

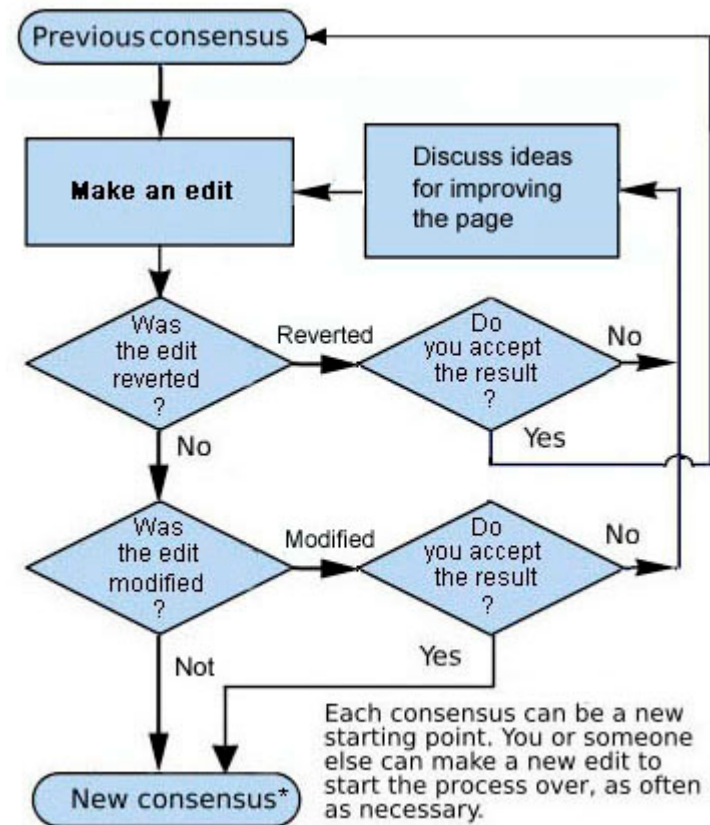
# 1. Intro loops

H6. Herhalingen, herhalingen, herhaling



# Program flow

- Na **'straight line'** en **'branching/beslissingen'** nu ook **'herhalingen'**



# Wanneer zijn loops nuttig?

- Moet je veel code (1 of meerdere lijnen) onder mekaar copy pasten?  
Loops to the rescue!

```
The programmer got stuck in the shower  
    because the instructions  
    on the shampoo bottle said...
```

```
Lather, Rinse, Repeat.
```

# Soorten loops

## Definite loop or counted loop

- Loop waar het aantal 'iteraties' vooraf van is gekend

## Indefinite of sentinel loop

- Waarde van de controlewaarde wordt niet door berekening aangepast maar bv door actie/invoer van de gebruiker

## Oneindige loop

- Soms bewust, soms een bug, soms het einde van de wereld.

## Empty body

- Codeblock zonder code in (enkel accoladepaar)...useless



# Soorten C# loops

- 3 types of loops in C#
  - **while loop**
  - **for loop**
  - **do loop (or do-while loop)**
- 3 manieren om zelfde problemen op te lossen, ieder met hun eigen subtiele verschillen
- 4e type loop “Foreach” zien we in volgende semester

# Nieuwe termen



## Loop

Structuur die het toelaat om een code block herhaaldelijk uit te voeren



## Loop body

Code block binnen de loop

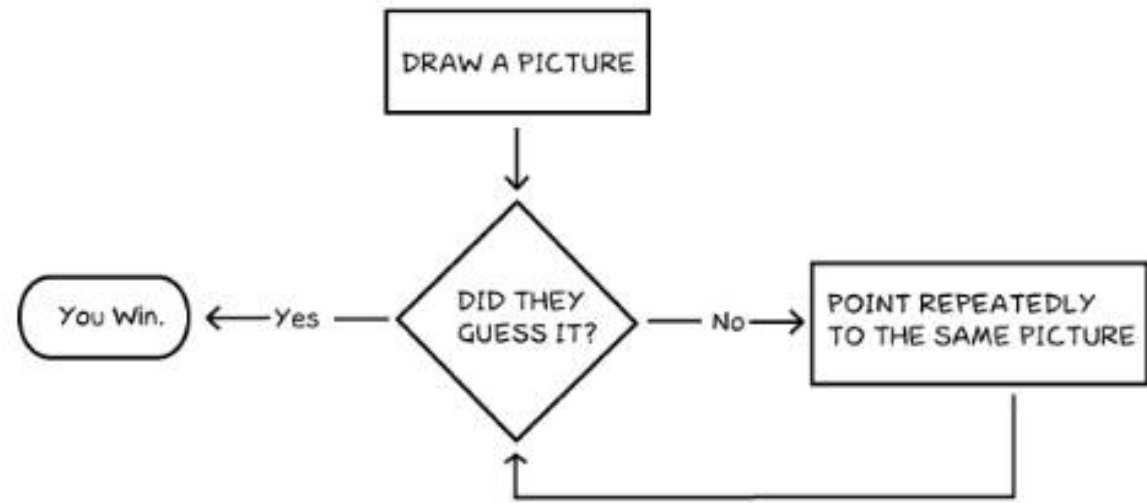
# 2. While en Do While

H6. Herhalingen, herhalingen, herhaling



# While

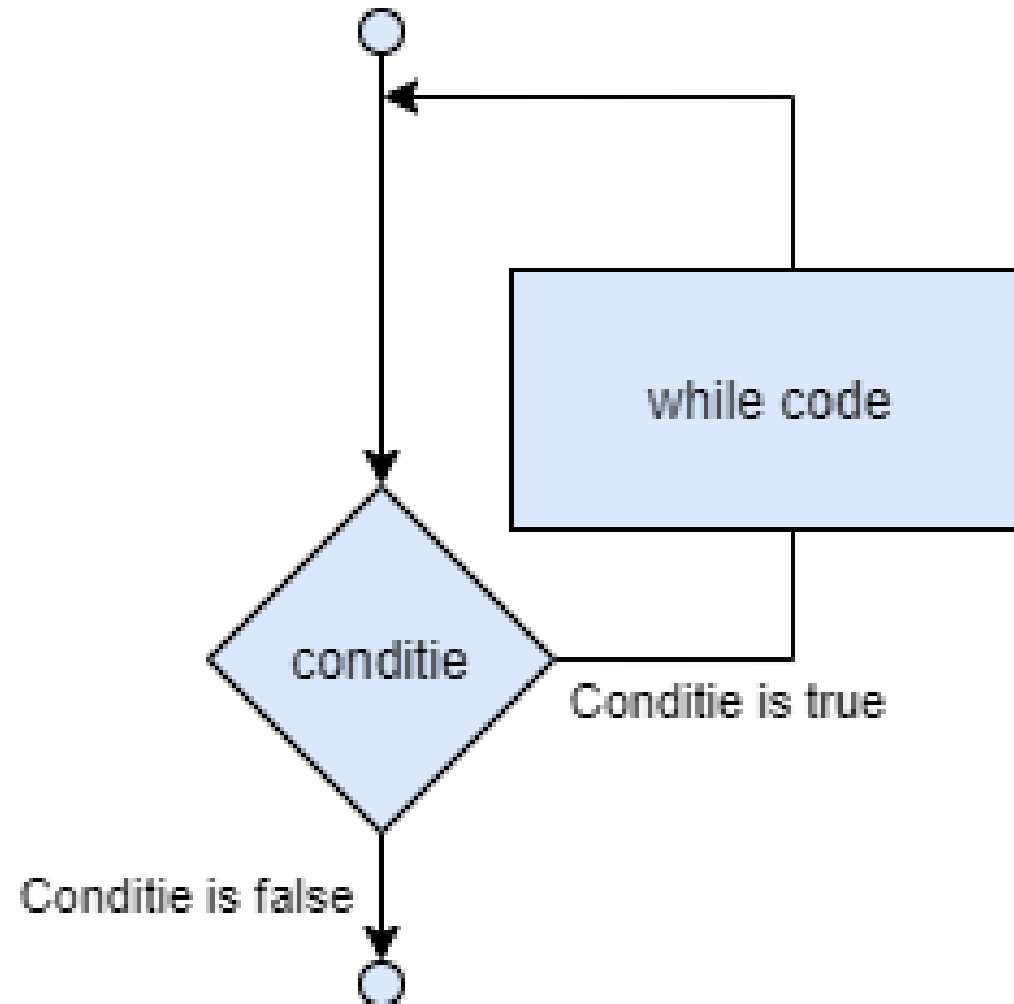
## How To Play Pictionary



Doghouse Diaries  
"Where pennies are a dime a dozen."



