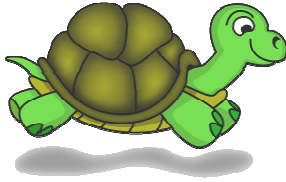


חומרים שהוכנו על-ידי מורי הניסוי תש"ע להוראת "יסודות מדעי המחשב"

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

לרקוד עם צבים: דף עבודה לשילוב עצמים ולולאות

כתיבה ועריכה:
שירלי רוזנברג-כהן



ארכוק? עם צבים

כאשר ציירנו **ריבוע** בעזרת המחלקה Turtle (ב- unit4.dll) השתמשנו בהוראות הבאות:

```
Turtle t1 = new Turtle();  
t1.TailDown();  
t1.MoveForward(100);  
t1.TurnRight(90);  
t1.MoveForward(100);  
t1.TurnRight(90);  
t1.MoveForward(100);  
t1.TurnRight(90);  
t1.MoveForward(100);  
t1.TurnRight(90);
```

נאתר את ההוראות שחוזרות על עצמן:

```
t1.MoveForward(100);  
t1.TurnRight(90);
```

במקום **לכתוב** ההוראות אלה 4 פעמים נוכל ליצור לולאה שתעבוד 4 פעמים:

```
Turtle t1 = new Turtle();  
t1.TailDown();  
for (int i = 1; i <= 4; i++)  
{  
    t1.MoveForward(100);  
    t1.TurnRight(90);  
}
```

תרגיל 1:

מה יהיו ההוראות ליצירת **משושה**?

מה יהיו ההוראות ליצירת **משולש**?

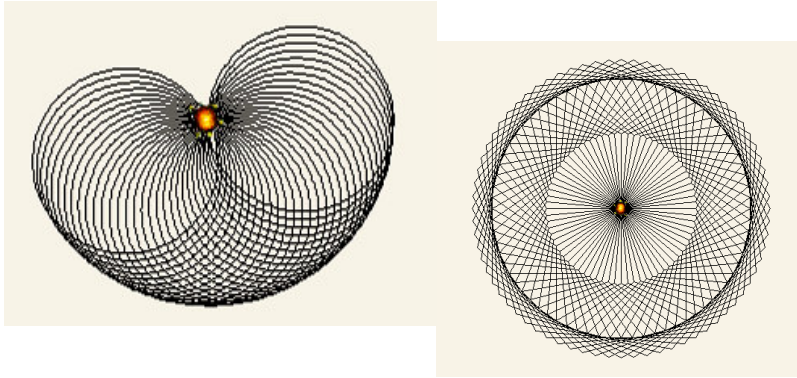
מה יהיו ההוראות ליצירת **עיגול**?

אם נרצה ליצור ציור יותר "דקורטיבי" : לאחר ציור צורה אחת – הצב יפנה ב-x מעלות ימינה. ויבצע את ציור הצורה פעמים רבות.

שים ♥ : כדאי להגביר את מהירות הצב באמצעות הפקודה `t1.SetDelay(10)`; (או מספר קטן יותר) אותה נוסף מיד לאחר יצירת הצב.

תרגיל 2:

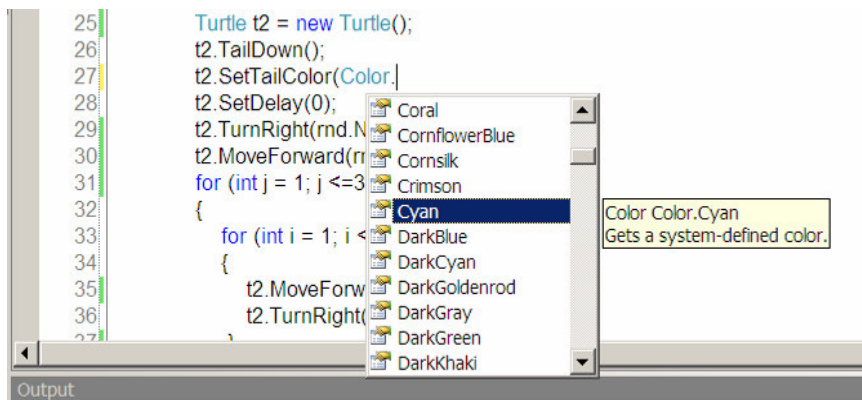
צייר ציור "דקורטיבי" למשל :



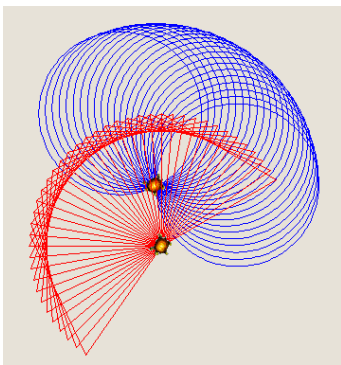
אפשר לשנות את צבע זנב הצב. יש צורך להוסיף את המחלקה `System.Drawing` ל- `references` כפי שעשינו עם המחלקה `unit4.dll`.

ההוראה לצביעת זנב הצב לכחול למשל : `t2.SetTailColor(Color.Blue)`;

חלק מהצבעים האפשריים : Red, Green, Orange, Pink עוד צבעים רבים אחרים אותם תוכל לגלות כך :



שים ♥ : לכל צב יכול להיות מוגדר צבע זנב אחר. אך עקב מגבלות התכנה ניתן ליצור צב עם צבע זנב אחר רק לאחר שהצב הקודם סיים את עבודתו. כלומר : שני צבים לא יכולים לרוץ במקביל בצבעים שונים.



תרגיל 3 :

כתוב תכנית שתצייר ציור "דקורטיבי" המורכב משתי צורות דקורטיביות בצבעים שונים. למשל:

דף למתקדמים

זוכרים שב"קארל" יכולנו לשפר את הרובוט... אז גם כאן.

בתפריט project נבחר ב-add class . בחלון שיפתח ניתן למחלקה שם MyTurtle. (או במקום My שם אחר) המחלקה החדשה תראה כך :

```
using System;  
using Unit4.TurtleLib;  
namespace TestTurtle
```

```
{  
    class MyTurtle:Turtle  
    {  
        public MyTurtle(int delay)  
        {  
            this.TailDown();  
            this.SetDelay(delay);  
        }  
        public void Shape(int sides, int length)  
        {  
            for (int i = 1; i <= sides; i++)  
            {  
                this.MoveForward(length);  
                this.TurnRight(360/sides);  
            }  
        }  
    }  
}
```

ניצור צב חדש היורש את תכונותיו ופעולותיו מהצב המקורי

ניצור פעולה בונה המקבלת בפרמטר את זמן ההשהיה (בד"כ ע"מ להקטינו). ננצל את ההזדמנות להוריד את הזנב.

נכתוב פעולה היוצרת צורה סגורה כלשהי. הפעולה מקבלת את מספר הצלעות ואורך צלע אחת.

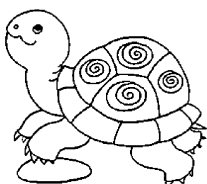
התכנית המנצלת את הצב המשופר ויוצרת צורה דקורטיבית מרובת ריבועים תראה כך :

```
using System;  
using Unit4.TurtleLib;  
using System.Drawing;  
namespace TestTurtle
```

```
{  
    class NewTurtle  
    {  
        static void Main(string[] args)  
        {  
            MyTurtle t1 = new MyTurtle(1);  
            for (int j = 1; j <= 30; j++)  
            {  
                t1.Shape(4, 100);  
                t1.TurnLeft(2);  
            }  
        }  
    }  
}
```

בהמשך נראה פעולות נוספות שניתן לכתוב על מנת לשפר את הצב...

Turtle



תרגיל 4:

ננצל את ידיעותינו במספרים אקראיים על מנת ליצור "שומר מסך" מפתיע. חשוב היכן תוכל לשבץ מספרים אקראיים.