

C# Essentials Controle structuren

Koen Bloemen



**DE HOGESCHOOL
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be





- Selectie
- Iteratie

Selectie

- Selecties worden in programma's gebruikt om keuzes te maken. We onderscheiden:
 - if
 - if else
 - Ternary operator
 - Nesting
 - Switch

Selectie – if

- if
 - Keuzes maken
 - Als (voorwaarde), dan {code uitvoeren}
 - *voorwaarde* is een booleaanse expressie (true of false)

```
static void Main(string[] args)
{
    Console.WriteLine("Geef een getal...");
    string tekst = Console.ReadLine();
    int getal = Convert.ToInt32(tekst);
    if (getal == 10)
    {
        Console.WriteLine("Dit getal is gelijk aan 10.");
    }
    getal = 100;
    if (getal == 5)
    {
        // Deze blok wordt niet uitgevoerd! Waarom?
        Console.WriteLine("Dit getal is gelijk aan 5.");
    }
    Console.ReadLine();
}
```

Selectie – if

- Als er maar 1 statement binnen {} van *if* staat, mag je {} weglaten

```
static void Main(string[] args)
{
    Console.WriteLine("Geef een getal...");
    string tekst = Console.ReadLine();
    int getal = Convert.ToInt32(tekst);
    if (getal == 10)
        Console.WriteLine("Dit getal is gelijk aan 10.");
    getal = 100;
    if (getal == 5) Console.WriteLine("Dit getal is gelijk aan 5.");
    Console.ReadLine();
}
```

Selectie – if else

- Voer code uit in {} van *else* indien niet voldaan aan voorwaarde in *if*

```
static void Main(string[] args)
{
    Console.WriteLine("Geef een getal...");
    string tekst = Console.ReadLine();
    int getal = Convert.ToInt32(tekst);
    if (getal == 10)
    {
        Console.WriteLine("Dit getal is gelijk aan 10.");
    }
    else
    {
        Console.WriteLine("Dit getal is verschillend van 10.");
    }
    Console.ReadLine();
}
```

Selectie – if else

- Als er maar 1 statement binnen {} van *if* of *else* staat, mag je {} weglaten

```
static void Main(string[] args)
{
    Console.WriteLine("Geef een getal...");
    string tekst = Console.ReadLine();
    int getal = Convert.ToInt32(tekst);
    if (getal == 10)
        Console.WriteLine("Dit getal is gelijk aan 10.");
    else
        Console.WriteLine("Dit getal is verschillend van 10.");
    Console.ReadLine();
}
```

Selectie – ternary operator

```
static void Main(string[] args)
{
    Console.WriteLine("Geef een getal...");
    string tekst = Console.ReadLine();
    int getal = Convert.ToInt32(tekst);
    string uitvoer = (getal == 10) ? "Dit getal is gelijk aan 10." : "Dit getal is verschillend van 10.";
    Console.WriteLine(uitvoer);
    Console.ReadLine();
}
```


Selectie – nesting

- Je kan onbeperkt *else if* blokken gebruiken

```
static void Main(string[] args)
{
    double getal = 15.2;
    if (getal < 10.0)
    {
        Console.WriteLine("Dit getal is kleiner dan 10.");
    }
    else if (getal >= 10 && getal < 20)
    {
        Console.WriteLine("Dit getal is groter of gelijk aan 10 en kleiner dan 20.");
    }
    else
    {
        Console.WriteLine("Dit getal groter of gelijk aan 20");
    }
}
```

Selectie – nesting

- Een *if* blok kan je ook in een andere *if* blok gebruiken.

```
static void Main(string[] args)
{
    double getal = 2.718;
    if (getal < 3)
    {
        if (getal > 2)
            Console.WriteLine("tussen 2 en 3");
        else if (getal > 1)
            Console.WriteLine("tussen 1 en 2");
        else
            Console.WriteLine("<= 1");
    }
}
```

Selectie – switch

- Test op expliciete gevallen (cases)
- Enkel echte gelijkheden (geen $<$ of $>$)
- Met *break* spring je uit de switch

Selectie – switch

```
static void Main(string[] args)
{
    string fruit = Console.ReadLine();
    bool isLekker = false;
    switch (fruit)
    {
        case "peer":
            isLekker = true;
            break;
        case "kers":
            isLekker = true;
            break;
        case "citroen":
        case "pompelmoes":
            isLekker = false;
            break;
        default:
            break;
    }
    string uitvoer = "Fruit is " + (isLekker ? "lekker" : "niet lekker");
    Console.WriteLine(uitvoer);
    Console.ReadLine();
}
```

Iteratie

- Wat is itereren?
 - Instructies herhalen (opnieuw doen)
 - Verschillende soorten
 - for
vaste hoeveelheid herhalingen
 - while
herhalen onder voorwaarde
 - do while
herhalen onder voorwaarde, maar eerst uitvoeren en dan pas voorwaarde controleren

Iteratie – for

- Een vaste hoeveelheid herhalingen
 - for (beginwaarde; voorwaarde; stapgrootte) { //code }

```
Console.WriteLine("Optellen van 0 t.e.m. 9 in stappen van 1");
for (int i = 0; i < 10; i++)
{
    Console.WriteLine(i);
}
Console.WriteLine("Aftellen van 9 t.e.m. 0 in stappen van 1");
for (int i = 9; i >= 0; i--)
{
    Console.WriteLine(i);
}
```