# Gebruik van de while loop



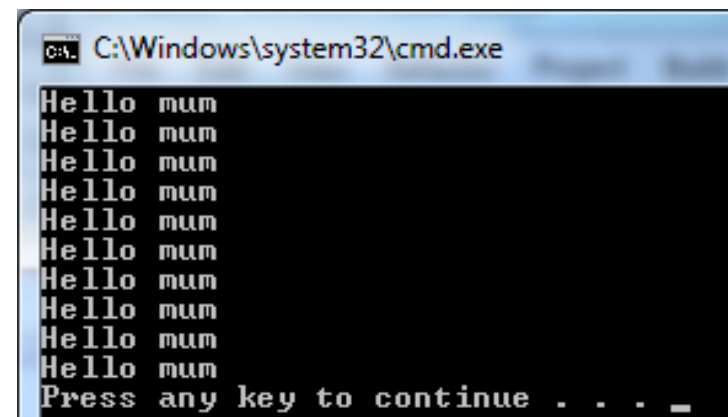
# Gebruik van de while loop



- **while loop**
  - Gebruikt om groep van statements (block) te blijven herhalen zolang de while conditie *true* blijft
- **Infinite loop**
  - Een loop die nooit stopt (bug of bewust?)

# Zelfde actie aantal keer herhalen

```
int i ;  
i = 1 ;  
while ( i < 11 ) {  
    Console.WriteLine ( "Hello mum" ) ;  
    i = i + 1 ;  
}
```



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The window displays the output of the C# program: "Hello mum" repeated 10 times on separate lines. At the bottom, it says "Press any key to continue . . . \_".

# While loop correct laten eindigen?

- Initializeer de 'loop controle variabele' (= **wachterwaarde/sentinel**) VOOR de loop
- Test de controle variabele IN de while expressie
- Pas in de while-loop code de controle variabele aan (of niet)

# Gebruik while loop

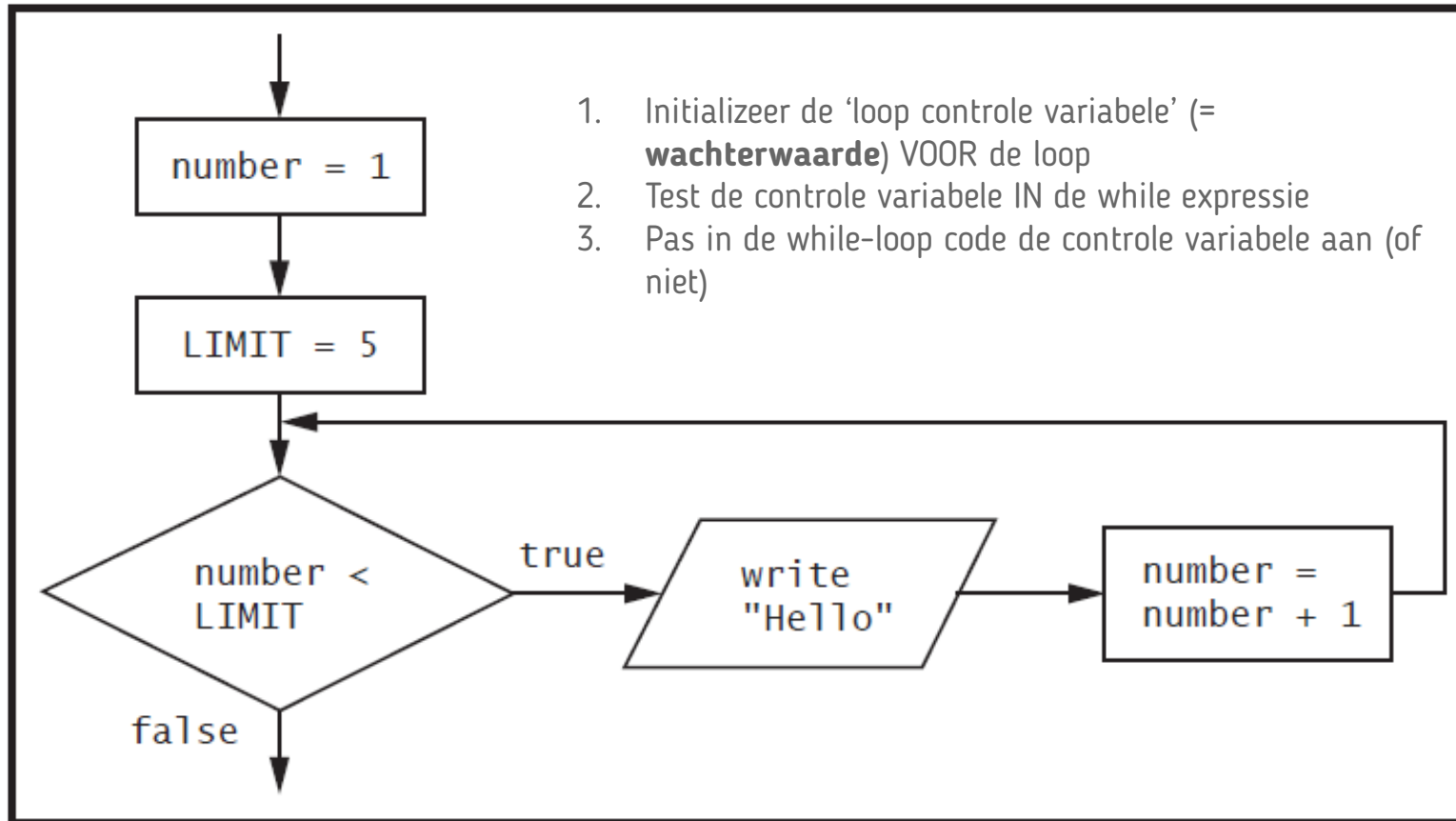


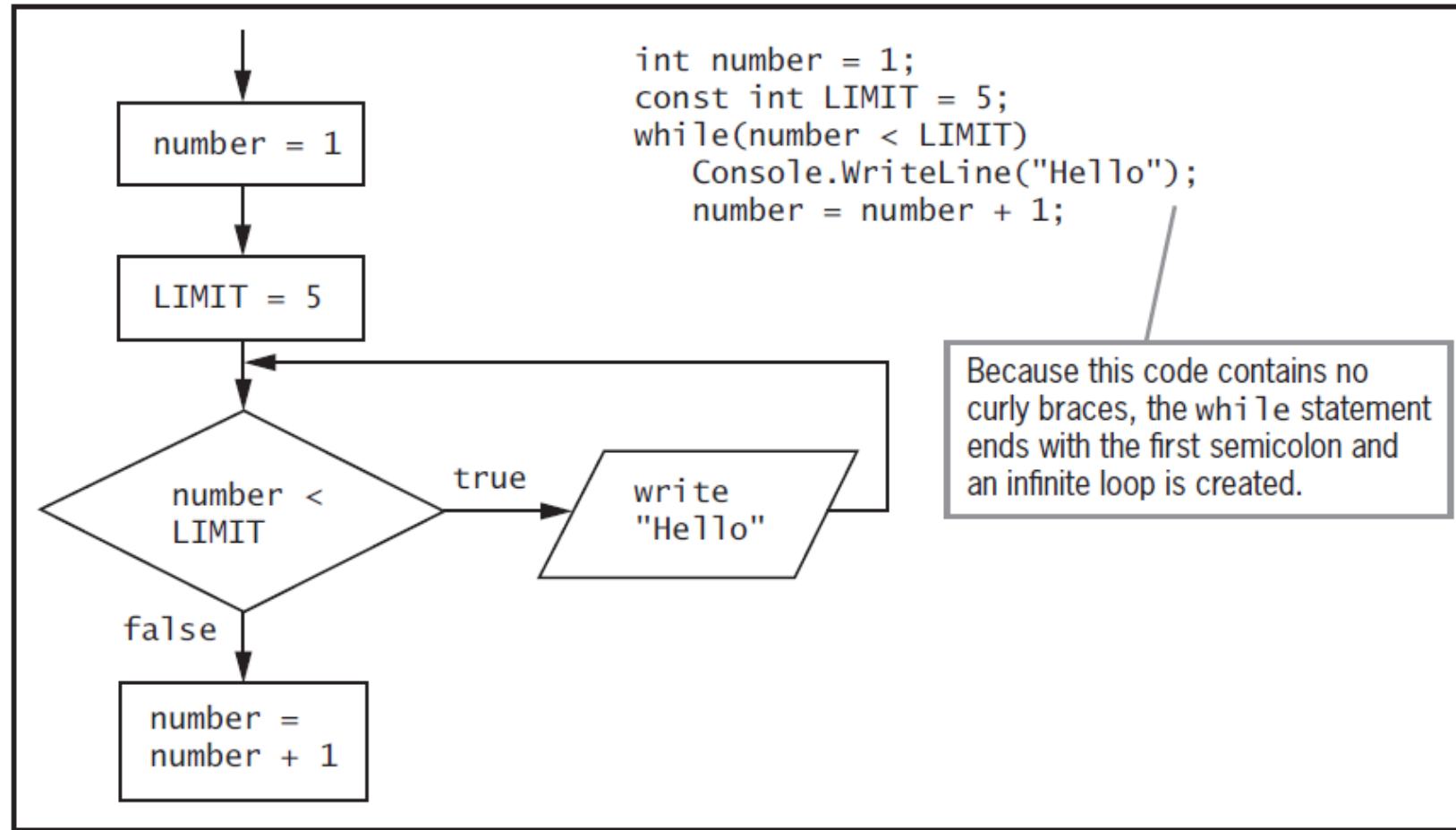
Figure 5-2 Flowchart for the logic of a while loop whose body executes four times

