

C# Essentials: Iteratie

Lector: Tom Quareme

Even tussendoor: const

- **const: constante**
 - Maak een waarde read-only. \Rightarrow enkel leesbaar, niet herschrijfbaar
 - De waarde kan niet veranderen.
- `const float pi = 3.141592f;`
- `const string naam = "Tom";`
- `const int dagenInWeek = 7;`

Wat is itereren?

- Instructies herhalen (opnieuw doen).
- Verschillende soorten lussen (loops):
 - **for**: vaste hoeveelheid herhalingen
 - **while**: herhalen onder voorwaarde
 - **do while**: herhalen onder voorwaarde, maar eerst uitvoeren en daarna voorwaarde controleren
- `if (booleaanse expressie)`
 {
 code
 }

for

- **for:** vaste hoeveelheid herhalingen
 - `for (beginwaarde; voorwaarde; stapgrootte)`

```
{  
    // code  
}
```

 - Beginwaarde: teller initialiseren
 - Voorwaarde: einde van de lus bepalen
 - Stapgrootte: optellen of aftellen in bepaalde stapgrootte
- ```
for (int i = 0; i < 10; i++)
{
 Console.WriteLine(i);
}
Console.WriteLine("Optellen van 0 t.e.m. 9 in stappen van 1");
```

# for

- **for-lus aftellen voorbeeld:**
- ```
for (int i = 9; i > 0; i--)  
{  
    Console.WriteLine(i);  
}  
Console.WriteLine("Aftellen van 9 t.e.m. 0 in stappen van 1");
```

while

- **while:** herhalen onder voorwaarde (zolang als aan de voorwaarde voldaan is)
- while (voorwaarde)
 {
 // code
 }
- ```
int waarde = 0;
while (waarde < 100)
{
 Console.WriteLine("Waarde {0} is kleiner dan 100!", waarde);
 waarde++;
}
```

# do while

- **do while:**
  - herhalen onder voorwaarde (zolang als aan de voorwaarde voldaan is)
  - Hetzelfde als while, alleen dat de voorwaarde pas op einde gecontroleerd wordt.
- ```
do
{
    // code die zeker minstens 1 keer wordt uitgevoerd
} while (voorwaarde);
```
- ```
int waarde = 0;
do
{
 waarde *= 2;
 Console.WriteLine(waarde);
} while (waarde < 100);
```

# Lussen in elkaar nesten

- Lussen in andere lussen steken.
  - For lussen in elkaar steken, while lussen in elkaar, for in while, while in for,...
- Nodig voor complexere programma's.

- ```
Console.WriteLine("Print een 10x10 vierkant:\r\n");
for (int i = 0; i < 10; i++)
{
    for (int j = 0; i < 10; j++)
    {
        Console.Write("* ");
    }
    Console.WriteLine(); // lege regel afdrukken
}
```


Oneindige lus (infinite loop)

- **Oneindige lus:**
 - Lus die het programma laat vastlopen. De lus blijft oneindig doorgaan en stopt niet.
 - **for:** eindvoorwaarde wordt nooit bereikt
 - **while:** voorwaarde wordt nooit false
- ```
int i = 0;
for (int i = 10; i >= 10; i++)
{
 Console.WriteLine(i);
}
Console.WriteLine("Tellen van 10 tot oneindig.");
```

# Oneindige lus (infinite loop)

- **Oneindige lus:**
  - Lus die het programma laat vastlopen. De lus blijft oneindig doorgaan en stopt niet.
  - **for:** eindvoorwaarde wordt nooit bereikt
  - **while:** voorwaarde wordt nooit false
- ```
int i = 0;
while (true)
{
    Console.WriteLine(i++);
}
Console.WriteLine("Dit raakt nooit afgeprint!");
```

Oneindige lus (infinite loop)

- Stel dat je in een oneindige lus komt.
- Dan moet je het programma voortijdig afbreken.
- **Programma voortijdig afbreken:**
 - **In console programma:**
 - CTRL+C drukken
 - **In GUI/console programma:**
 - Stop debugging knop
 - Shift + F5 in Visual Studio

