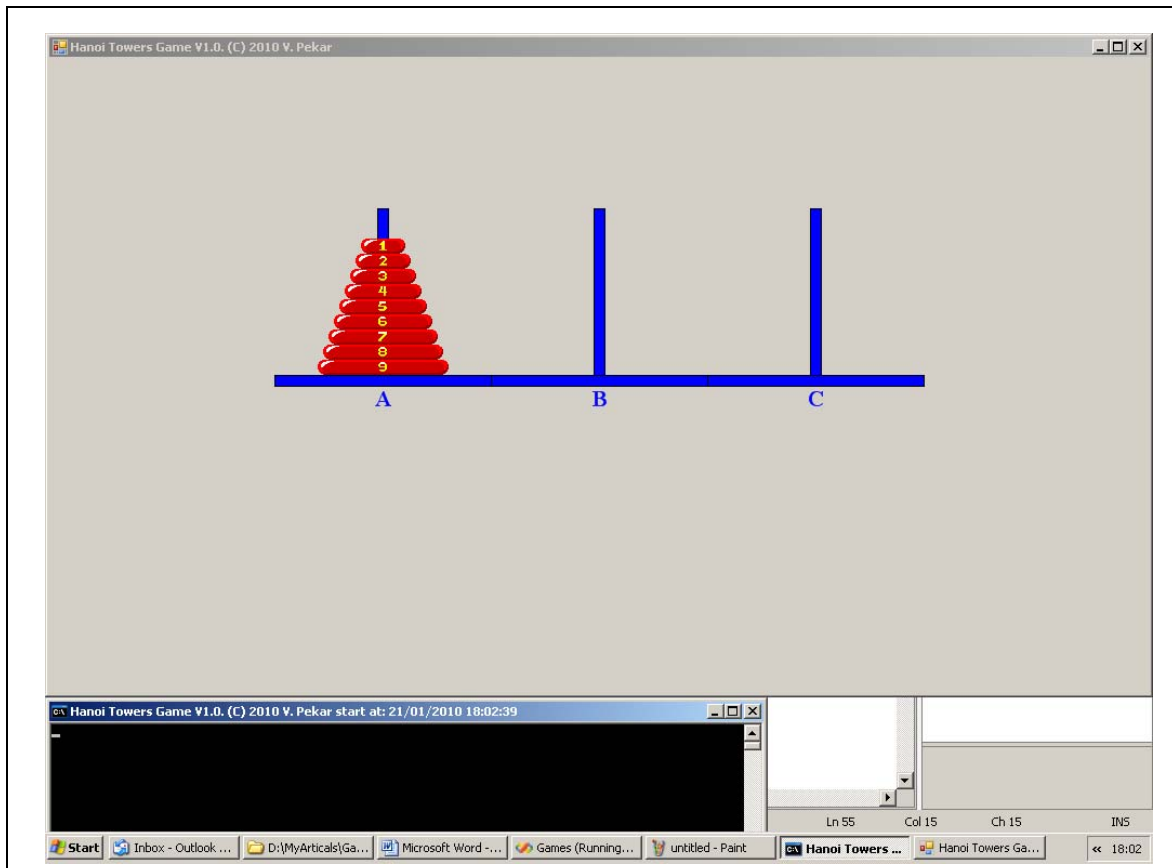
ניתן ליצור רק 5 מוטות בוזמנית.

לפניכם ממשק המחלקה:

<i>Tower(string caption)</i>	הפעולה בונה מוט ריק שכותרתו <i>caption</i> מתקבל כפרמטר.
<i>Tower(int n, string caption)</i>	הפעולה בונה מוט. על מוט מושחלות <i>n</i> דסקיות בסדר יורד ביחס להיקפן, ושכותרתו של המוט <i>caption</i> .
<i>int GetNumOfDiscs()</i>	הפעולה מחזירה את מספר הדסקיות המושחלות על המוט הנוכחי.
<i>int Top()</i>	הפעולה מחזירה את גודל הדסקית שנמצאת בראש המוט הנוכחי. אם המוט הנוכחי ריק, יוחזר הערך 0.
<i>void MoveDisc(Tower toTower)</i>	הפעולה מעבירה את הדסקית שנמצאת בראש המוט הנוכחי למוט <i>toTower</i> שהתקבל כפרמטר. הפעולה מעבירה את הדסקית רק בצורה תקינה (דסקית גדולה לא יהיה על דסקית קטנה, והמוט הנוכחי לא ריק).