# Gebruik while loop

1. Initializeer de 'loop controle variabele' (= **wachterwaarde**) VOOR de loop
2. Test de controle variabele IN de while expressie
3. Pas in de while-loop code de controle variabele aan (of niet)

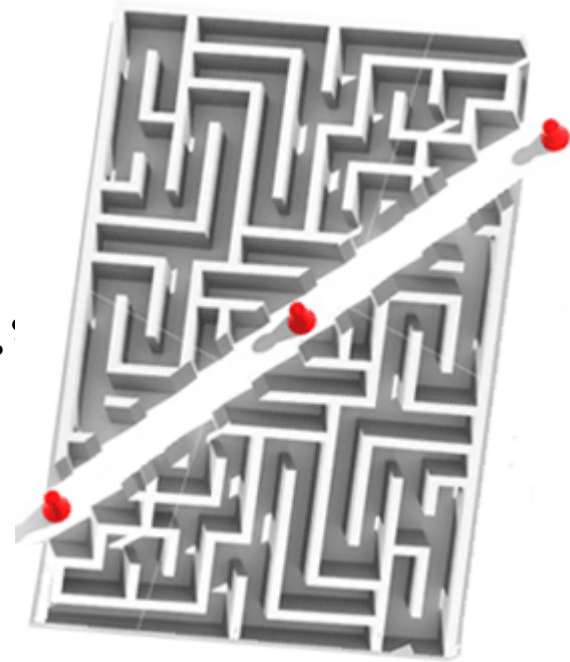
```
using System;
public class FourHellos
{
    public static void Main()
    {
        int number = 1;
        const int LIMIT = 5;
        while(number < LIMIT)
        {
            Console.WriteLine("Hello");
            number = number + 1;
        }
    }
}
```

**Figure 5-3** A program that contains a `while` loop whose body executes four times



# Visual studio shortcut

**“while <tab> <tab>”**





# Do while

# Do-while



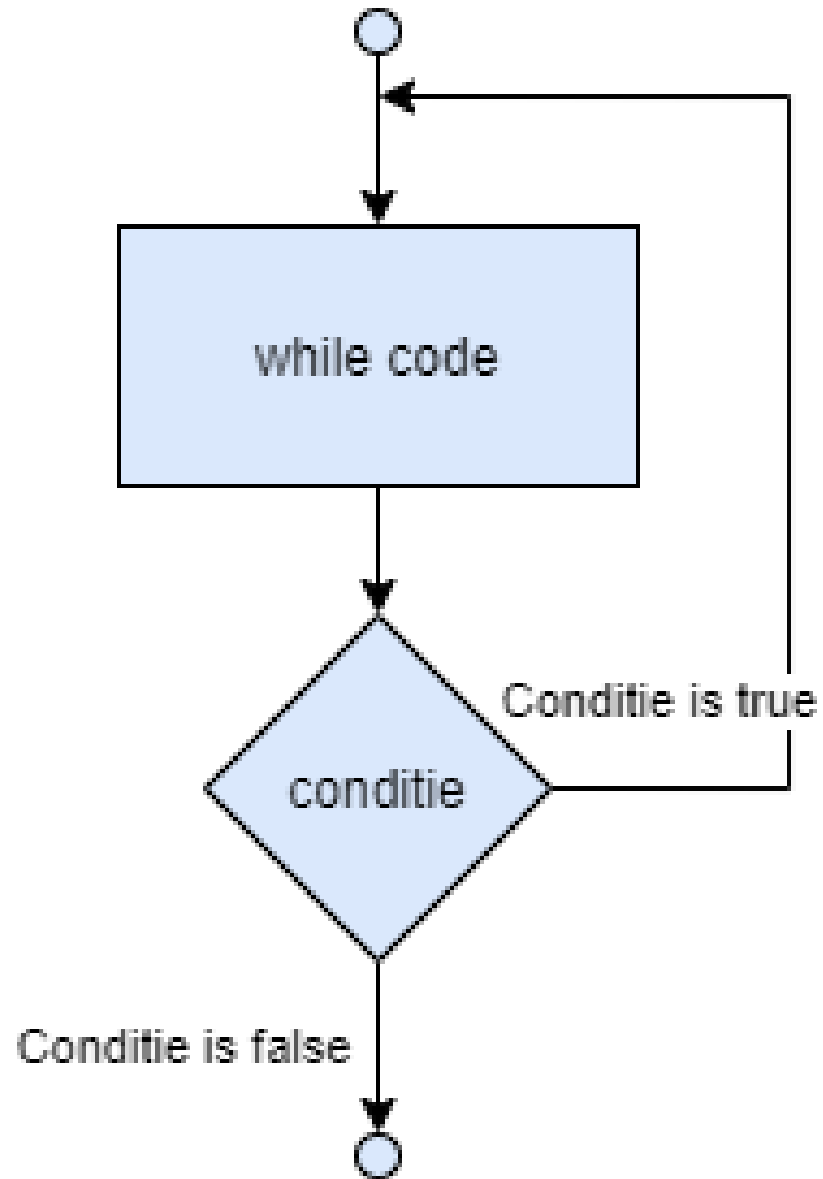
Test wordt **na** loop uitgevoerd



code in loop wordt  
**minstens 1 maal**  
uitgevoerd

*(bij while  
wordt code in  
loop mogelijk  
0 maal  
uitgevoerd)*

# Gebruik van de do while loop



# Syntax

```
int teller = 0;  
do  
{  
    Console.WriteLine(teller);  
    teller++;  
} while (teller < 10);
```

Geen ; vergeten na do-while!  
(niet bij while)

