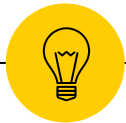




OO APPLICATION DEVELOPMENT – LES01a: OBJECTEN

ROGIER VAN DER LINDE

Enkele nuttige klassen





Math klasse

Enkele voorbeelden van de `Math` klasse:

```
double getalA = 3.1415;
double getalB = 25;

// afrondingen
int getal1 = Math.Ceiling(getalA); // 4 (naar boven afronden)
int getal2 = Math.Floor(getalA); // 3 (naar beneden afronden)
int getal3 = Math.Round(getalA, 2); // 3.14 (afronden tot op 2 cijfers)

// berekeningen
double getal4 = Math.Sqrt(getalB); // 5 (vierkantswortel)
double getal5 = Math.Max(getalA, getalB); // 25 (grootste)

// constanten
double getal6 = Math.PI; // 3.1415926535897931
```

- volledige lijst zie <https://docs.microsoft.com/en-us/dotnet/api/system.math?view=netframework-4.8>



Random klasse

Voor de oefeningen is het soms leuk om met willekeurig gegenereerde waarden te kunnen werken. Dat kan met de `Random` klasse:

```
// maak random generator
Random rnd = new Random();

// genereer random getallen
Console.WriteLine("Random getal tussen 1 en 20: {0}", rnd.Next(1, 21));
Console.WriteLine("Random getal tussen 1 en 20: {0}", rnd.Next(1, 21));
Console.WriteLine("Random getal tussen 1 en 20: {0}", rnd.Next(1, 21));
Console.WriteLine("Random getal tussen 1 en 20: {0}", rnd.Next(1, 21));
Console.WriteLine("Random getal tussen 1 en 20: {0}", rnd.Next(1, 21));

Console.WriteLine("Random getal tussen 1 en 10: {0}", rnd.Next(1, 11));
Console.WriteLine("Random getal tussen 1 en 5: {0}", rnd.Next(1, 6));
```

```
C:\Users\sam.vanbuggenhout...
Random getal tussen 1 en 20: 2
Random getal tussen 1 en 20: 14
Random getal tussen 1 en 20: 9
Random getal tussen 1 en 20: 8
Random getal tussen 1 en 20: 12
Random getal tussen 1 en 10: 6
Random getal tussen 1 en 5: 3
```



DispatcherTimer / TimeSpan klasse

Deze klasse(n) kan je gebruiken om een timer te maken die op vaste tijden een methode uitvoert. Eenvoudig voorbeeld (XAML):

```
...
Title="DispatcherTimer demo" Height="184.478" Width="313.743">
<Grid Margin="0,0,0,0">
  <Label x:Name="lblCount" Content="" HorizontalContentAlignment="Right" HorizontalAlignment="Left"
    Height="103" Margin="132,24,0,0" VerticalAlignment="Top" Width="151" FontSize="48"/>
</Grid>
</Window>
```



DispatcherTimer / TimeSpan klasse

Deze klasse(n) kan je gebruiken om een timer te maken die op vaste tijden een methode uitvoert. Eenvoudig voorbeeld (code behind):

```
using System.Windows.Threading; // bevat DispatcherTimer

namespace DemoDispatcherTimer {
    public partial class MainWindow : Window {
        private DispatcherTimer timer = new DispatcherTimer();
        private int count = 0;

        public MainWindow() {
            InitializeComponent();
            lblCount.Content = count;
            timer.Interval = TimeSpan.FromMilliseconds(1000);
            timer.Tick += timer_Tick; // voeg methode toe aan timer
            timer.Start(); // start de timer
        }

        private void timer_Tick(object sender, EventArgs e) {
            count++;
            lblCount.Content = count;
        }
    }
}
```