יסודות מדעי המחשב עפ"י ת"ל החדשה – חומרי עזר למורה  
ניתן להשתמש בחומרים לצורך הוראה בלבד. אסור לפרסם את החומרים או לעשות בהם שימוש מסחרי כלשהו ללא קבלת אישור מראש מצוות הפיתוח

דוגמה: יצירת שלושה מגדלים שמייצגים המשחק "מגדלי הנוי"



```
Tower[] towers = new Tower[3];  
towers[0] = new Tower(10, "A");  
towers[1] = new Tower("B");  
towers[2] = new Tower("C");
```

ניתן להשתמש בחומרים לצורך הוראה בלבד.  
לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

```
using System;
using Games.TowerLib;
namespace HanoiTowersTest
{
    class Program
    {
        public static void Main()
        {
            Tower[] towers = new Tower[3];
            towers[0] = new Tower(5, "A");
            towers[1] = new Tower("B");
            towers[2] = new Tower("C");
            HanoiTowersSolution(towers, 5, 0, 2);
        }
        private static void HanoiTowersSolution(Tower[] towers, int nRings,
            int fromPole, int toPole)
        {
            int extraPole;
            if (nRings > 0)
            {
                extraPole = 3 - (fromPole + toPole);
                HanoiTowersSolution(towers, nRings - 1, fromPole, extraPole);
                towers[fromPole].MoveDisc(towers[toPole]);
                HanoiTowersSolution(towers, nRings - 1, extraPole, toPole);
            }
        }
    }
}
```

## המחלקה Die

המחלקה Die מגדירה 'קובייה' הנחוצה לסימולציה של "הטלת קובייה".

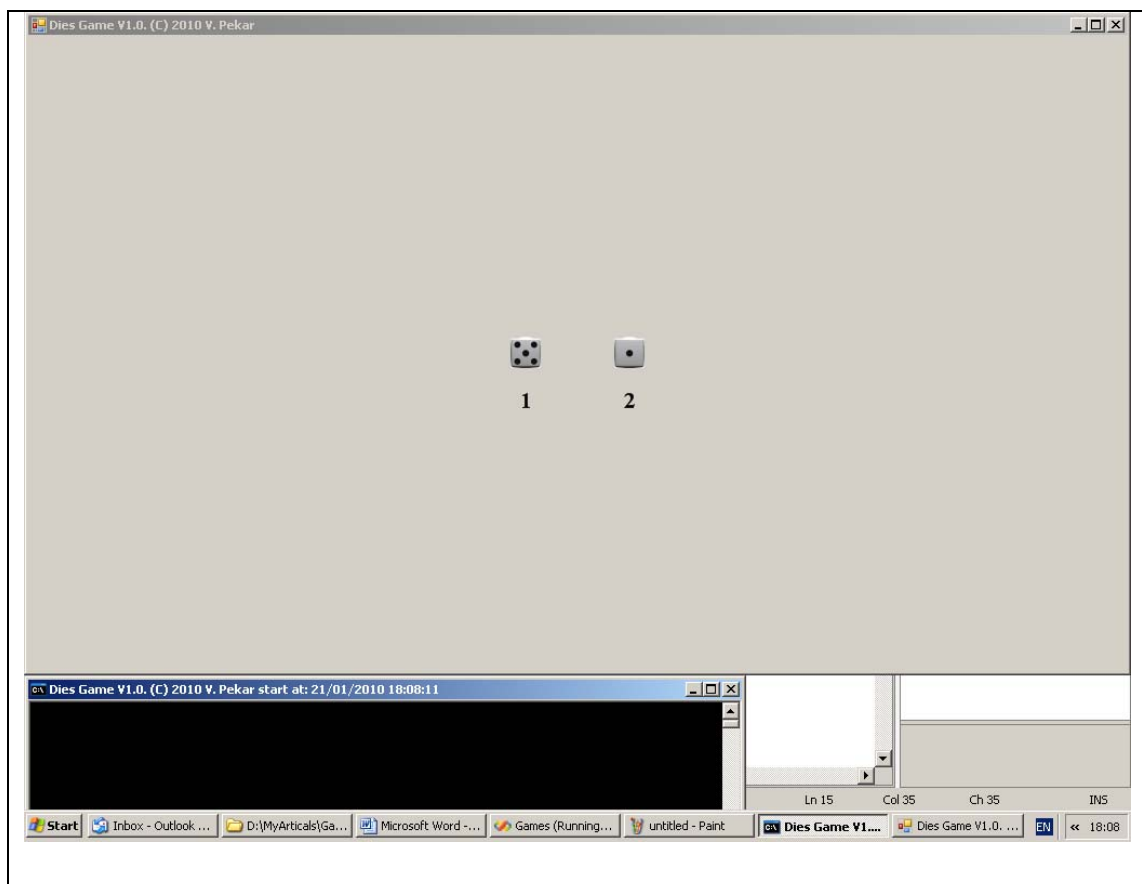
### הערה:

ניתן ליצור רק 5 קוביות בוזמנית.

לפניכם ממשק המחלקה:

<code>Die(string caption)</code>	הפעולה בונה קובייה שכותרתה <code>caption</code> , מתקבל כפרמטר. הקובייה שנוצרה מראה מספר אקראי בין 1 ל-6.
<code>void Roll()</code>	הפעולה מדמה "הטלת קובייה". בתום הפעולה מתעדכן המספר שהקובייה מראה לאחר הטלה.
<code>int GetNum()</code>	פעולה מחזירה מספר שמראה הקובייה.

דוגמה: יצירת שני קוביות



```
Die d1 = new Die("1");
```

```
Die d2 = new Die("2");
```

```
using System;
using Games.DieLib;
namespace DieTest
{
    class Program
    {
        static void Main(string[] args)
        {
            Die d1 = new Die("1");
            Die d2 = new Die("2");
            Console.ReadLine();
            int count = 0, n1, n2;
            d1.Roll();
            n1 = d1.GetNum();
            d2.Roll();
            n2 = d2.GetNum();
            while (n1 != 6 || n2 != 6)
            {
                d1.Roll();
                d2.Roll();
                n1 = d1.GetNum();
                n2 = d2.GetNum();
                count++;
            }
            Console.WriteLine(count);
        }
    }
}
```

## המחלקה Dreidle

המחלקה Dreidle מגדירה 'סביבון' הנחוצה לסימולציה של "הגרלת סביבון".

