

Selecties

Programming Basics

INHOUD

1	BOOLEAANSE VARIABELEN	3
2	BOOLEAANSE OPERATOREN	4
3	VOORWAARDELIJKE OPERATOREN	5
4	VOLGORDE VAN BEWERKINGEN	6
5	SELECTIE: IF	7
5.1	If: als de bewering klopt...	7
5.2	... en als de bewering niet klopt: else	7
5.3	Statements groeperen: codeblokken	8
5.4	If statements nesten: else if	9
6	SWITCH	12

1 BOOLEAANSE VARIABLEN

In de wereld van het programmeren wordt gewerkt met bits. Een bit heeft de waarde 1 of 0. Dit zijn de twee toestanden van een bit.

In het programmeren moet een expressie altijd als waar of onwaar geëvalueerd worden: 1 of 0. In het dagelijkse leven kan een antwoord op een vraag vaak nogal vaag zijn: misschien, waarschijnlijk, ... Wanneer we programmeren, moet het antwoord steeds exact gekend zijn.

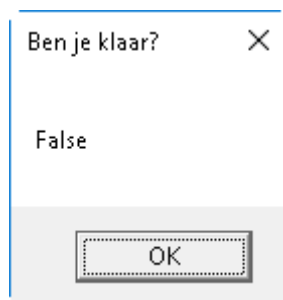
We wijzen een variabele *x* de waarde 99 toe. Laten we even kijken hoe we dan statements in verband met *x* kunnen evalueren:

- *x* bevat de waarde 99 → evalueert als waar (true)
- *x* is kleiner dan 10 → evalueert als onwaar (false)

C# gebruikt een type **bool** dat één van de twee statussen waar of onwaar kan bevatten. De naam bool is afkomstig van de Booleaanse algebra en is een tak uit de exacte wiskunde.

- Maak een nieuwe Visual Studio solution **Prb.Selection**.
- Maak in de solution Selectie een nieuwe WPF Application **Prb.Selection.Wpf**.
- Voorzie in MainWindow een Button **btnTestBool**
- Wanneer op de knop wordt geklikt, wordt de volgende code uitgevoerd:

```
private void BtnTestBool_Click(object sender, RoutedEventArgs e)
{
    bool areYouReady = false;
    MessageBox.Show(areYouReady.ToString(), "Ben je klaar?");
}
```



2 BOOLEAANSE OPERATOREN

Een Booleaanse operator voert een berekening uit die resulteert in waar of onwaar (true/false). De eenvoudigste operator is ! (het uitroepteken): dit is de operator **NOT**—true wordt false en false wordt true.

- We voegen een knop toe aan de applicatie met de naam **btnTestNot** als opschrift 'Test NOT'.

```
private void BtnTestNot_Click(object sender, RoutedEventArgs e)
{
    bool areYouReady = false;
    areYouReady = !areYouReady;
    MessageBox.Show(areYouReady.ToString(), "Ben je klaar?");
}
```

Het antwoord is nu *true*.

In C# heb je verschillende operatoren om na te gaan of zaken al dan niet gelijk zijn en kleiner/groter.

Operator	Betekenis	Voorbeeld	Uitkomst als x = 99
==	Gelijk aan	x == 100	false
!=	Niet gelijk aan	x != 100	true
<	Kleiner dan	x < 21	false
>	Groter dan	x > 21	true
<=	Kleiner dan of gelijk	x <= 21	false
>=	Groter dan of gelijk	x >= 21	true

3 VOORWAARDELIJKE OPERATOREN

C# kent ook twee operatoren waarmee voorwaarden aan elkaar gekoppeld kunnen worden:

- De logische EN (AND) voorgesteld door **&&**
- De logische OF (OR) voorgesteld door **||**

Voorbeelden:

```
void IsEvening()
{
    int hour = 19;
    bool evening = (hour >= 18) && (hour <= 22);
    MessageBox.Show("Om " + hour + " uur: " + evening, "Is het al avond?");
}
```

De variabele *evening* bevat *true* wanneer *hour* groter of gelijk is aan 18 **EN** kleiner of gelijk is aan 22.

```
void IsNight()
{
    int hour = 23;
    bool night = (hour > 22) || (hour < 6);
    MessageBox.Show("Om " + hour + " uur: " + night, "Is het nacht?");
}
```

De variabele *night* bevat *true* wanneer *hour* groter is aan 22 **OF** kleiner dan 6.

