

שיעורי בית יסודות מחלקת עיגולים – אופיר הופמן י3

```
class Circle
{
    Random rnd = new Random();

    private double radius;
    private ConsoleColor color;
    private bool isFull;
    private bool isBenGvirSupporter;

    // בנאי
    public Circle(double r, ConsoleColor c)
    {
        this.radius = r;
        this.color = c;
    }

    // בנאי עם קביעה כמלא
    public Circle(double r, ConsoleColor c, bool isItFull)
    {
        this.radius = r;
        this.color = c;
        this.isFull = isItFull;
    }

    // בנאי - עם תכונה האם תומך בבן גביר
    public Circle(double r, ConsoleColor c, bool isItFull, bool SupportsBenGvir)
    {
        this.radius = r;
        this.color = c;
        this.isFull = isItFull;
        this.isBenGvirSupporter = SupportsBenGvir;
    }

    // פעולה קובעת ממלא עיגול
    public void Fill()
    {
        isFull = true;
    }

    // פעולה קובעת עיגול ריק
    public void Empty()
    {
        isFull = false;
    }

    // פעולה מאחזרת - האם עיגול מלא
    public bool IsFull()
    {
        return isFull;
    }

    public void Print()
    {
        Console.Write("I am a ");
        Console.ForegroundColor = color;
        Console.Write(color);
        Console.ForegroundColor = ConsoleColor.White;
        Console.WriteLine(" Circle with radius " + radius);
        Console.WriteLine("Is full = " + isFull);
        Console.WriteLine("Circle is an Itamar Ban Gvir supporter: " + isBenGvirSupporter);
    }
}
```

```

// פעולה קובעת רדיוס
public void SetRadius(double r)
{
    radius = r;
}

// פעולה מאחזרת רדיוס
public double GetRadius()
{
    return radius;
}

// פעולה קובעת צבע
public void SetColor(ConsoleColor c)
{
    color = c;
}

// פעולה מאחזרת צבע
public ConsoleColor GetColor()
{
    return color;
}

// פעולה מגדילה רדיוס
public void Radius_add()
{
    radius++;
}

// פעולה מקטינה רדיוס
public void Radius_sub()
{
    radius--;
}

// פעולה מאחזרת שטח
public double GetArea()
{
    return (Math.PI * radius * radius);
}

// פעולה מאחזרת היקף
public double GetPerimeter()
{
    return 2 * Math.PI * radius;
}

// פעולה מאחזרת רדיוס רנדומלי
public Circle(ConsoleColor c)
{
    radius = rnd.Next(1,11);
    color = c;
}

// פעולה קובעת האם העיגול תומך בבן גביר
public void SetBenGvirSupporter(bool b)
{
    isBenGvirSupporter = b;
}

// פעולה מאחזרת האם העיגול תומך בבן גביר
public bool GetOpinionAboutBenGvir()
{
    return isBenGvirSupporter;
}

```

```

// פעולת אתגר - ציור עיגול
public void Draw()
{
    Console.SetBufferSize(201, 201);
    Console.CursorVisible = false;
    Console.Clear();

    for (double angle = 0; angle < 2 * Math.PI; angle += Math.PI / 180)
    {
        int x = (int)Math.Round(this.radius * Math.Cos(angle) + this.radius);
        int y = (int)(0.5 * Math.Round(this.radius * Math.Sin(angle) + this.radius));
        Console.SetCursorPosition(x, y);
        for (int i = 0; i < Math.Abs(this.radius * Math.Sin(angle)); i++)
        {
            Console.SetCursorPosition((int)x, (int)(y - (i *
Math.Sign(Math.Sin(angle)))));
            Console.Write("■");
        }
        Console.WriteLine("■");
    }

    Console.SetCursorPosition(0, (int)this.radius + 1);
}

}

static void Main(string[] args)
{
    Random rnd = new Random();
    Circle circle1 = new Circle(7.5, ConsoleColor.Blue); // Creates small blue Circle
    circle1.Print(); // Prints Circle1
    Circle circle2 = new Circle(80, ConsoleColor.Red); // Creates big red Circle
    circle2.Print(); // Prints Circle2

    Circle circle3 = new Circle(rnd.Next(1, 11), ConsoleColor.Green);
    circle3.Print();
    double radius = circle3.GetRadius(); // get the radius of circle 3
    Circle circle4 = new Circle(radius + 1, ConsoleColor.DarkRed);
    circle4.Print();
    Circle circle5 = new Circle(10, ConsoleColor.Magenta);
    circle5.Print();
    Circle circle6 = new Circle(3, ConsoleColor.Green);
    circle6.Print();

    circle5.SetColor(ConsoleColor.DarkRed);
    circle5.Print();

    circle6.SetRadius(4.5);
    circle6.Print();

    Console.WriteLine("enter circle 7 radius");
    int radius7 = int.Parse(Console.ReadLine());
    Circle circle7 = new Circle(radius7, ConsoleColor.Yellow);
    circle7.Print();

    int radius8 = rnd.Next(1, 11);
    Circle circle8 = new Circle(radius8, ConsoleColor.Black);
    circle8.Print();

    circle6.Radius_add();
    circle6.Print();
    circle7.Radius_add();
    circle7.Print();

    circle6.Radius_sub();
    circle6.Print();
    circle7.Radius_sub();
    circle7.Print();
}

```

```
Console.WriteLine(circle6.GetArea());

Console.WriteLine(circle7.GetPerimeter());

Circle circle9 = new Circle(ConsoleColor.Red);
circle9.Print();

Circle circle10 = new Circle(ConsoleColor.Magenta);
circle10.Print();

Circle circle11 = new Circle(5, ConsoleColor.Blue);
circle11.Print();

Circle circle12 = new Circle(5, ConsoleColor.Yellow, true, true);
circle12.Print();

Circle circle13 = new Circle(15, ConsoleColor.DarkRed, true, false);
circle13.Print();

circle13.Draw();
}
```