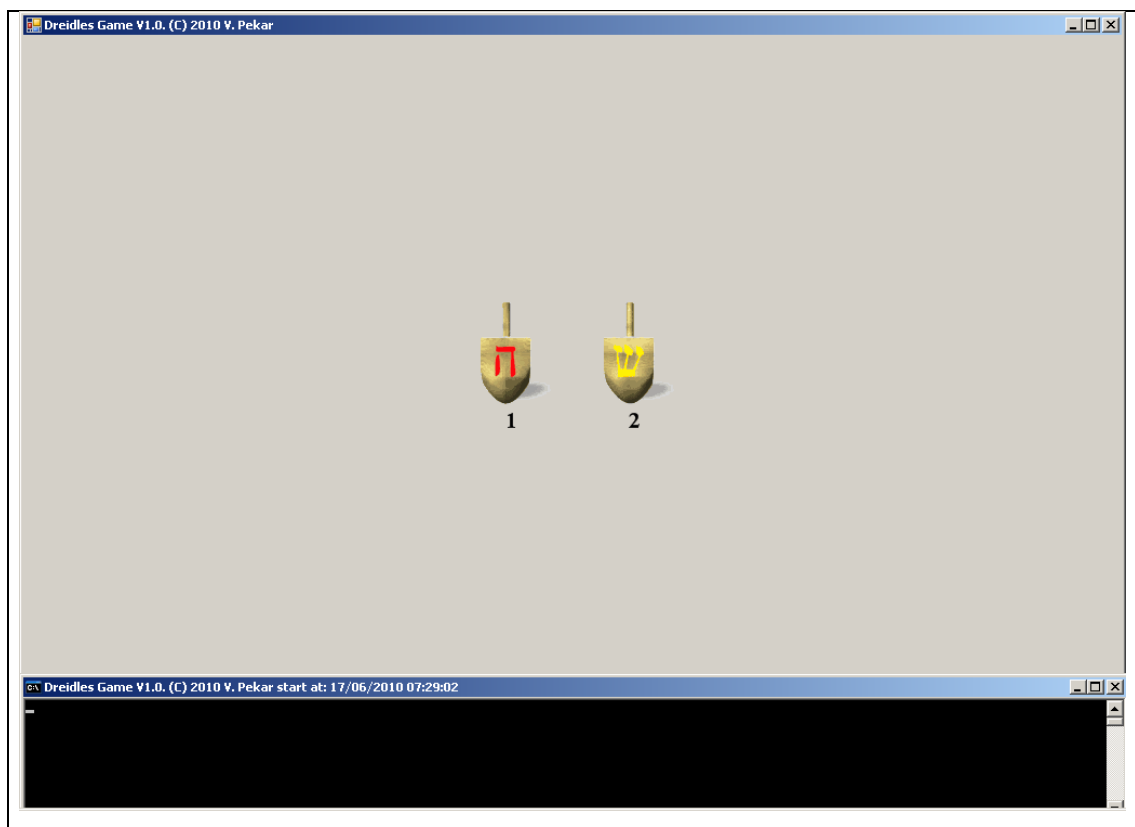
### הערה:

ניתן ליצור רק 5 סביבונים בוזמנית.

לפניכם ממשק המחלקה:

<code>Dreidle (string caption)</code>	הפעולה בונה סביבון שותרתו <code>caption</code> , מתקבל כפרמטר. הסביבון שנוצר מראה אות אקראית בין האותיות נ ג ה ש.
<code>void Roll()</code>	פעולה מדמה "הגרלת סביבון". בתום הפעולה מתעדכנת האות שהסביבון מראה לאחר הגרלה.
<code>string GetLetter()</code>	פעולה מחזירה מחרוזת שמייצגת את האות שמראה הסביבון: <code>ש="shin"</code> , <code>ה="chay"</code> , <code>ג="gimel"</code> , <code>נ="nun"</code> .

דוגמה: יצירת עצמים – שני סביבונים



```
Dreidle d1 = new Dreidle("1");
```

```
Dreidle d2 = new Dreidle("2");
```



```
using System;
using Games.DieLib;
namespace DieTest
{
    class Program
    {
        static void Main(string[] args)
        {
            Dreidle d1 = new Dreidle("1");
            Dreidle d2 = new Dreidle("2");
            int count = 0;
            string let1, let2;
            d1.Roll();
            let1 = d1.GetLetter();
            d2.Roll();
            let2 = d2.GetLetter();
            while (let1 != "shin" || let2 != "shin")
            {
                d1.Roll();
                d2.Roll();
                let1 = d1.GetLetter();
                let2 = d2.GetLetter();
                count++;
            }
            Console.WriteLine(count);
        }
    }
}
```