aantal seconden sinds start

MainWindow() en instantievariabelen





Voorbeeld: raadspelletje

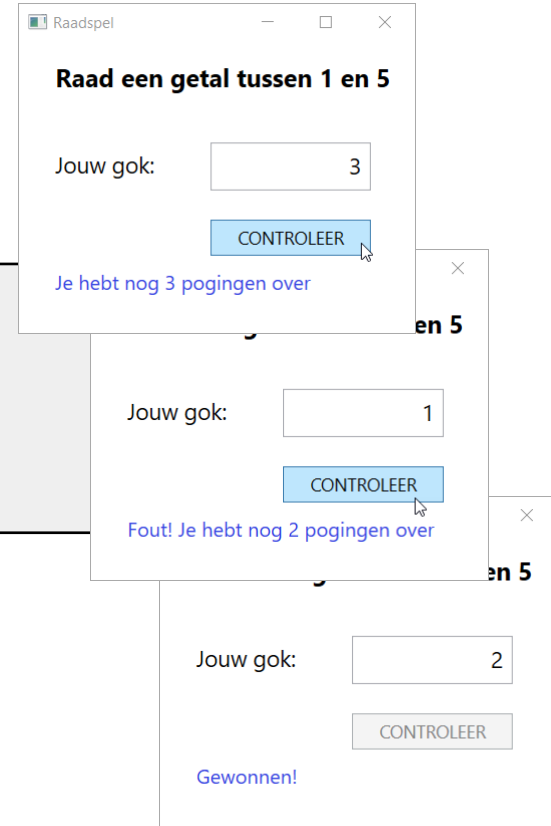
We maken een WPF spelletje waarbij je drie keer een getal kan raden. Voor het getal gebruiken we de Random generator, en verder hebben we een variabele nodig voor het aantal resterende pogingen:

```
// genereer random getal tussen 1 en 5
Random rnd = new Random();
int getal = rnd.Next(1, 6);

// onthoud aantal resterende pogingen
int pogingenOver = 3;
```

We hebben twee vragen:

1. waar in het programma genereren we dit random getal?
2. waar en hoe houden we het aantal resterende pogingen bij?





Voorbeeld: raadspelletje

Waarom kan je niet gewoon alle code in het Click event van de button zetten? Wat is de logische fout?

```
public partial class MainWindow : Window {
    public MainWindow() {
        InitializeComponent();
    }

    private void btnCheck_Click(object sender, RoutedEventArgs e) {
        Random rnd = new Random();
        int getal = rnd.Next(1, 6);
        int pogingenOver = 3;
        pogingenOver--;
        ...
    }
}
```

- *omdat dan bij elke klik een nieuw getal gegenereerd wordt, en het aantal pogingen weer op 3 gezet wordt*



MainWindow()

De juiste plek om het random getal te genereren is in MainWindow(), dat exact één keer uitgevoerd wordt, namelijk bij de opstart van het programma:

```
public partial class MainWindow : Window {
    int getal;
    int pogingenOver = 3;

    public MainWindow() {
        InitializeComponent();
        Random rnd = new Random();
        getal = rnd.Next(1, 6);
    }

    private void btnCheck_Click(object sender, RoutedEventArgs e) {
        pogingenOver--;
        ...
    }
}
```


} MainWindow() wordt exact één keer uitgevoerd, bij de start van het programma



Instantievariabelen

Om variabelen in het hele programma beschikbaar te maken, zet je ze bovenaan:

```
public partial class MainWindow : Window {  
    int getal;  
    int pogingenOver = 3; } instantievariabelen: zijn in de hele  
                               klasse beschikbaar  
  
    public MainWindow() {  
        InitializeComponent();  
        Random rnd = new Random();  
        getal = rnd.Next(1, 6); ← beschikbaar in MainWindow()  
    }  
  
    private void btnCheck_Click(object sender, RoutedEventArgs e) {  
        pogingenOver--; ← ook beschikbaar in btnCheck_Click()  
        ...  
    }  
}
```





Raadspelletje – complete code

```
public partial class MainWindow : Window {
    int getal;
    int pogingenOver = 3;

    public MainWindow() {
        InitializeComponent();
        Random rnd = new Random();
        getal = rnd.Next(1, 6);
    }

    private void btnCheck_Click(object sender, RoutedEventArgs e) {
        pogingenOver--;
        int gok = Convert.ToInt32(txtGok.Text);
        if (gok == getal) {
            lblPogingen.Content = "Gewonnen!";
            btnCheck.IsEnabled = false;
        } else {
            lblPogingen.Content = $"Fout! Je hebt nog {pogingenOver} pogingen over";
            if (pogingenOver == 0) {
                btnCheck.IsEnabled = false;
            }
        }
    }
}
```

Canvas / Ellipse / Slider





Canvas / Ellipse / Slider

De **Canvas** control kan gebruikt worden om op te tekenen, o.a. met klassen als **Rectangle** of **Ellipse**. Een eenvoudig voorbeeld (XAML):

```
...
Title="Ellips Sliders" Height="268.927" Width="313.418">
<Grid>
  <Canvas x:Name="canvas" HorizontalAlignment="Left" Height="150" Margin="21,9,0,0" VerticalAlignment="Top"
    Width="257" Background="#FFF1F1"/>
  <Slider x:Name="sldWidth" HorizontalAlignment="Left" Margin="85,164,0,0" VerticalAlignment="Top"
    Width="150" ValueChanged="sld_ValueChanged"/>
  <Slider x:Name="sldHeight" HorizontalAlignment="Left" Margin="85,190,0,0" VerticalAlignment="Top"
    Width="150" ValueChanged="sld_ValueChanged"/>
  <Label x:Name="lblHeight" Content="Label" HorizontalAlignment="Left" Margin="240,186,0,0"
    VerticalAlignment="Top"/>
  <Label x:Name="lblWidth" Content="Label" HorizontalAlignment="Left" Margin="240,160,0,0"
    VerticalAlignment="Top"/>
  <Label Content="Hoogte:" HorizontalAlignment="Left" Margin="21,160,0,0" VerticalAlignment="Top"/>
  <Label Content="Breedte" HorizontalAlignment="Left" Margin="21,186,0,0" VerticalAlignment="Top"/>
</Grid>
</Window>
```