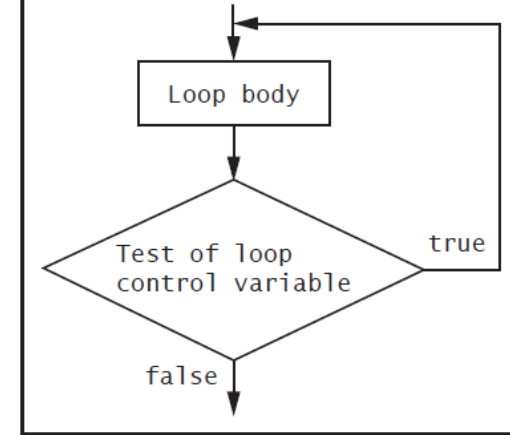
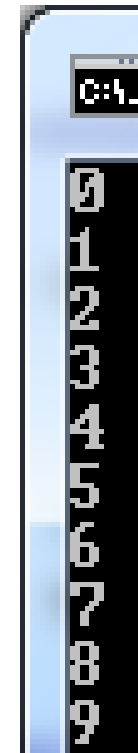
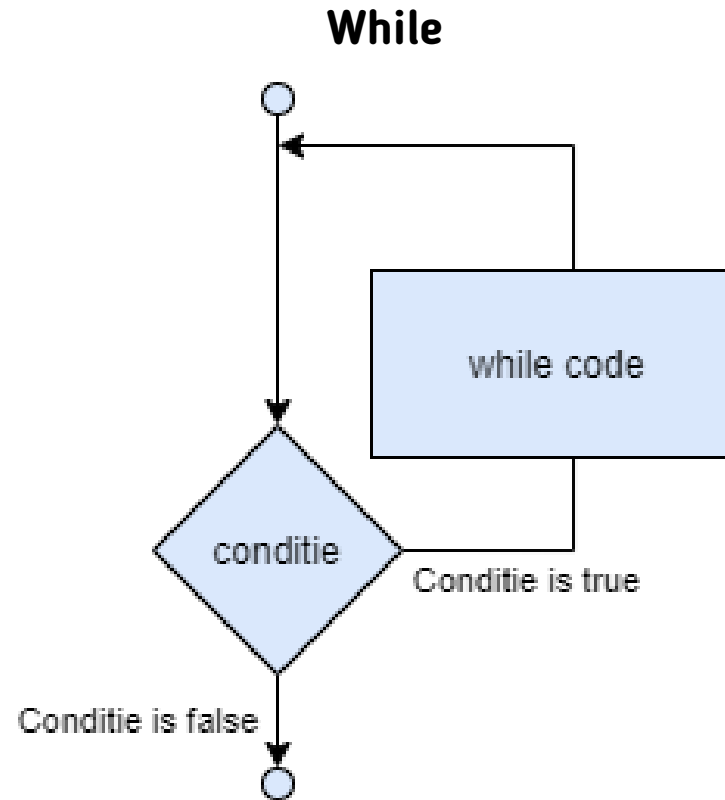
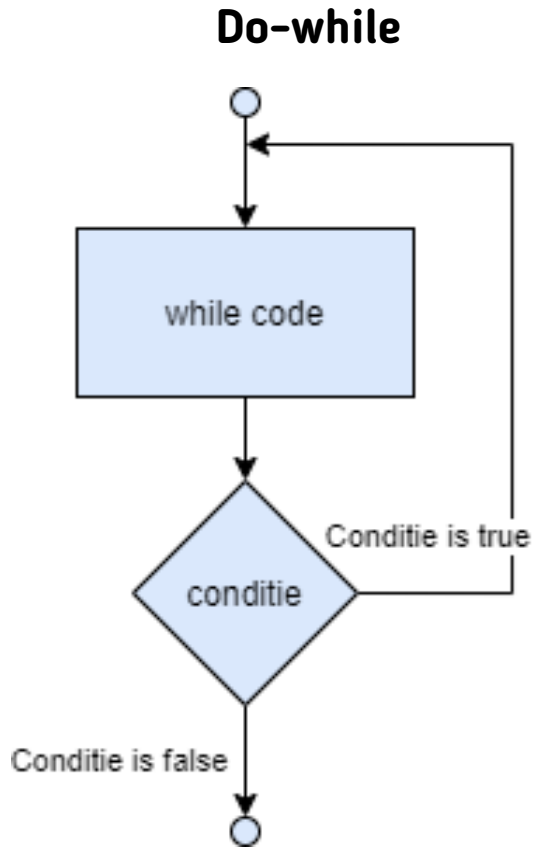
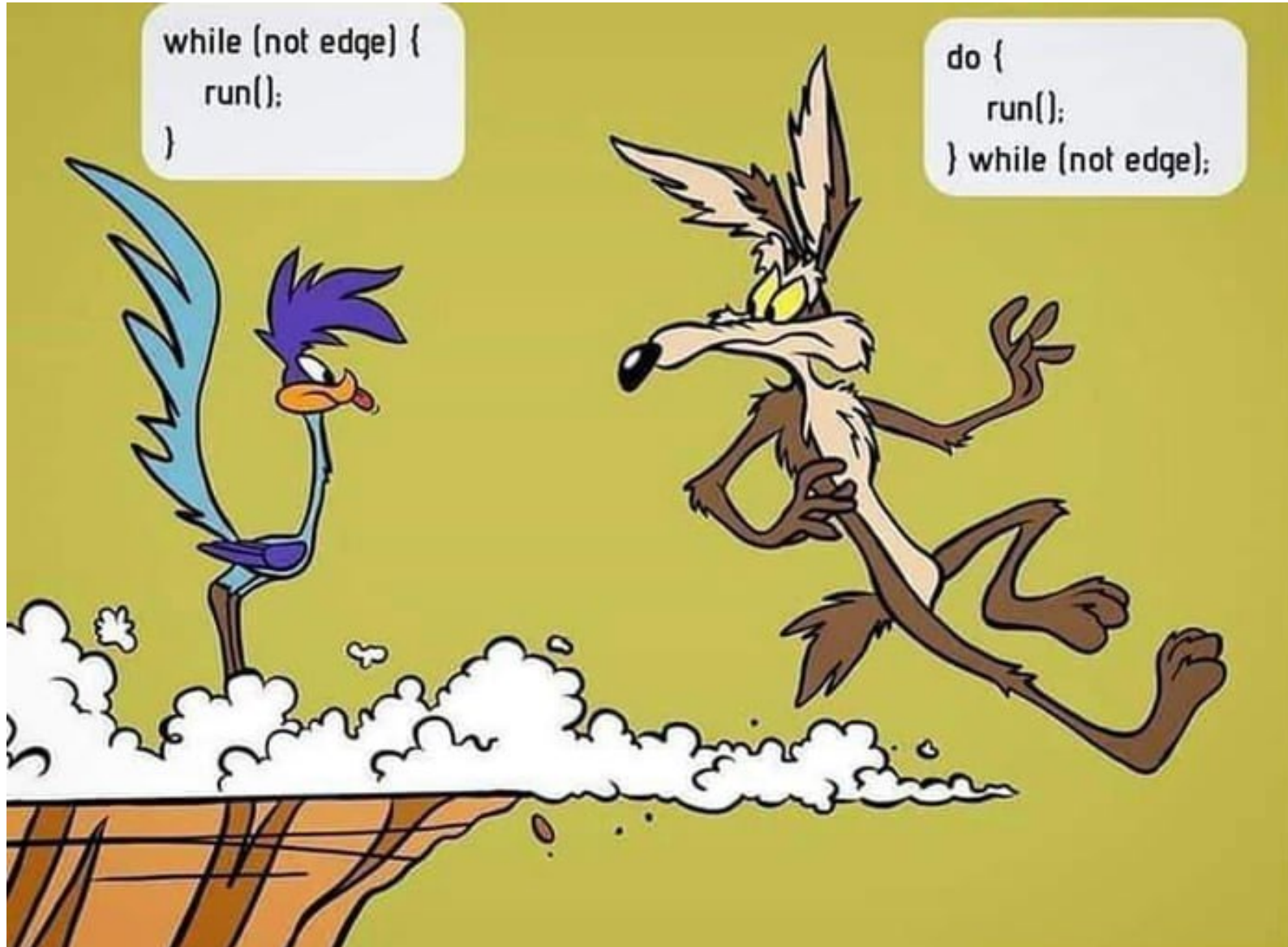


Figure 5-11 Flowchart of a do loop



# Do-while vs while





# Voorbeeld indefinite loop (kan ook met while)

```
1  string input;  
2  do  
3  {  
4      Console.WriteLine("Geef uw keuze in: a,b of c");  
5      input= Console.ReadLine();  
6  }while( input != "a"  && input != "b" && input != "c");
```