## המחלקות *BlueFrog* ו-*GreenFrog*

המחלקות מגדירות 'צפרדע כחולה' ו-'צפרדע ירוקה' הנחוצות לסימולציה של פיתרון "בעיית 4 הצפרדעים" על לוח בן חמש משבצות. לפניכם ממשק המחלקה:

<i>BrownFrog(int place)</i>	הפעולה בונה 'צפרדע כחולה' הממוקמת במיקום <i>place</i> , מתקבל כפרמטר. המיקום הוא מספר בין 0 ל-4. ניתן למקם את הצפרדע רק במקום פנוי.
<i>GreenFrog(int place)</i>	הפעולה בונה 'צפרדע ירוקה' הממוקמת במיקום <i>place</i> , מתקבל כפרמטר. המיקום הוא מספר בין 0 ל-4. ניתן למקם את הצפרדע רק במקום פנוי.
<i>void Move()</i>	הפעלה מזיזה 'צפרדע כחולה' או 'צפרדע ירוקה' (בכיוון המותר) למיקום סמוך פנוי. כאשר כל הצפרדעים הירוקות יודעות לזוז רק שמאלה, וכל הצפרדעים הכחולות יודעת לזוז רק ימינה.
<i>void Jump()</i>	הפעלה מקפיצה 'צפרדע כחולה' או 'צפרדע ירוקה' (בכיוון המותר). צפרדע בצע אחד יכולה לקפץ מעל צפרדע מצבע אחר אל מיקום פנוי. כאשר כל הצפרדעים הירוקות יודעות לקפוץ רק שמאלה, וכל הצפרדעים הכחולות יודעות לקפוץ רק ימינה.

יסודות מדעי המחשב עפ"י ת"ל החדשה – חומרי עזר למורה  
ניתן להשתמש בחומרים לצורך הוראה בלבד. אסור לפרסם את החומרים או לעשות בהם שימוש מסחרי כלשהו ללא קבלת אישור מראש מצוות הפיתוח

## שלב ראשון – יצירת עצמים לייצוג משחק הצפרדעים



```
BlueFrog g1 = new BrownFrog("0");
```

```
BlueFrog g2 = new BrownFrog("1");
```

```
GreenFrog g3 = new GreenFrog("3");
```

```
GreenFrog g4 = new GreenFrog("4");
```

```
using System;
using Games.FrogLib;
namespace DieTest
{
    class Program
    {
        static void Main(string[] args)
        {
            // שלב ראשון - יצירת עצמים לייצוג משחק הצפרדעים
            BlueFrog g1 = new BrownFrog("0");
            BlueFrog g2 = new BrownFrog("1");
            GreenFrog g3 = new GreenFrog("3");
            GreenFrog g4 = new GreenFrog("4");
            Console.WriteLine("To solve Groggs Press Enter !!!");
            Console.ReadLine();
            // שלב שני - ביצוע פעולות של עצמים לפתרון משחק הצפרדעים
            g2.Move();
            g3.Jump();
            g4.Move();
            g2.Jump();
            g1.Jump();
            g3.Move();
            g4.Jump();
            g1.Move();
        }
    }
}
```