4 VOLGORDE VAN BEWERKINGEN

Volgende tabel geeft aan welke prioriteit een bewerking heeft ten opzichte van een andere. Hoe hoger de sectie in de tabel, hoe hoger de prioriteit. Een bewerking met een **hogere prioriteit** wordt in een opdracht uitgevoerd vóór een bewerking met lagere prioriteit. Bewerkingen met een gelijke prioriteit worden **van links naar rechts** uitgevoerd.

operator	beschrijving
()	prioriteit aanpassen
++	verhogen als suffix (vb. i++)
--	verlagen als suffix (vb. i--)
!	NOT
++	verhogen als prefix (vb ++i)
--	verlagen als prefix (vb. --i)
*	vermenigvuldigen
/	delen
%	modulo: rest bij deling
+	optellen
-	aftrekken
<	kleiner dan
>	groter dan
<=	kleiner dan of gelijk aan
>=	groter dan of gelijk aan
==	gelijk aan
!=	niet gelijk aan
&&	en
	of
=	toekenning

5 SELECTIE: IF

Een selectie is een moment in een programma waarop een beslissing moet worden genomen over hoe het programma verder zal worden uitgevoerd. Enkel ALS aan een voorwaarde is voldaan, worden bepaalde statements uitgevoerd.

Het sleutelwoord voor het nemen van een beslissing is **if**.

5.1 IF: ALS DE BEWERING KLOPT...

De algemene syntax voor het nemen van een beslissing is als volgt:

```
if (booleaanse expressie)
    //statement als booleaanse expressie waar is;
```

Voorbeeld:

```
int age = 19;
if (age >= 18)
    MessageBox.Show("Op de leeftijd van " + age + " ben je meerderjarig");
if (age < 18)
    MessageBox.Show("Op de leeftijd van " + age + " ben je nog niet meerderjarig");
```

5.2 ... EN ALS DE BEWERING NIET KLOPT: ELSE

Algemene syntax:

```
if (booleaanse expressie)
    //statement als booleaanse expressie waar is;
else
    //statement als booleaanse expressie niet waar is;
```

In het vorige voorbeeld hadden we met **else** kunnen werken:

```
if (age >= 18 )
    MessageBox.Show("Op de leeftijd van " + age + " ben je meerderjarig");
else
    MessageBox.Show("Op de leeftijd van " + age + " ben je nog niet meerderjarig");
```

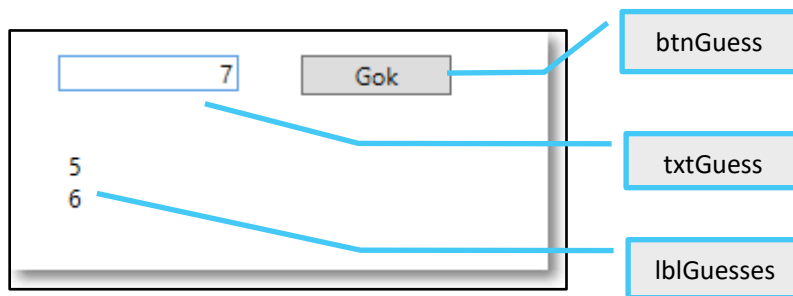
5.3 STATEMENTS GROEPEREN: CODEBLOKKEN

Gebruik accolades { ... } om statements te groeperen: meer dan 1 statement binnen if / else.

```
if (booleaanse expressie)
{
    //statements als booleaanse expressie waar is;
}
else
{
    //statements als booleaanse expressie niet waar is;
}
```

Voorbeeld: raadspel

Maak binnen je solution een nieuw project **Prb.Selection.Gambling.Wpf** aan.



Wanneer het formulier geladen wordt, laten we de computer een getal van 1 – 10 kiezen via de Random-klasse.

Declareer een random altijd als een instance-variabele (dwz op klasseniveau). Op die manier is de kans veel groter dat je echt willekeurige getallen krijgt.

```
int toGuess;
int attempts;
Random rnd = new Random();

private void Window_Loaded(object sender, RoutedEventArgs e)
{
    toGuess = rnd.Next(1, 11);
    Debug.WriteLine("Te raden: " + toGuess);
    attempts = 0;
}
```

Bij een klik op de knop *gok* wordt het ingegeven getal gelezen en zijn er 2 mogelijkheden:

- het getal is geraden: de speler krijgt feedback over het geraden getal en het aantal pogingen.
- het getal is niet geraden: het geraden getal wordt getoond in een label. De inhoud van txtGuess wordt gewist en de focus gaat naar deze textbox.


```
private void BtnGuess_Click(object sender, RoutedEventArgs e)
{
    int guess = int.Parse(txtGuess.Text);
    attempts++;

    if (guess == toGuess)
    {
        lblGuesses.Content = "Je hebt het getal " + toGuess + " geraden na " +
attempts + " pogingen";
    }
    else
    {
        lblGuesses.Content += guess + Environment.NewLine;
        txtGuess.Text = "";
        txtGuess.Focus();
    }
}
```

5.4 IF STATEMENTS NESTEN: ELSE IF

Bij een if ... else – selectie ben je als programmeur beperkt tot twee mogelijkheden: ofwel is het if-statement waar, ofwel niet. De realiteit is echter vaak ingewikkelder dan dat.

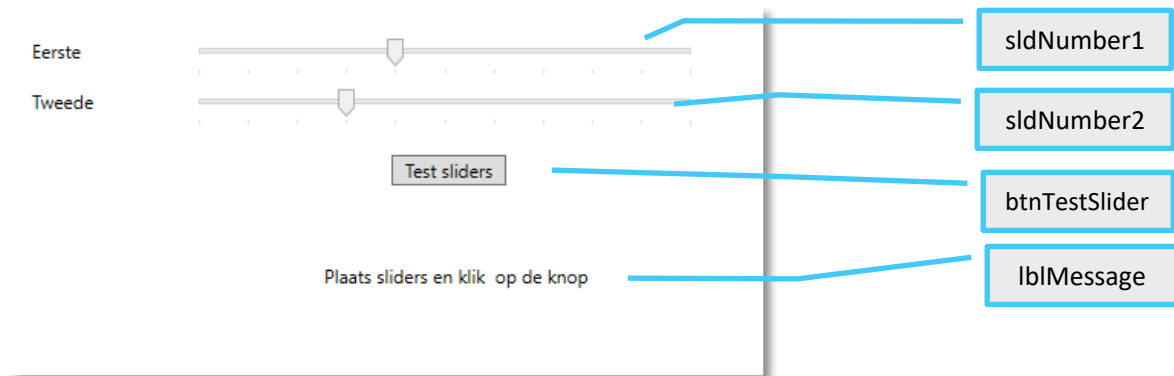
```
string color = "oranje";
string action;
if (color == "rood")
    action = "stop";
else if (color == "groen")
    action = "doorrijden";
else
    action = "stop indien mogelijk";
```

Een ander voorbeeld: voeg een button **btnWeekDay** toe aan het MainWindow van Prb.Selection.Wpf. Het click-event ziet er als volgt uit:

```
string dayName;
DateTime currentDate = DateTime.Now;
DayOfWeek currentDay = currentDate.DayOfWeek;
if (currentDay == DayOfWeek.Sunday)
    dayName = "zondag";
else if (currentDay == DayOfWeek.Monday)
    dayName = "maandag";
else if (currentDay == DayOfWeek.Tuesday)
    dayName = "dinsdag";
else if (currentDay == DayOfWeek.Wednesday)
    dayName = "woensdag";
else if (currentDay == DayOfWeek.Thursday)
    dayName = "donderdag";
else if (currentDay == DayOfWeek.Friday)
    dayName = "vrijdag";
else
    dayName = "zaterdag";
MessageBox.Show("Vandaag is het " + dayName, "Dag van de week");
```

Toepassing: waarden vergelijken

Maak binnen je solution een nieuw project aan met de naam **Prb.Selection.Sliders.Wpf**



De sliders hebben o.a. volgende instellingen:

Property	Value	Omschrijving
AutoToolTipPlacement	BottomRight	Waarde wordt getoond tijdens slepen
TickPlacement	BottomRight	Intervalstreepjes worden getoond
IsSnapToTickEnabled	True	Enkel een waarde op de intervalstreepjes kan geselecteerd worden

Als op de knop wordt geklikt, wordt een melding getoond:

- eerste = tweede
- eerste < tweede
- eerste > tweede



Extra

- verander de tekstkleur van de melding
- gebruik het event `ValueChanged` van de Sliders om de melding te plaatsen
verwijder de knop.

```
int number1 = (int)sldNumber1.Value;
int number2 = (int)sldNumber2.Value;
string message;
if (number1 == number2) {
    message = "eerste = tweede";
}
else if (number1 < number2) {
    message = "eerste < tweede";
}
else {
    message = "eerste > tweede";
}
lblMessage.Content = message;
```

Een if-statement nesten, kan ook betekenen dat er een if-statement binnen een ander selectieblok geplaatst wordt. Dit zien we in het volgende voorbeeld:

```
int age = 30;
int ticketPrice;
string profession = "ambtenaar";

if (profession == "werkloos" || profession == "gepensioneerd")
{
    ticketPrice = 0;
}
else
{
    if (age < 12)
    {
        ticketPrice = 0;
    }
    else
    {
        ticketPrice = 12;
    }
}

MessageBox.Show("De prijs van je ticket bedraagt: " + ticketPrice + " euro");
```

6 SWITCH

In het voorbeeld over de weekdays wordt bij elke else if 'dag ==' herhaald. Om de code in dergelijke gevallen leesbaarder te maken, kun je beter een switch statement gebruiken.

```
switch(controleExpressie)
{
    case constanteExpressie :
        statements;
        break;
    case constanteExpressie :
        statements;
        break;
    ...
    default :
        statements;
        break;
}
```

Opgepast, het sleutelwoord **break** is heel belangrijk als laatste statement van elke **case**. Dit statement beëindigt de switch-constructie. Zonder break wordt de volgende case ook uitgevoerd, ook al is de voorwaarde niet vervuld. In Visual Studio krijg je een foutmelding als je de break vergeet. I.p.v. break kan ook **return** of **throw** (zie hoofdstuk over exceptions) gebruikt worden.

Toegepast op ons concreet voorbeeld:

```
string dayName;
DateTime currentDate = DateTime.Now;
DayOfWeek currentDay = currentDate.DayOfWeek;
switch(currentDay)
{
    case DayOfWeek.Sunday:
        dayName = "zondag";
        break;
    case DayOfWeek.Monday:
        dayName = "maandag";
        break;
    case DayOfWeek.Tuesday:
        dayName = "dinsdag";
        break;
    case DayOfWeek.Wednesday:
        dayName = "woensdag";
        break;
    case DayOfWeek.Thursday:
        dayName = "donderdag";
        break;
    case DayOfWeek.Friday:
        dayName = "vrijdag";
        break;
    default:
        dayName = "zaterdag";
        break;
}
MessageBox.Show("Vandaag is het " + dayName, "Dag van de week");
```



Code repository

De volledige broncode van dit hoofdstuk is te vinden op

 `git clone` <https://github.com/howest-gp-prb/cu-h5-selectie.git>