Canvas / Ellipse / Slider

De **Canvas** control kan gebruikt worden om op te tekenen, o.a. met klassen als **Rectangle** of **Ellipse**. Een eenvoudig voorbeeld (code behind):

```
public partial class MainWindow : Window {
    private Ellipse ellipse;

    public MainWindow() {
        InitializeComponent();

        // init sliders
        sldWidth.Minimum = 0;
        sldWidth.Maximum = canvas.Width;
        sldHeight.Minimum = 0;
        sldHeight.Maximum = canvas.Height;

        // init ellips
        ellipse = new Ellipse();
        ellipse.Stroke = new SolidColorBrush(Colors.Orange);
        ellipse.Fill = new SolidColorBrush(Colors.Black);
        ellipse.Margin = new Thickness(0, 0, 0, 0);
        canvas.Children.Add(ellipse);

        ...
    }
}
```



Canvas / Ellipse / Slider

De **Canvas** control kan gebruikt worden om op te tekenen, o.a. met klassen als **Rectangle** of **Ellipse**. Een eenvoudig voorbeeld (code behind, vervolg):

```
...

// stel startwaarden van de slider in
sldWidth.Value = 100;
sldHeight.Value = 50;
Update(); // is eigenlijk niet nodig - waarom?
}

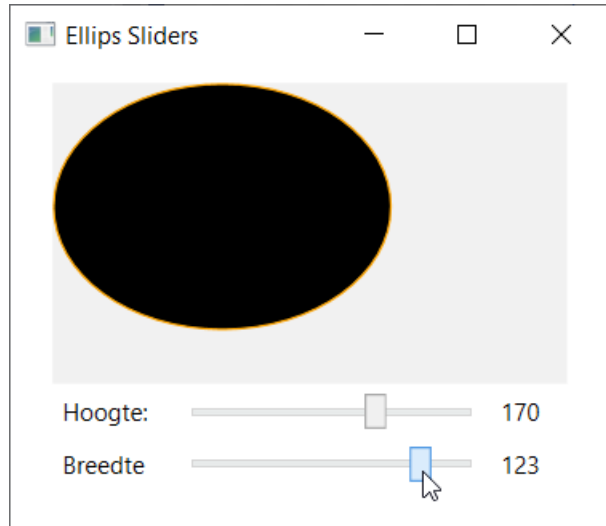
private void Update() {
    ellipse.Width = sldWidth.Value;
    ellipse.Height = sldHeight.Value;
    lblWidth.Content = sldWidth.Value;
    lblHeight.Content = sldHeight.Value;
}

private void sld_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e) {
    Update();
}
}
```

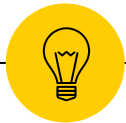



Canvas / Ellipse / Slider

De **Canvas** control kan gebruikt worden om op te tekenen, o.a. met klassen als **Rectangle** of **Ellipse**. Een eenvoudig voorbeeld (resultaat):