## מחלקה גרפית לייצוג עץ בינרי

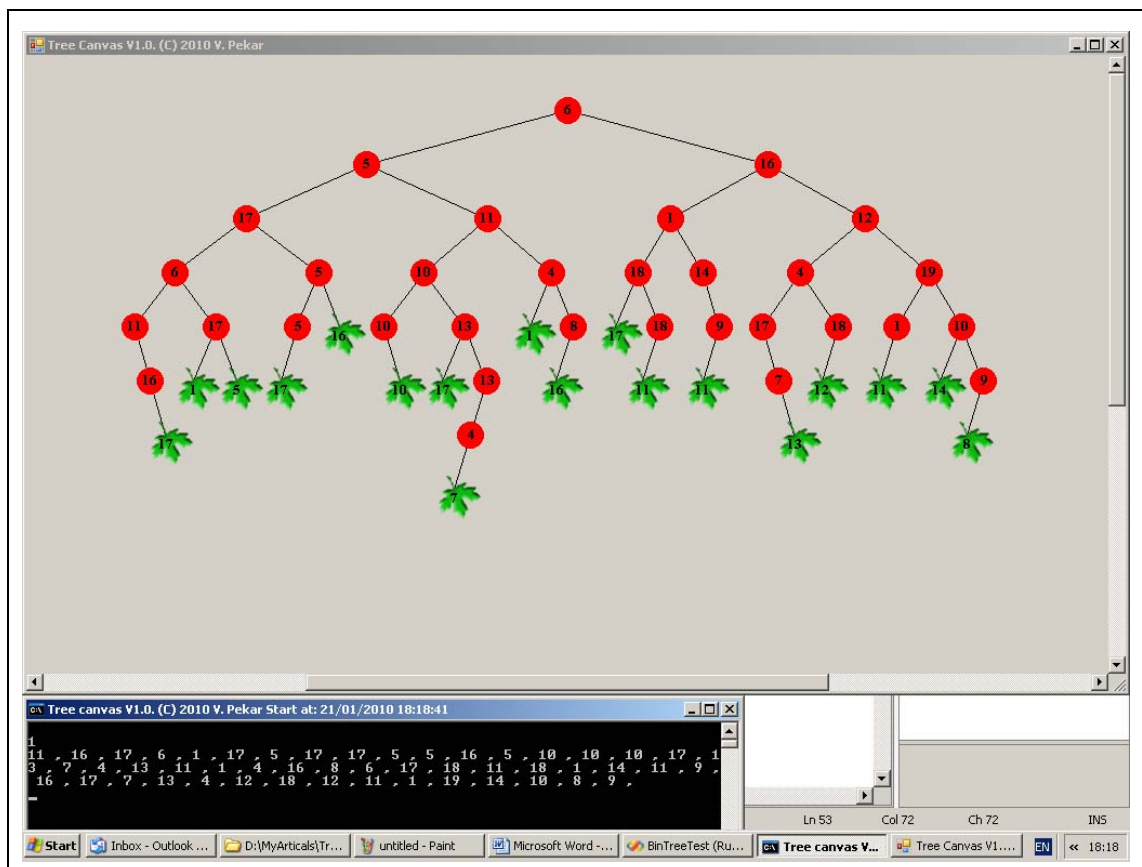
הקובץ *BinTreeCanvasLib.dll* מכיל את מחלקת השירות *TreeCanvas*. מחלקת השירות מאפשרת הצגה גרפית של עצים בינריים המבוססים על המחלקות הגנריות *BinTreeNode<T>* ו-*BinTreeUtils*, שנמצאות בתוך ספריית העזר *Unit4* "עיצוב תוכנה מבוסס עצמים בשפת C#".

```
static void TreeDraw<T>  
(BinTreeNode<T> tree)
```

הפעולה מציירת עץ בינרי גנרי *tree* המתקבל כפרמטר.

**הערה:**

מחלקת השירות *BinTreeUtils* מאפשרת להציג רק עץ בינרי אחד בזמנית.



```
BinTreeNode<int> t = BinTreeUtils.BuildRandomTree(50, 1, 3);  
TreeCanvas.TreeDraw(t);
```

דוגמה :

```
using System;
using Unit4.CollectionsLib;
using Unit4.UtillsLib;
using BinTreeCanvasLib;
namespace BinTreeTest
{
    class Program
    {
        public static void Main()
        {
            BinTreeNode<int> t = BinTreeUtils.BuildRandomTree(50, 1, 3);
            TreeCanvas.TreeDraw(t);
            Console.WriteLine(BinTreeUtils.InOrderTraversal(t));
            Console.WriteLine("To Build next Tree Press Enter");
            Console.ReadLine();
            Console.Clear();
            BinTreeNode<string> t1 = new BinTreeNode<string>("ab");
            TreeCanvas.TreeDraw(t1);
        }
    }
}
```