

המחלקה Grid

המחלקה הגרפית 'רשת'.

הערה: ניתן ליצור רק 20 רשתות בוזמנית.

<i>Grid(int rows, int columns, int cellWidth, int cellHeight)</i>	פעולה בונה רשת של <i>row</i> שורות על <i>columns</i> עמודות לפי המידות <i>cellWidth</i> ו- <i>cellHeight</i> של כל משבצת.
<i>void SetLocation(int xLocation, int yLocation)</i>	הפעולה פעולה קובעת את המיקום של הפינה השמאלית התחתונה של הרשת.
<i>void SetCellText(int row, int column, string cellText)</i>	הפעולה מציגה <i>cellText</i> במשבצת הנמצאת בשורה <i>row</i> ועמודה <i>column</i> .
<i>void SetCellColor(int row, int column, Color cellColor)</i>	הפעולה קובעת את צבע המילוי <i>cellColor</i> של משבצת הנמצאת בשורה <i>row</i> ועמודה <i>column</i> .
<i>void SetCellTextColor(int row, int column, Color cellTextColor)</i>	הפעולה קובעת את צבע הטקסט <i>cellTextColor</i> של משבצת הנמצאת בשורה <i>row</i> ועמודה <i>column</i> .
<i>void SetGridLinesColor(Color gridLinesColor)</i>	פעולה קובעת את צבע <i>gridLinesColor</i> של קווי הרשת.
<i>void SetFontSize(int fontSize)</i>	הפעולה קובעת את גודל הפונט של כל המשבצות.
<i>void SetDelay(int milliseconds)</i>	הפעולה קובעת את קצב החלפת של צבע המילוי או החלפת הטקסט, על-פי הפרמטר המועבר <i>milliseconds</i> : מספר מילי שניות (גדול מאפס) להמתנה בהזזת הצב .
<i>void FlipHorizontal(int row, int column)</i>	הפעולה גורמת להיפוך אופקי על הציר האנכי של המשבצת הנמצאת בשורה <i>row</i> ועמודה <i>column</i> .
<i>void FlipHorizontal()</i>	הפעולה גורמת להיפוך אופקי של הרשת על הציר האנכי.
<i>void FlipVertical()</i>	הפעולה גורמת להיפוך אנכי של הרשת על הציר האופקי.
<i>void FlipVertical(int row, int column)</i>	הפעולה גורמת להיפוך אנכי על הציר האופקי של המשבצת הנמצאת בשורה <i>row</i> ועמודה <i>column</i> .
<i>void Hide()</i>	הפעולה גורמת לרשת להיות מוסתרת.
<i>void Show()</i>	הפעולה גורמת לרשת להיות גלויה.
<i>int GetRowsCount()</i>	פעולה מחזירה מספר שורות המשבצות.
<i>int GetColumnsCount()</i>	פעולה מחזירה מספר עמודות המשבצות.

```

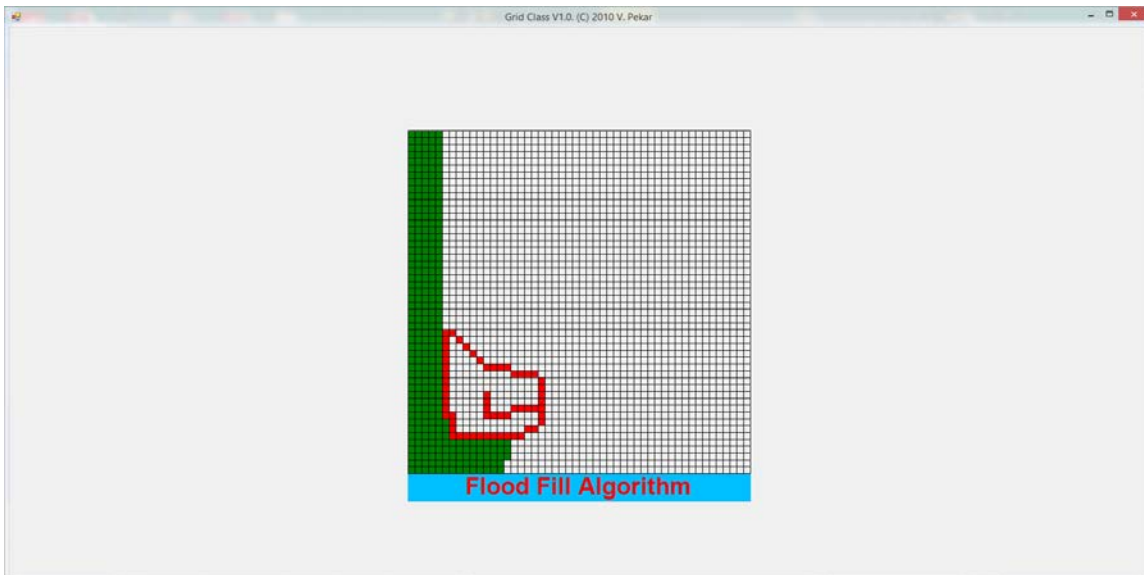
using System;
using System.Drawing;
using Unit4.GridLib;
using System.Text;
using System.Threading;
namespace TestUnit4{
    public class TestGrids{
        public static void Main(){
            int x = 0;
            int y = 0;
            Grid g = new Grid(50, 50, 10, 10);
            Grid g1 = new Grid(1, 1, 50 * 10 + 1, 40);
            g1.SetLocation(0, -50 * 10 / 2 - 21);
            g1.SetFontSize(26);
            g1.SetCellTextColor(0, 0, Color.Red);
            g1.SetGridLinesColor(Color.Empty);
            g1.SetCellColor(0, 0, Color.DeepSkyBlue);
            g1.SetCellText(0, 0, "Flood Fill Algorithm ");
            g1.Show();
            ///////////////////////////////////////////////////////////////////
            for (int i = 8; i < 21; i++)
                g.SetCellColor(i, 5, Color.Red);
            for (int i = 1; i < 6; i++)
                g.SetCellColor(21 - i, 5 + i, Color.Red);
            for (int i = 11; i < 15; i++){
                g.SetCellColor(15, i, Color.Red);
                g.SetCellColor(15 - 1, i + 4, Color.Red);
                g.SetCellColor(8 + 1, i + 4, Color.Red);
                g.SetCellColor(8, i, Color.Red);
            }
            for (int i = 14 - 7; i < 14; i++)
                g.SetCellColor(i, 19, Color.Red);
            for (int i = 5; i < 9; i++){
                g.SetCellColor(i, 6, Color.Red);
                g.SetCellColor(i + 3, 11, Color.Red);
            }
            for (int i = 7; i < 17; i++)
                g.SetCellColor(5, i, Color.Red);
            g.SetCellColor(6, 17, Color.Red);
            g.SetCellColor(6, 18,, Color.Red);
            g.Show();
            g.SetDelay(10000);
            Flood(g, Green, 1, 1);
            Flood(g, Color.Yellow, 10, 10);
        }
    }
}