Iteratie – while

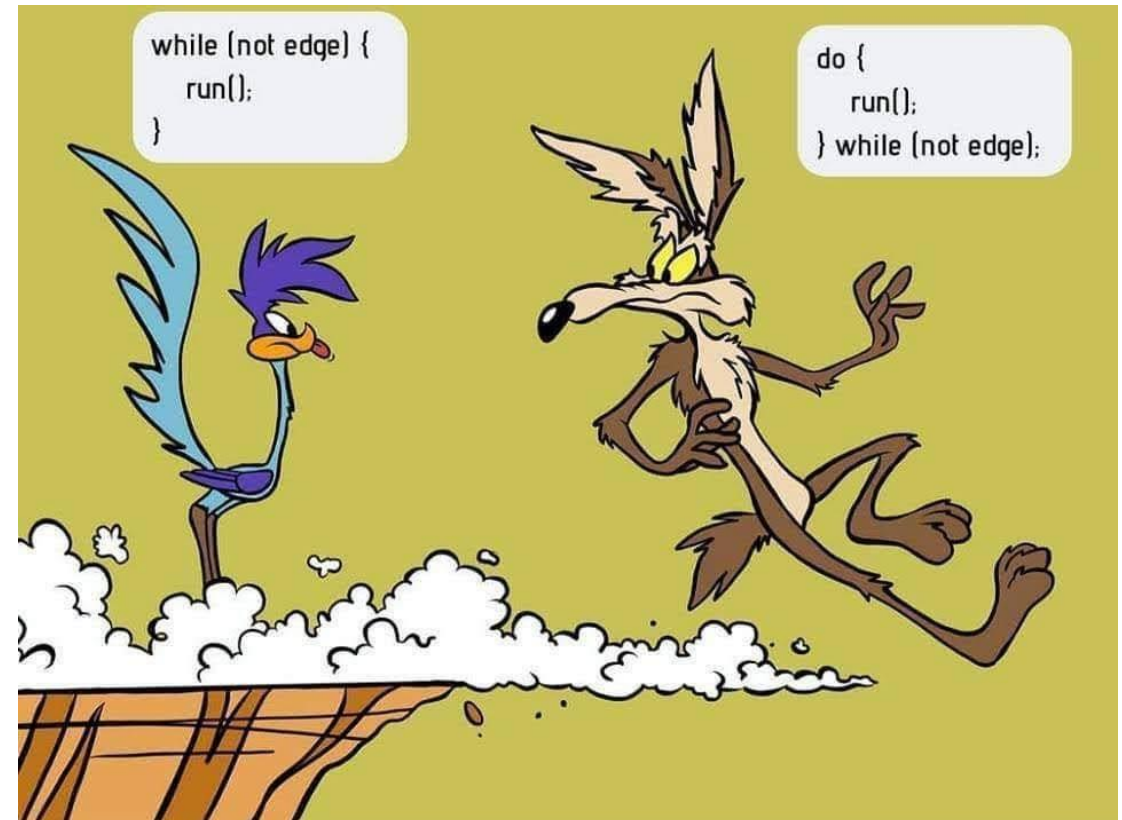
- Herhalen onder voorwaarde
 - while (voorwaarde) { //code }

```
int waarde = 0;
while (waarde < 100)
{
    Console.WriteLine("Waarde {0} is kleiner dan 100!", waarde);
    waarde++;
}
```

Iteratie – do while

- Herhalen onder voorwaarde
 - Hetzelfde als while, maar de voorwaarde wordt pas op het einde gecontroleerd
 - Minstens 1 herhaling
 - `do { //code } while (voorwaarde);`
 - Afsluiten met `;`

```
int waarde = 1;  
do  
{  
    waarde *= 2;  
    Console.WriteLine(waarde);  
} while (waarde < 100);
```



Iteratie – nesting

- Lussen in andere lussen steken
- Soms nodig in complexere programma's
- Proberen dit te minimaliseren!

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

Iteratie – oneindige lus


- Lus die het programma laat vastlopen
- Blijft oneindig doorgaan en stopt niet

```
Console.WriteLine("Tellen van 10 tot oneindig.");  
for (int i = 10; i >= 10; i++)  
{  
    Console.WriteLine(i);  
}
```


Iteratie – oneindige lus

```
int i = 0;
while (true)
{
    Console.WriteLine(i++);
}
Console.WriteLine("Dit raakt nooit afgeprint!");
```

```
static void Main(string[] args)
{
    int i = 0;
    while (true)
    {
        Console.WriteLine(i++);
    }
    Console.WriteLine("Dit raakt nooit afgeprint!");
}
```

 **class System.Console**

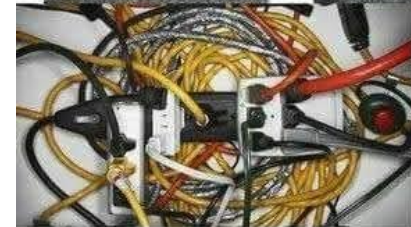
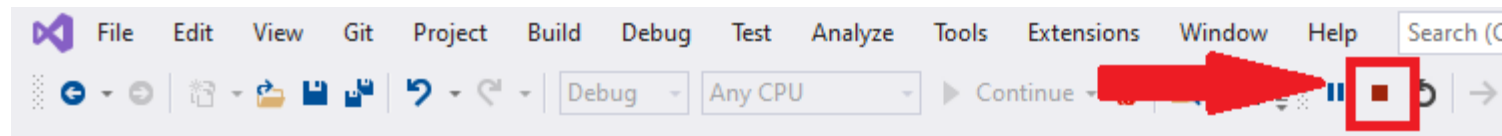
Represents the standard input, output, and error streams for console applications. This class cannot be inherited. To browse the .NET Framework source code for this type, see the Reference Source.

CS0162: Unreachable code detected

[Show potential fixes](#) (Alt+Enter or Ctrl+;)

Iteratie – oneindige lus

- Oneindige lus => programma voortijdig afbreken
 - Enkel in console applicatie => CTRL-C drukken
 - In GUI / console applicatie
 - Stop debugging knop
 - Shift-F5 in Visual Studio



try



catch