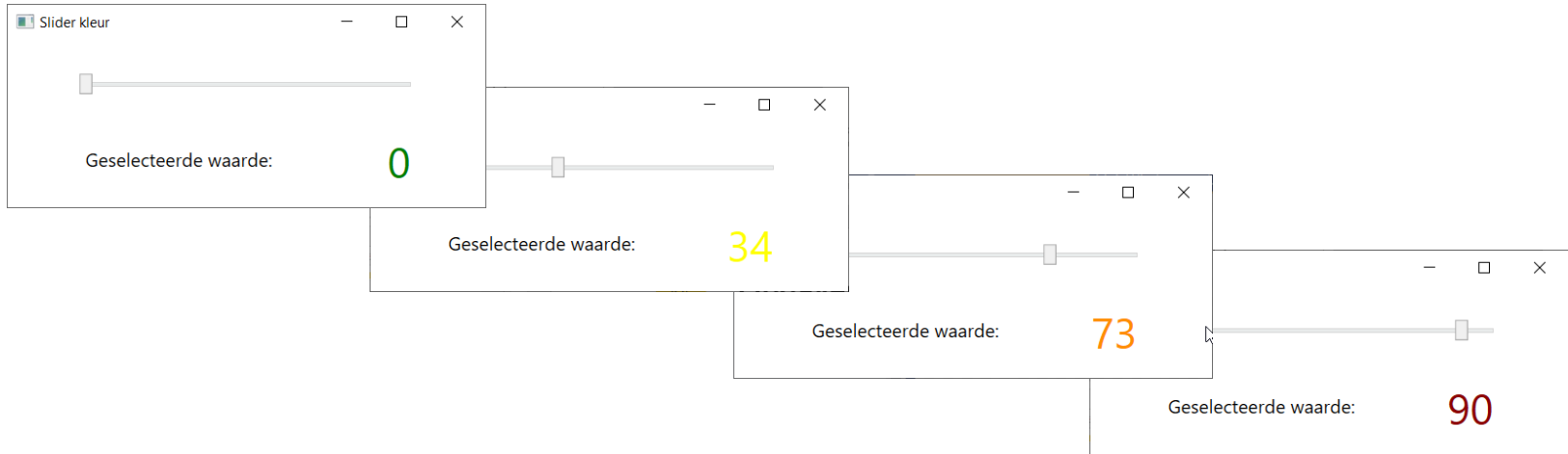
Opgaves





[Opgave WpfSliderKleur]

Maak een slider; verander de kleur van de weergegeven waarde naargelang de hoogte van de waarde (werk in stappen)



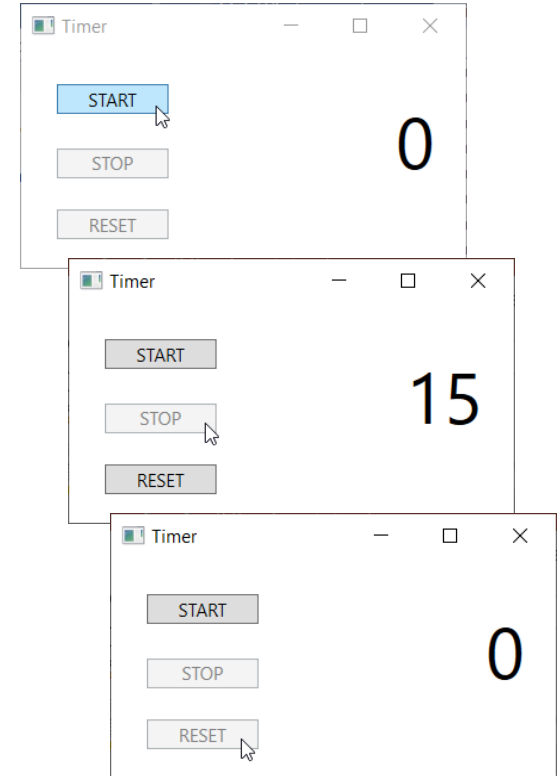
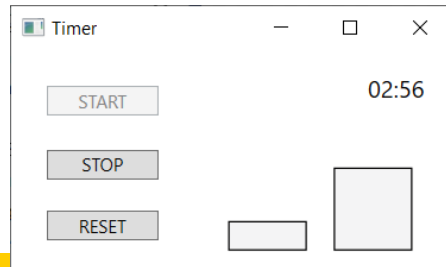


Opgave WpfTimer

Breid het voorbeeld bij de DispatcherTimer uit met een start, stop en reset knop. Let op de IsEnabled state van de buttons.

Uitbreidingen:

1. Toon in minuten en seconden
2. Maak een versie waarbij je aftelt (bv. van 5min); verander de achtergrondkleur naarmate je dichterbij 0 komt
3. Toon de seconden en minuten met twee staven (gebruik de [Rectangle](#) class)





Opgave WpfPunten

Maak een toepassing waarbij je punten kan toevoegen en verwijderen, en kan filteren op een zoektekst.

Punten

Naam:

Punt (op 100):

Filter op:

- Chris - 20/100
- Christophe - 40/100
- Victor - 60/100
- Louis - 80/100
- Jean-Louis - 80/100
- Bettina - 100/100

Punten

Naam:

Punt (op 100):

Filter op:

- Louis - 80/100
- Jean-Louis - 80/100

Punten

Naam:

Punt (op 100):

Filter op:

- Chris - 20/100
- Christophe - 40/100
- Victor - 60/100
- Louis - 80/100
- Jean-Louis - 80/100
- Bettina - 100/100

Punten

Naam:

Punt (op 100):

Filter op:

- Chris - 20/100
- Christophe - 40/100
- Victor - 60/100
- Louis - 80/100
- Jean-Louis - 80/100
- Bettina - 100/100



[Opgaves boek]

Mogelijke oefeningen uit het boek:

- 6.4 (a, b)
- 6.6
- 6.11