# Visual studio shortcut

**“do <tab> <tab>”**





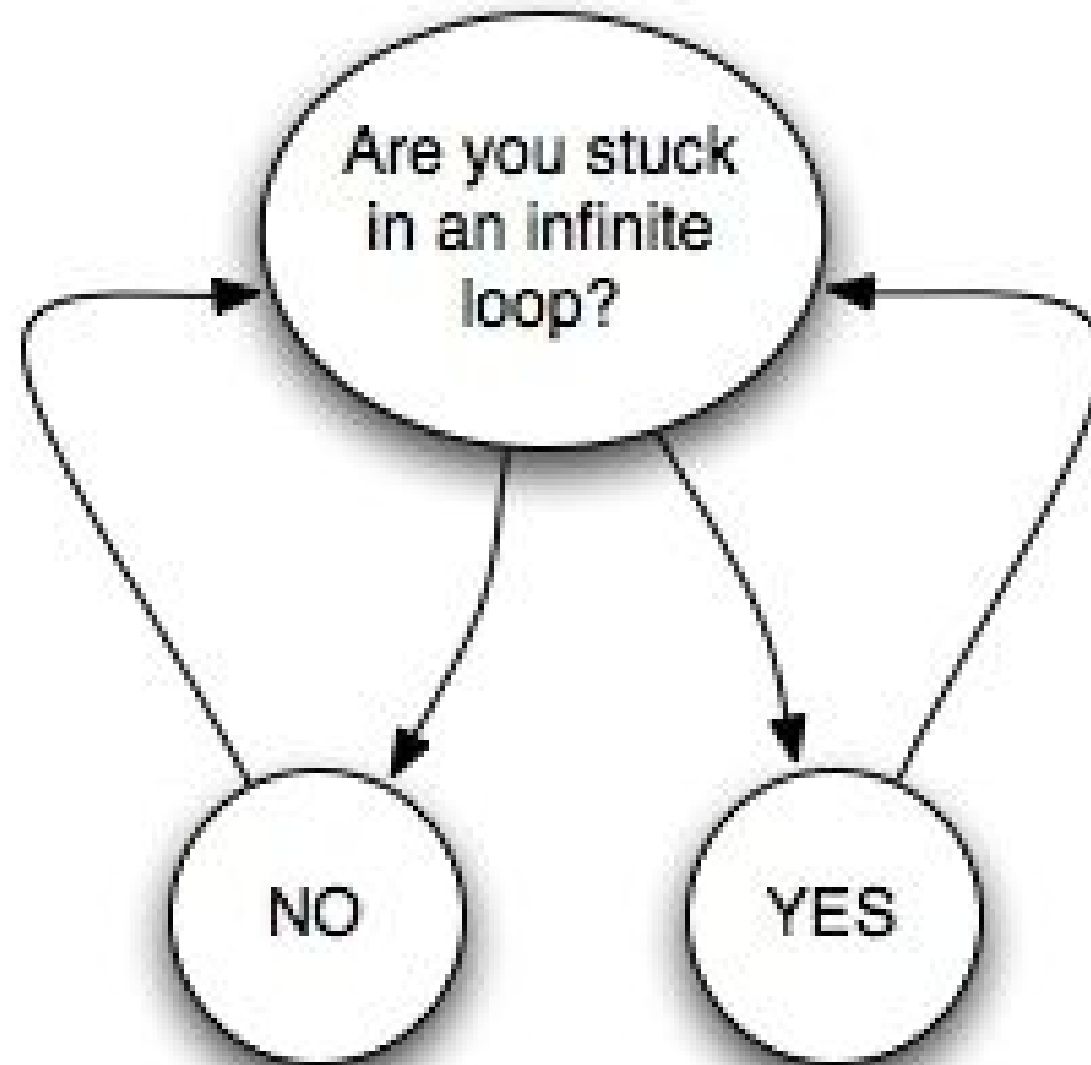
# Wanneer stopt dit programma?

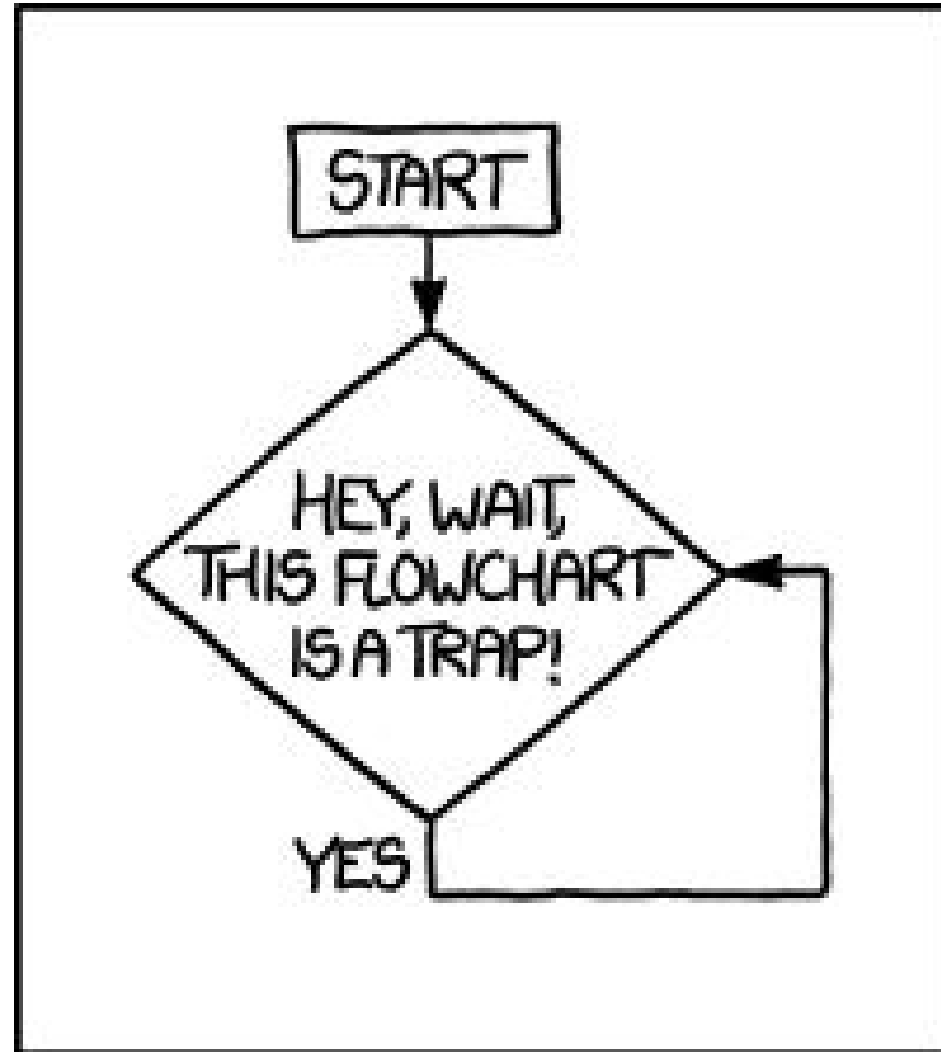
- Dit programma stopt nooit: de testconditie van de while-loop is altijd true!

```
do  
    Console.WriteLine ( "Hello mum" ) ;  
while ( true );
```

- Altijd opletten dat je geen oneindige loops programmeert!







