

Samenvatting

- Statements (opdrachten) worden altijd netjes van boven naar beneden achter elkaar uitgevoerd, tenzij we expliciet om iets anders vragen.
- C# heeft een arsenaal tekenmethoden die je kunt aanroepen om plaatjes te tekenen.
- Het grafisch systeem is gebaseerd op coördinaten.
- We kunnen bij het aanroepen van methoden de bijbehorende argumenten meegeven.
- Objecten aangemaakt via C# en XAML zijn gelijkwaardig.

Opgaven

Opmerking: we raden je aan om voorafgaand aan het schrijven van de programma's eerst wat ruwe schetsen en berekeningen te maken. Je kunt een soortgelijk project gebruiken voor alle vragen, met een Canvas om te tekenen en een knop met bijbehorend event om het tekenen te initiëren. Experimenteer ook eens met een XAML-versie in plaats van een C#-versie, zoals uitgelegd in paragraaf 3.11.

3.1

Schrijf een programma dat een rechthoekige driehoek tekent. Kies geschikte afmetingen.

3.2

Schrijf een programma dat een leeg boter-kaas-en-eierenbord tekent, opgebouwd uit lijnen. Dit bord ziet er zo uit:

3.3

Ontwerp een simpel huis en schrijf een programma dat dit huis tekent.

3.4

Dit zijn wat gegevens voor de jaarlijkse regenval in het fictieve land Xanadu:

2009	150 cm
2010	175 cm
2011	120 cm
2012	130 cm

- Geef deze gegevens weer in de vorm van een serie horizontale lijntjes.
- Gebruik nu in plaats van lijntjes ingekleurde rechthoeken.

3.5

Schrijf een programma dat een roos van verschillend gekleurde concentrische cirkels tekent op het scherm. (De aanschaf van een pistool met zuignappijltjes om daarmee op het scherm te schieten is facultatief!)

3.6

Schrijf een programma dat in grote lijnen een gezicht tekent. Je kunt hierbij ellipsen gebruiken, zowel voor de omtrek van het gezicht, als voor de ogen, de oren, de neus en de mond.

Antwoorden op de testvragen

3.1 We doen net alsof de cirkel precies omvat wordt door een vierkant met zijden van honderd pixels:

→ **Project: h03\TestVraag3_1 | Bestand: MainWindow.xaml.cs**

```
Ellipse circle = new Ellipse();
circle.Width = 100;
circle.Height = 100;
circle.Margin = new Thickness(10, 10, 0, 0);
circle.Stroke = new SolidColorBrush(Colors.Black);
circle.Fill = new SolidColorBrush(Colors.Red);
paperCanvas.Children.Add(circle);
```

3.2 Geef het BitmapImage-object een UriSource, als volgt:

```
bi.UriSource = new Uri("imagedemo.jpg", UriKind.RelativeOrAbsolute);
```

Voeg het bestand **imagedemo.jpg** toe aan het project en zet de property **Copy to Output Directory** op **Copy always** en stel de property **Build Action** in op **Content**.

3.3 De backslash in het pad naar de afbeelding is een speciaal teken. Door het @-teken zorg je ervoor dat je dit teken niet op een speciale manier moet herschrijven (*escapen*). Als je het @-teken wilt weglaten, moet je schrijven:

```
bi.UriSource = new Uri("C:\\Users\\Kris\\Pictures\\imagedemo.jpg");
```

3.4 Een T-vorm teken je als volgt:

→ **Project: h03\TestVraag3_4 | Bestand: MainWindows.xaml.cs**

```
Line horizontal = new Line();
horizontal.X1 = 20; horizontal.Y1 = 20;
horizontal.X2 = 120; horizontal.Y2 = 20;
```