```

```

}
private static void Flood(Grid g, Color fillColor, int row, int col)
{
    if (row < 0 || row >= g.GetRowCount() || col < 0 ||
        col >= g.GetColumnsCount())
        return;
    if (g.GetCellColor(row, col) != Color.Empty)
        return;
    g.SetCellColor(row, col, fillColor);
    Flood(g, fillColor, row - 1, col);
    Flood(g, fillColor, row + 1, col);
    Flood(g, fillColor, row, col - 1);
    Flood(g, fillColor, row, col + 1);
}
}
}

```



כדי שנוכל להשתמש באחת או יותר מהמחלקות שלעיל בפרויקט נוכחי צריך לצרף את הקובץ *Unit4.dll* כחלק אינטגרלי לפרויקט.

על מנת לצרף את הקובץ *Unit4.dll* לפרויקט, יש לבצע את השלבים הבאים:

- ✓ להעתיק את הקובץ *Unit4.dll* פיזי לתיקייה *bin/debug* של הפרויקט.
- ✓ להוסיף *Reference* לפרויקט לקובץ *Unit4.dll*, שנמצא בתיקייה *bin/debug*.
לחץ לחיצה ימנית עם העכבר על *References*, מתוך חלון ה-*Solution* של הפרויקט,
בחר *Add reference...*

בחלון שהתקבל בחר בכרטיסיה *Browse* ואז היכנס לתיקייה *bin/debug*, סמן את הקובץ *Unit4.dll* ולסיום לחץ על הכפתור *OK*.

- ✓ עכשיו אפשר להשתמש בכל המחלקות שקיימות בקובץ *Unit4.dll*. לדוגמה, בהצהרה על העצם מהמחלקה *Stack* יצירת אותו, יש להשתמש בשם המחלקה המלא:

```
Unit4.CollectionsLib.Stack<int> st =  
new Unit4.CollectionsLib.Stack<int>();
```

שם המחלקה המלא של המחלקה מעיד על כך שהמחלקה *Stack*, שייכת למרחב השמות של *Unit4.CollectionsLib* (הכוללת מחלקות שמופיעות ביחידת לימוד 4).

- ✓ כדי שנוכל לכתוב את שם המחלקה בצורה המקוצרת (המרומזת), יש להצהיר על מרחב השמות בראש התוכנית:

```
using Unit4.CollectionsLib;
```

לאחר הצהרה על מרחב השמות, אין צורך לשייך למרחב השמות למחלקה וניתן לכתוב:

```
Stack<int> st = new Stack<int>();
```