# Demo time

- While
- Do While
- Scope bij loops



# 3. for

H6. Herhalingen, herhalingen, herhaling



# for loop

- Andere manier om eindige loops te schrijven

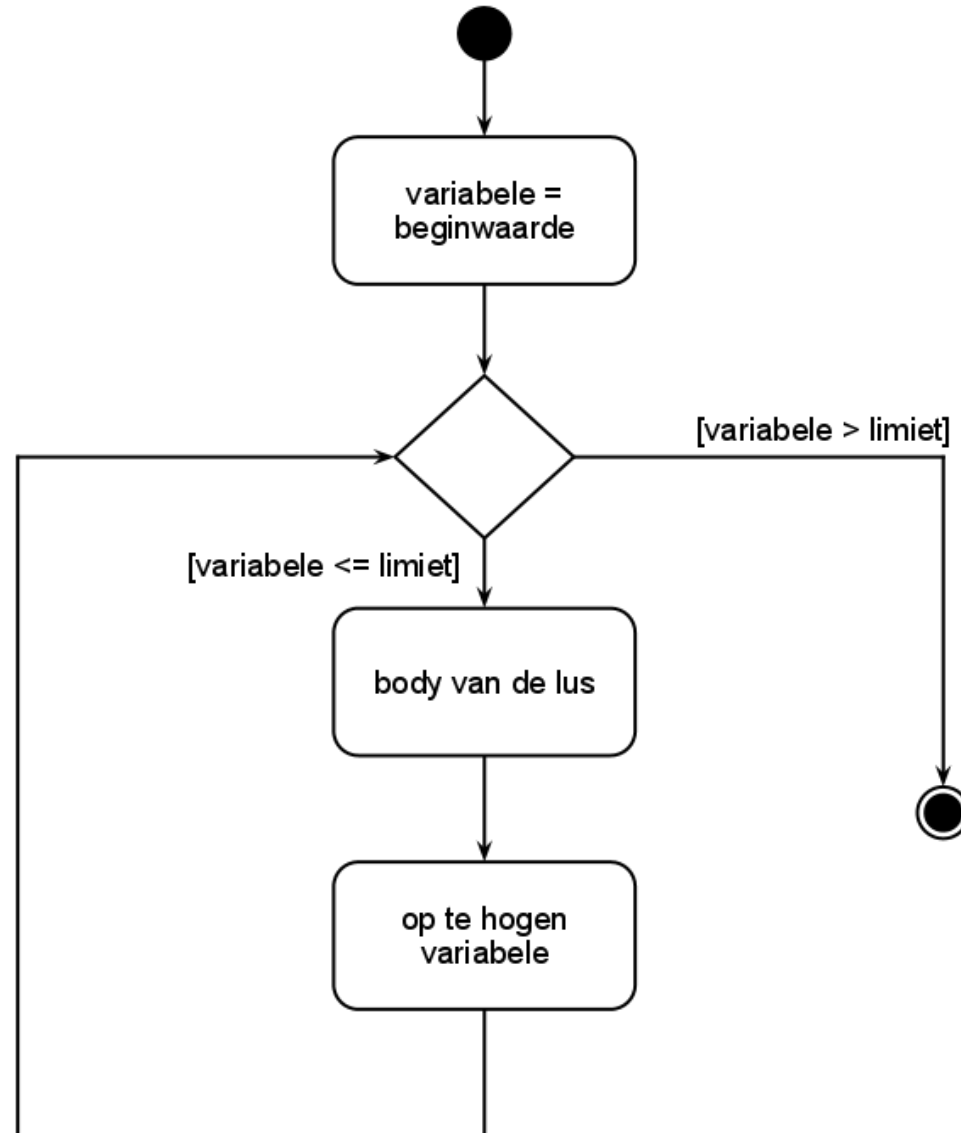
```
for ( setup ; finish test ; update ) {  
    things we want to do a given  
    number of times  
}
```

```
int i ;  
i = 1 ;  
while ( i < 11 ) {  
    Console.WriteLine ( "Hello mum" ) ;  
    i = i + 1 ;  
}
```

=

```
int i ;  
for ( i = 1 ; i < 11 ; i = i + 1 ) {  
    Console.WriteLine ( "Hello mum" ) ;  
}
```

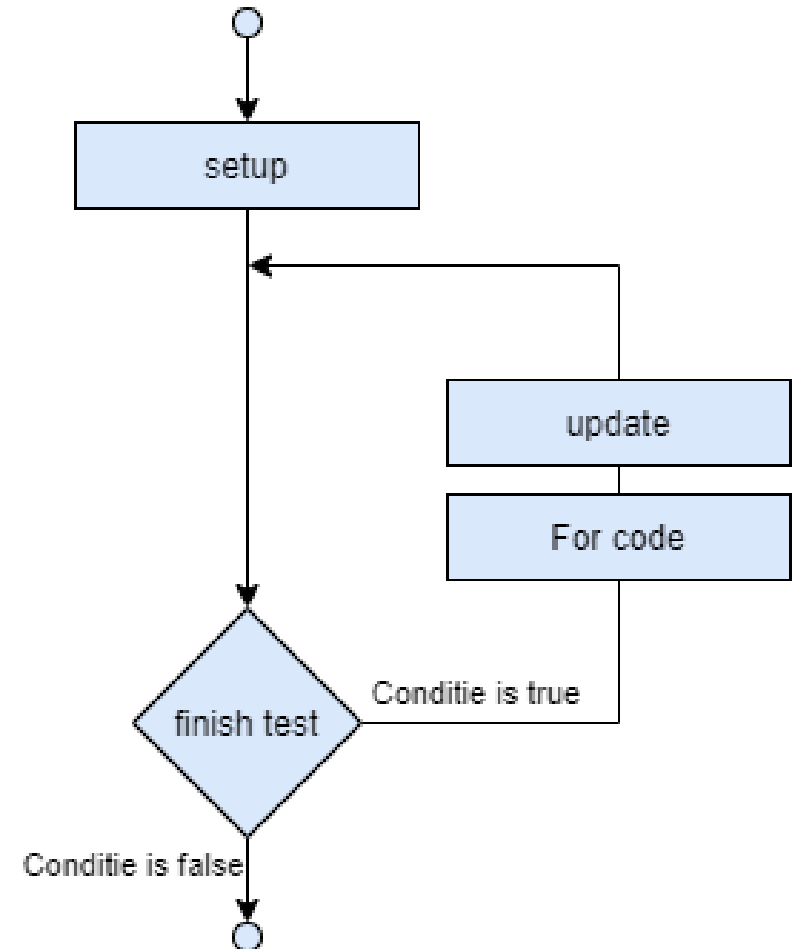
# for



# for

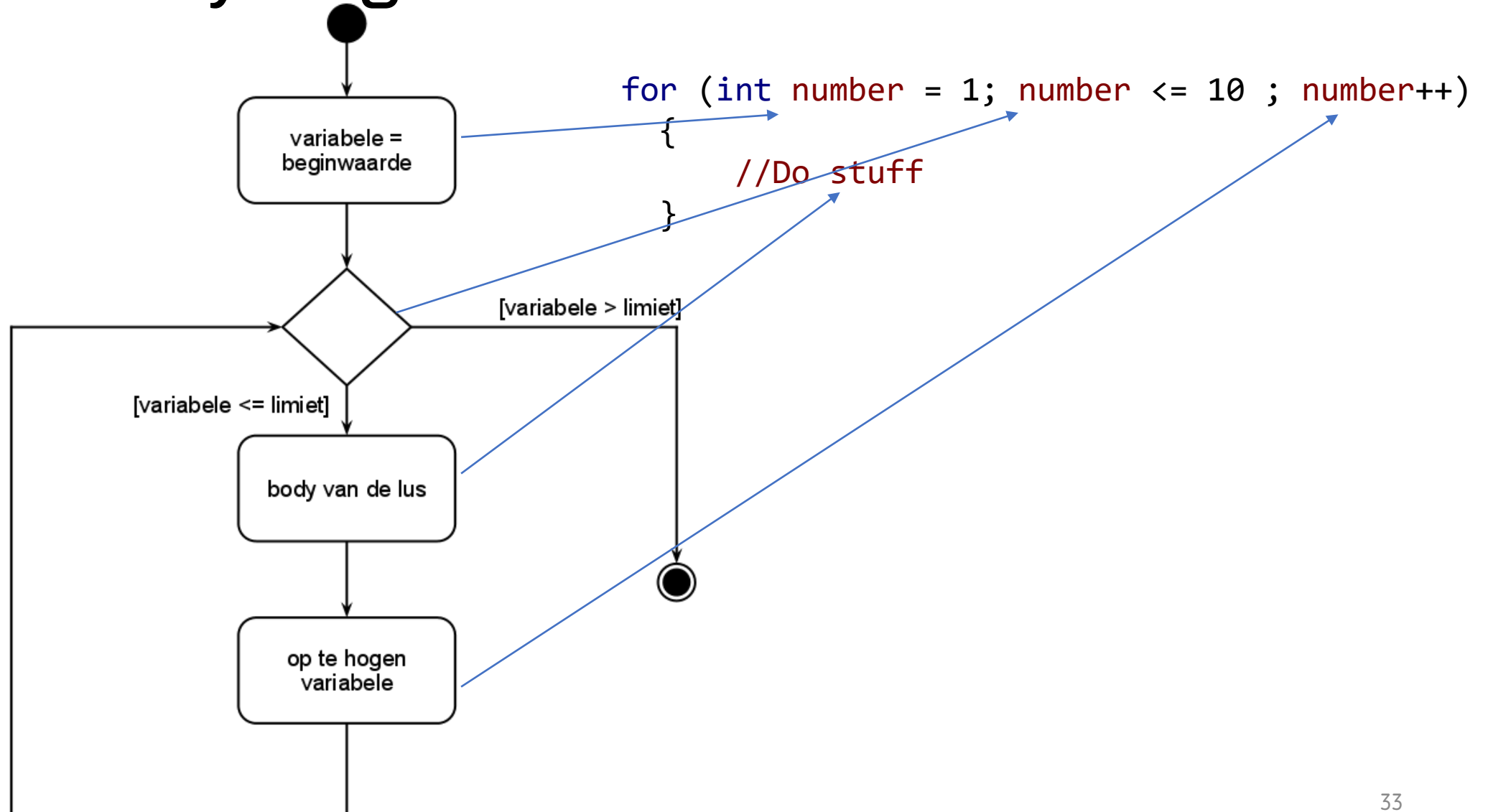
```
for ( setup ; finish test ; update ) {  
    things we want to do a given  
    number of times  
}
```

1. Plaats setupwaarde in de controle variabele
2. Test of de loops reeds beëindigd werd en indien dit het geval is, ga verder naar de uitdrukking na de loop
3. Anders, voer de uitdrukking binnen de loop uit
4. Update de controle variabele
5. Ga naar stap 2



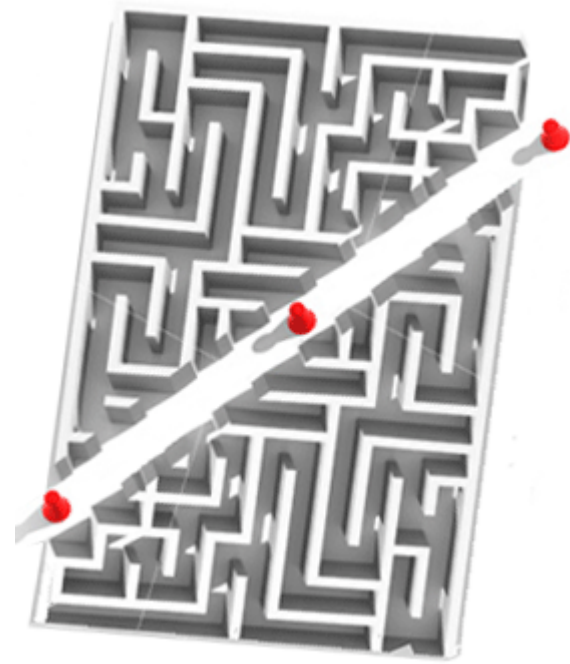


# for: activity-diagram



# Visual studio shortcut

**“for <tab> <tab>”**



# for

- *Nooit (of zelden) de lusvariabele in de lus zelf aanpassen!*

```
for (int number = 1; number <= 10 ; number++)  
{  
    //Do stuff  
  
    //But don't do this  
    number = 10;  
}
```

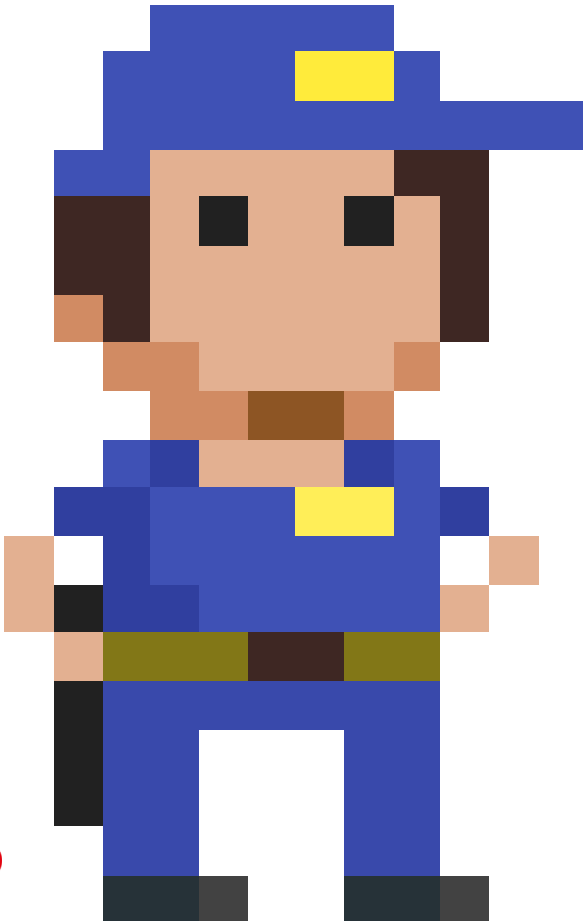


# Break & continue

- continue: huidige iteratie in loop stoppen
- break: volledige loop stoppen

```
for (int i = 1; i <= 10; i++)  
{  
    if (i == 5)  
    {  
        continue;  
    }  
    Console.WriteLine(i);  
}
```

- Vermijd deze keywords als de pest!



# conclusie

- Gebruik een `for`-lus als je op voorhand weet hoeveel keer de body moet uitgevoerd worden
- De lusvariabele (number) zou **nooit** mogen veranderd worden in de body
  - Hiervoor bestaat de `while` lus!
- De lusvariabele mag wel gelezen of geraadpleegd worden

# Verkorte notatie

- Komt veel voor bij for-syntax:

```
for (int i = 0; i < 10; i++)  
{  
    //Do cool stuff  
}
```

```
for (int i = 10; i > 0; i--)  
{  
    //Do cool stuff  
}
```

```
for (int i = 0; i < 10; i+=2)  
{  
    //Do cool stuff  
}
```

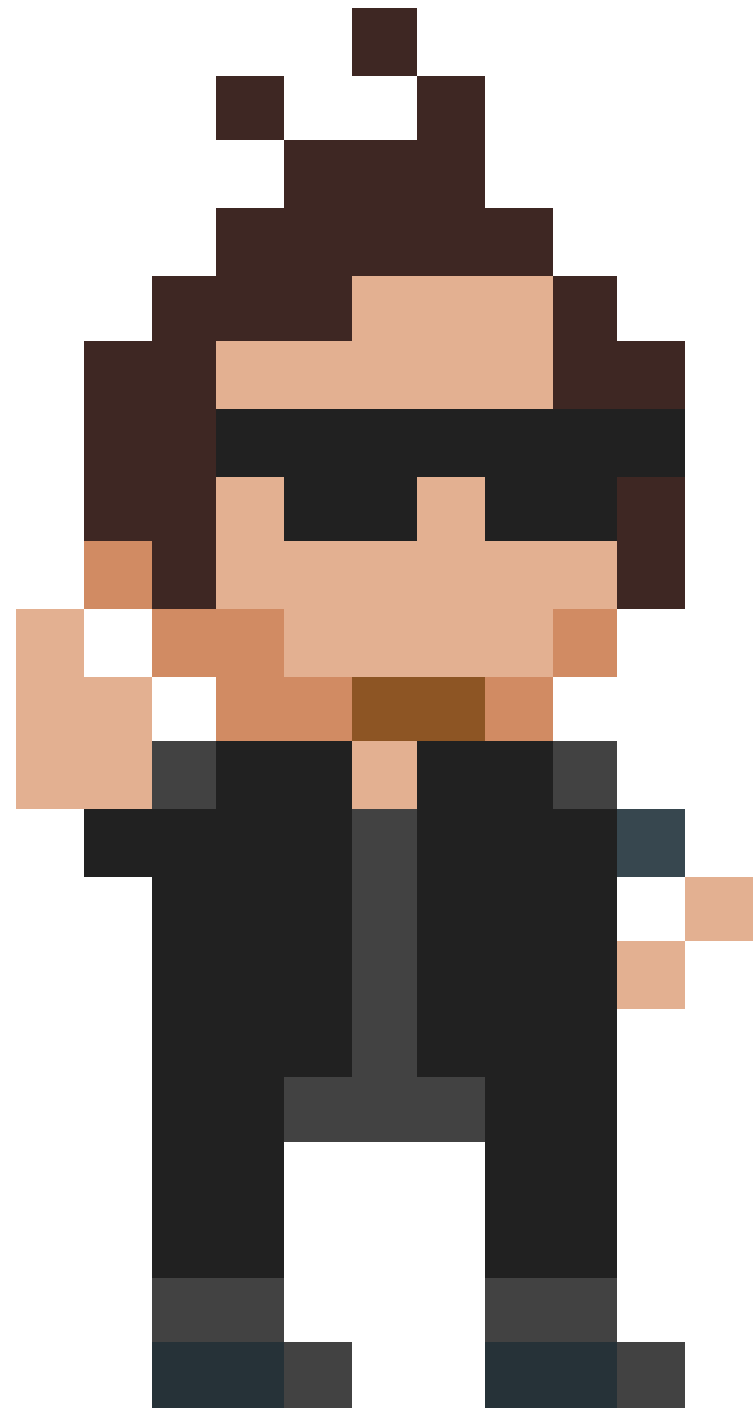
```
for (int i = 0; i < 10; i+=3)  
{  
    //Do cool stuff  
}
```

```
for (int i = 10; i > 0; i-=2)  
{  
    //Do cool stuff  
}
```

...

# Demo time

- For loop



# 4. Loop nesting

H6. Herhalingen, herhalingen, herhaling

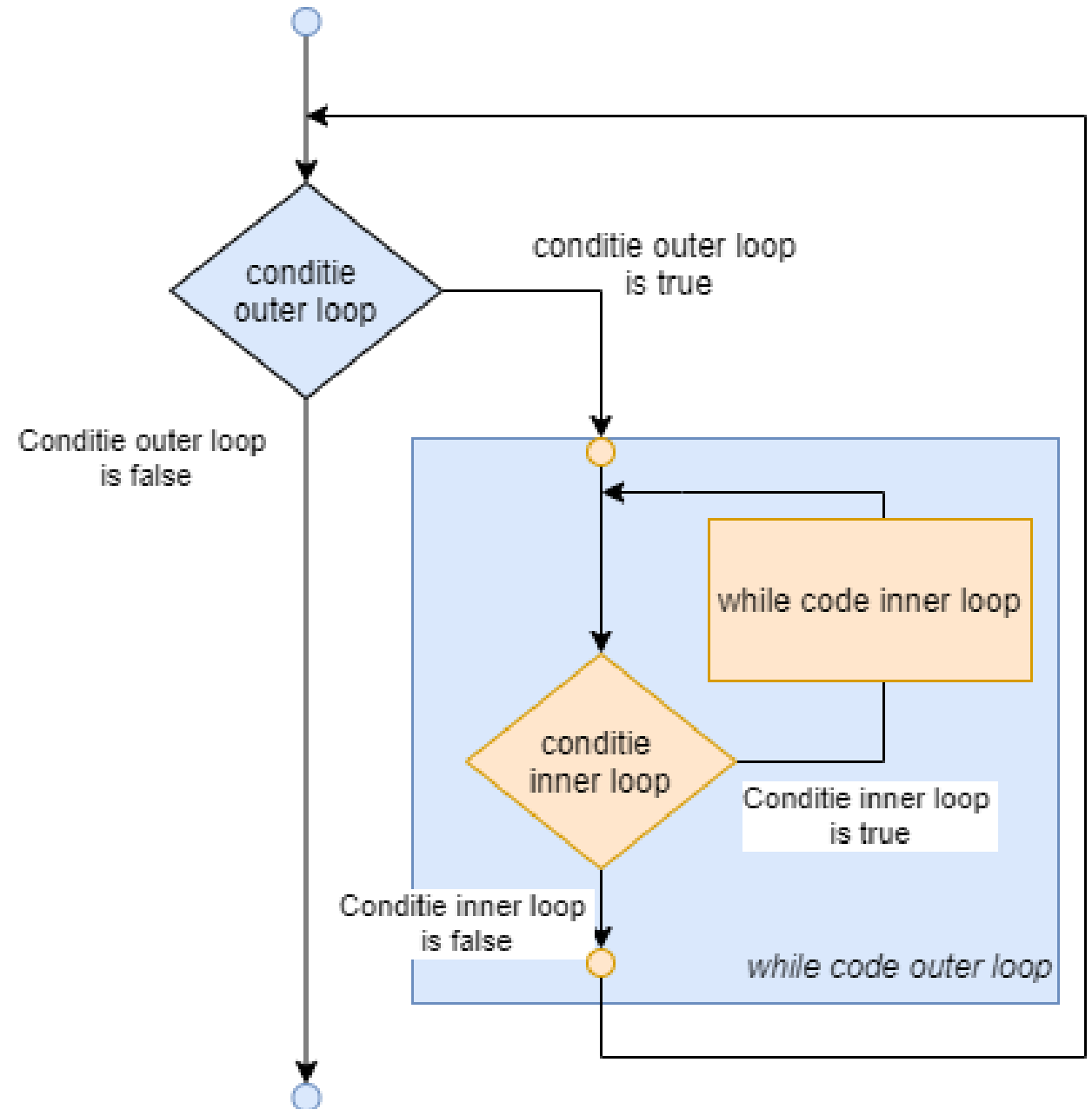




# Geneste loops

- Indien een loop in een andere loop staat spreken we van een “**inner**” en “**outer**” loop
  - Goed opletten bij het programmeren van deze structuren! (debuggen!)

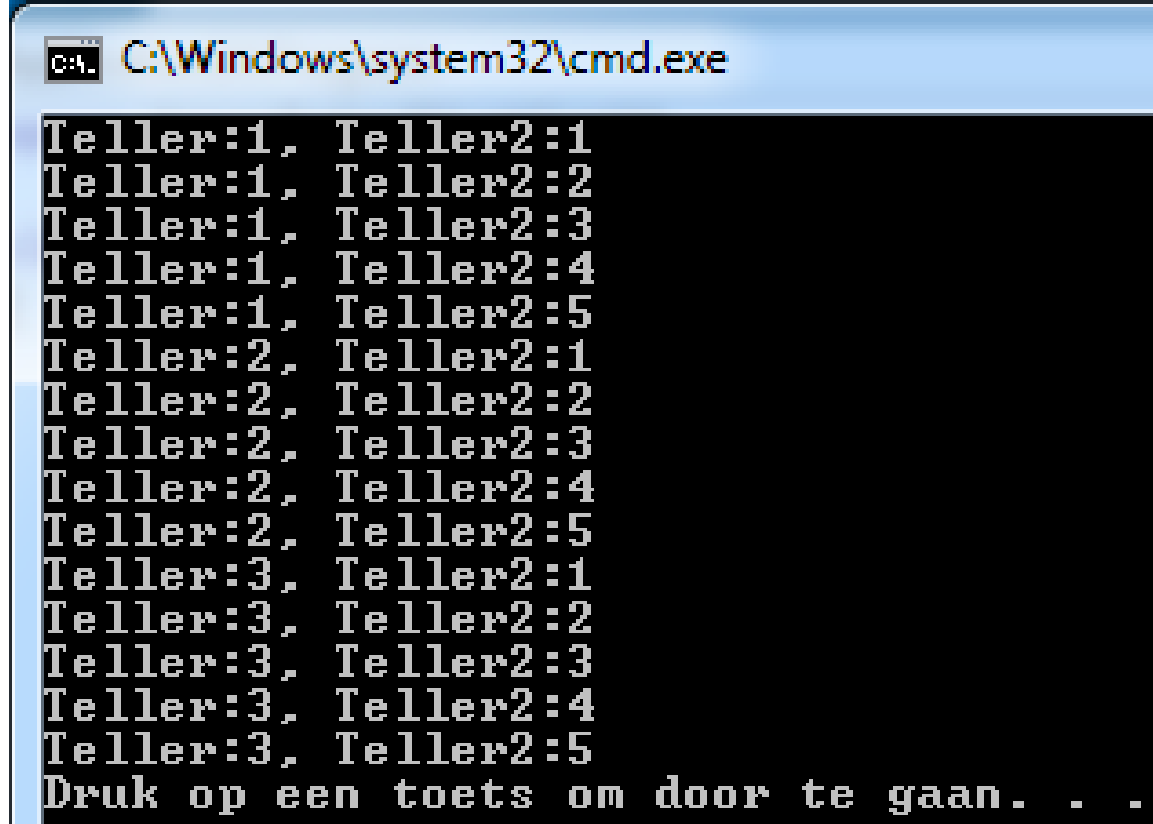
# Geneste loops



# Voorbeeld van geneste loops

```
int tellerA = 0;
int tellerB = 0;

while (tellerA < 3) //outer loop
{
    tellerA++;
    tellerB = 0;
    while (tellerB < 5) //inner loop
    {
        tellerB++;
        Console.WriteLine($"Teller:{tellerA}, Teller2:{tellerB}");
    }
}
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The output of the program is displayed as follows:

```
Teller:1, Teller2:1
Teller:1, Teller2:2
Teller:1, Teller2:3
Teller:1, Teller2:4
Teller:1, