

שימוש בגרפיקה להוראת לולאות

גרסת סי שרפ לתלמיד

ברוח דפי המעבדה של ד"ר תמר פז (בגישת "המעבדה תחילה")

צוות הפיתוח:

לודמר רחל

על-דור שפירא עירית

קונצמן זיוה

רוזנברג כהן שירלי

חזרה קבועה: לולאת מונה

הוראת החזרה for מאפשרת לחזור על ביצוע גוף הלולאה מספר ידוע מראש של פעמים

משימה 1 - חלק א:

פתח פרוייקט ב- windows application.

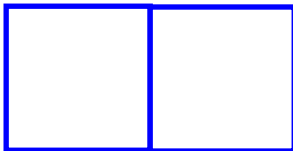
הקלד את הקטע הבא בתוך הקובץ Form1.cs בפעולה `public Form1()`.
הקטע הבא מצייר ריבוע.

```
InitializeComponent();  
Graphics dc = this.CreateGraphics();  
this.Show();  
Pen BluePen = new Pen(Color.Blue, 3);  
dc.DrawRectangle(BluePen, 0, 0, 50, 50);
```

הקלד את הקטע, הרץ ובדוק:

כיצד משנים את מיקום ההתחלה של הריבוע בחלון? _____

כיצד משנים את גודל הריבוע? _____



משימה 1 - חלק ב:

הוסף קטע המצייר ריבוע נוסף הצמוד לריבוע הקודם.

מה שינית כדי שהריבוע יזוז ימינה? _____

אם נרצה לצייר 5 ריבועים צמודים, בזה אחר זה, נצטרך להעתיק את השורה שוב ושוב בשינוי שהצעת.

יש דרך פשוטה יותר לבצע זאת – לולאה:

```
public Form1()  
{  
    InitializeComponent();  
    Graphics dc = this.CreateGraphics();  
    this.Show();  
    Pen RedPen = new Pen(Color.Red, 2);  
    int place = 0;  
    for (int i = 1; i<=5; i++)  
    {  
        dc.DrawRectangle(RedPen, place, 0, 50, 50);  
        place = place + 50;  
    }  
}
```

ראש הלולאה

גוף הלולאה

הוראת החזרה הבאה

```
for (int i = 1; i<=5; i++)
```

```
{
    dc.DrawRectangle(RedPen, place, 0, 50, 50);
    place = place + 50;
}
```

מתבצעת כך:

1. הערך 1 מוצב במשתנה הלולאה i.
2. מתבצע גוף הלולאה:
ציור של ריבוע בעט אדום במיקום $x=place, y=0$, בגודל 50×50 .
place מתעדכן ל- 50 קדימה, כהכנה לקראת הריבוע הבא.
3. משתנה הלולאה i מתעדכן וגדל ב- 1.
4. אם $i \leq 5$ חוזרים לבצע את סעיף 2. אחרת, מסתיים ביצוע הלולאה.

משימה 1 – חלק ג:

הריצו את התכנית פעמים נוספות. לפני כל הרצה שנו את כותרת הלולאה בהתאם לטבלה:

מספר הפעמים שהלולאה התבצעה	גבולות הלולאה
	<code>for (int i = 0; i<5; i++)</code>
	<code>for (int i = 1; i<7; i++)</code>
	<code>for (int i = -5; i<=0; i++)</code>
	<code>for (int i = 5; i>0; i--)</code>
	<code>for (int i = 10; i<=17; i++)</code>

משימה 1 – חלק ד:

נוכל לקצר את גוף הלולאה אם נשתמש במשתנה i במקום place:

הפעם, i לא ימנה את מספר הריבועים, אלא ייצג את המיקומים של הריבועים על ציר x.

מלא את הקטעים החסרים, הרץ ובדוק.

```
for (int i = 0; i<=____; i____)
{
    Square c=new Square (____, 0, 50, "red");
    dc.DrawRectangle(RedPen, _____, 0, 50, 50);
}
```

משימה 2:

לפניך הוראה משרטטת מעגל כחול:

```
Graphics dc = this.CreateGraphics();
Pen bluePen = new Pen(Color.Blue, 3);
dc.DrawEllipse(bluePen, 0, 0, 50, 50);
```

א. הוסף הוראות מתאימות לשרטוט 10 מעגלים.

ב. שנה את ההוראות כך שמיקום המעגלים (עבור כל עיגול: x,y) יקלט מהמשתמש.

ג. שנה את ההוראות כך שמיקום המעגלים ייווצר באופן אקראי.

ד. נתונה הפעולה הבאה, אשר מקבלת מספר שלם ומחזירה את צבע העט.

```
public Pen PenColor(int numColor)
{
    switch (numColor)
    {
        case 1: return new Pen(Color.Blue, 3);
        case 2: return new Pen(Color.Red, 3);
        case 3: return new Pen(Color.Yellow, 3);
        default: return new Pen(Color.Green, 3);
    }
}
```

ה. הקלד את הפעולה במחלקה form1 לאחר הקטע הבא:

```
public Form1()
{
    InitializeComponent();
}
```

ו. השתמש בפעולה כדי לשרטט 5 מעגליים אשר צבעם ישתנה באופן אקראי.

משימה 3:

לפניך סדרת הוראות לציור מדרגות יורדות

```
Graphics dc = this.CreateGraphics();
Pen RedPen = new Pen(Color.Red, 3);
dc.DrawLine(RedPen, new Point(10, 10), new Point(10, 20));
dc.DrawLine(RedPen, new Point(10, 20), new Point(20, 20));
dc.DrawLine(RedPen, new Point(20, 20), new Point(20, 30));
dc.DrawLine(RedPen, new Point(20, 30), new Point(30, 30));
dc.DrawLine(RedPen, new Point(30, 30), new Point(30, 40));
dc.DrawLine(RedPen, new Point(30, 40), new Point(40, 40));
dc.DrawLine(RedPen, new Point(40, 40), new Point(40, 50));
dc.DrawLine(RedPen, new Point(40, 50), new Point(50, 50));
dc.DrawLine(RedPen, new Point(50, 50), new Point(50, 60));
```

עליך לבנות לולאת for מתאימה שתצייר את המדרגות עם מספר מינימלי של הוראות בגוף הלולאה.

בעמוד הבא נתון שלד הלולאה. השלם את החסר.

```

Graphics dc = this.CreateGraphics();
Pen RedPen = new Pen(Color.Red, 3);
int x=10, y=10;
for (int i=1; i<=___; i++)
{
    dc.DrawLine(RedPen, new Point(x, y), new Point(____, ____));
    dc.DrawLine(RedPen, new Point(x, y+10), new Point(____, ____));
    x=____; y=____;
}

```

השלם את החסר

משימה 4 :

הפעולה הבאה יוצרת עיגול מלא.

בציור המעגל (המסגרת) משתמשים ב- Draw. בציור העיגול (המלא) משתמשים ב- Fill

בציור המעגל (המסגרת) משתמשים בעט Pen. בציור העיגול (המלא) משתמשים במברשת - Brush

```
Graphics dc = this.CreateGraphics();
```

```
Brush RedBrush = new SolidBrush(Color.Red);
```

```
dc.FillEllipse(RedBrush, x, y, width, height);
```

שנה את התכנית במשימה 2

כך שתצייר עיגולים מלאים במקומות וגדלים אקראיים.

משימה 5 :

כתוב תוכנית היוצרת לוח קליעה למטרה – המורכב מ- 10 עיגולים, אחד בתוך השני, בצבעים מתחלפים

לסירוגין : שחור, צהוב, שחור,

רדיוסי המעגלים הולכים וקטנים בגודל שווה.

דוגמא לפלט המשימה :

**משימה 6 : משולש שרפינסקי**

לפניך אלגוריתם מילולי, עליך לתרגם את האלגוריתם לשפת C# והרץ :

1. הצב שיעורי קורדינטות (x,y) לנקודה 1 – (x1,y1) מרכז השורה העליונה במסך
2. הצב שיעורי קורדינטות (x,y) לנקודה 2 – (x2,y2) פינה ימנית תחתונה של המסך
3. הצב שיעורי קורדינטות (x,y) לנקודה 3 – (x3,y3) פינה שמאלית תחתונה של המסך
4. הצב ב pointX שיעור X של נקודה אקראית על המסך
5. הצב ב pointY שיעור Y של נקודה אקראית על המסך
6. בחר מספר אקראי בין 1-3
7. חזור 10000 פעמים על הפעולות הבאות :
 - 7.1 צייר על המסך את הנקודה pointX, pointY
 - 7.2 בחר מספר אקראי בין 1-3
 - 7.3 חשב ערך pointX מחדש : ממוצע של pointX עם שיעור ה- X של הנקודה שנבחרה באקראי
 - 7.4 חשב ערך pointY מחדש : ממוצע של pointY עם שיעור ה- Y של הנקודה שנבחרה באקראי

חזרה מותנית: לולאת While

הוראת החזרה while מאפשרת לחזור על ביצוע הוראה (או סדרת הוראות) כל עוד תנאי מסוים מתקיים.

כפי שראינו, הוראת for מורכבת משלושה חלקים:

```
for ( int i=1; i<10; i++)
```

int i=1 אתחול משתנה הלולאה:

i<10 תנאי כניסה ללולאה:

i++ קידום משתנה הלולאה:

נוכל לתרגם את לולאת ה- for ישירות ללולאת while באופן הבא:

```
int i=1;
while ( i<10 )
{
    הוראות לביצוע
    i++;
}
```

אך, לשם חזרה קבועה אין צורך בלולאת while.

הצורך העיקרי של לולאה מסוג זה הוא כאשר מספר החזרות אינו ידוע מראש ותלוי באירוע מסוים המתרחש במהלך הרצת התכנית.

לדוגמה: נקלוט מספרים ונדפיס את ריבועיהם עד אשר יקלט המספר 0.

```
Int num = int.Parse(Console.ReadLine());
While (num > 0)
{
    Console.WriteLine(num*num);
    num = int.Parse(Console.ReadLine());
}
```

משימה 7 :

לפניך קטע תכנית הקולט מהמשתמש את נקודות ההתחלה של ריבוע ומצייר אותו, כל עוד לא נקלט מספר שלילי בנתון הראשון. השלם את הקטעים החסרים.

```
int x,y;
Console.WriteLine(" enter x,y for ellipse: ");
x = int.Parse(Console.ReadLine());
while ( _____ )
{
    y = int.Parse(Console.ReadLine());
    y = Math.abs(y);
    dc.DrawEllipse(bluePen, x, y, 50, 50);
    x = _____;
}
```

משימה 8:

כדי לצייר קו באורך אקראי בתחום 100-300, ניתן את ההוראה הבאה:

```
dc.DrawLine(BluePen, new Point (new Random().Next(1, 100),
                                new Random().Next(1, 100)),
            new Point (new Random().Next(100, 300),
                                new Random().Next(100, 300)));
```

כתוב תכנית המציירת קו באורך אקראי על הצג.

על המשתמש לנחש את אורך הקו. אם הניחוש שלו גדול מידי או קטן מידי תוצג הוראה מתאימה.

התכנית תעצר כאשר המשתמש ינחש את אורכו המדויק של הקו.

הערה: אורך הקו מחושב ע"י הנוסחה: $d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$

משימה 9:

כדי לצייר עיגול מלא ניתן את ההוראות הבאות:

```
Graphics dc = this.CreateGraphics();
Brush RedBrush = new SolidBrush (Color.Red);
dc.FillEllipse (RedBrush, 50, 50, 10, 10);
```

כתוב תוכנית היוצרת תנועה של עיגול (כדור) על הצג עד אשר יגיע למסגרת.

על התנועה להתקדם בצורת זיגזג: באלכסון כלפי מעלה ומטה לסירוגין.

במידה וברצונך למחוק את הכדור הקודם – צבע אותו בלבן.

התייחס לשני המקרים הבאים:



• גודל הזיגזג קבוע.

• גודל הזיגזג אקראי.

הערה: כדי לראות תנועה על הצג יש להשתמש בפקודת השהייה

```
System.Threading.Thread.Sleep(40);
```

לולאות מקוננות

משימה 10:

לפניך קטע תכנית המצייר שורה אחת של לוח שחמט.

```
int x=0, y=0, side=40;
for (int i= 1; i<=8; i++)
{
    if (i %2 ==0)
        color = color.Black;
    else
        color = color.White;

    Brush Brush = new SolidColorBrush(color);
    dc.FillRectangle(Brush, X, Y, side, side);
}
```

הוסף הוראות ליצירת 8 שורות של לוח שחמט.

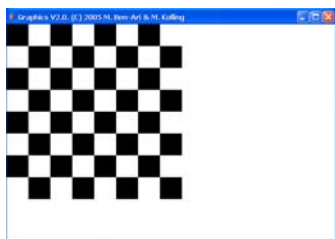
שים לב: השורות האי זוגיות יתחילו בריבוע שחור, ואילו השורות הזוגיות יתחילו בריבוע לבן.

היעזר בשלד הלולאה הבאה והשלם את החסר.

שים לב: המשבצות השחורות צריכות להופיע לסירוגין. (רמז: בדוק את סכום מספר השורה ומספר העמודה במשבצות השחורות).

```
for (_____)
{
    for (int j=1; j<=8; j++)
    {
        if ((_____)%2==0)
            color = color.Black;
        else
            color = color.White;
        Brush Brush = new SolidColorBrush(color);
        dc.FillRectangle(Brush, x, y, side, side);
        x=x+side;
    }
    x=____;
    y=____;
}
```

מה הקשר בין מספר השורה ומספר
העמודה לבין צבע המשבצת?



שמור והרץ.

דוגמת פלט:

משימה 11:

מטרת המשימה היא לייצר את "לוח הכפל" על טבלה בצבע צהוב בגודל 10 על 10.

א. הקלד את הקטע הבא:

```
InitializeComponent();
Graphics dc = this.CreateGraphics();
this.Show();
Pen BlackPen = new Pen(Color.Black, 2);
Brush YellowBrush = new SolidBrush(Color.Yellow);
dc.DrawRectangle(BlackPen, 0, 0, 250, 250);
dc.FillRectangle(YellowBrush, 2, 2, 248, 248);
```

הקש את גודל הריבוע של חלון הצויר.

שמור והרץ. מה הפלט שקיבלת? _____

ב. השלב הבא במשימה הוא יצירת קווי אורך ורוחב בגודל אחיד על הריבוע שהתקבל.

הוסף את ההוראות הבאות בסוף התוכנית והשלם את ההוראות החסרות:

```
int x=0, y=0; // נקודת איתחול של הקווים
int d=_____;
```

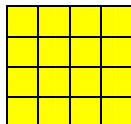
חשב את רוחב המשבצת בטבלה.

ג-1. השלם את החסר לשרטוט קווים אופקיים לרוחב הטבלה, כאשר המרחק בין כל שני קווים הוא d.

```
for (int i=0; i<=11; i++)
{
    dc.DrawLine (BlackPen, new Point(0,y), new Point(____, ____));
    y=_____;
```

ג-2. השלם את החסר לשרטוט קווים אנכיים לאורך הטבלה, כאשר המרחק בין כל שני קווים הוא d.

```
x=0;
for (int j=0; j<=11; j++)
{
    dc.DrawLine(BlackPen, new Point(x,0), new Point(____, ____));
    x=_____;
```



ד. שמור והרץ. בדוק שאכן קבלת טבלה צהובה עם קווים לאורך ולרוחב הטבלה.

ה. כעת נדפיס את ערכי "לוח הכפל", כל ערך יירשם באמצע תא המשבצת.

הדפסת טקסט על גבי הטבלה תעשה ע"י הפקודות הבאות:

- הגדרת גופן הטקסט: `Font f = new Font("Ariel", 5);`
- המחרוזת s היא בעצם מחרוזת המייצגת את הערך המספרי של המכפלה `string s = "" + i * j`
- כתיבת הטקסט במשבצת: `dc.DrawString(s, f, Brushes.Black, x, y);` כאשר x,y הן הקואורדינטות של אמצע המשבצת.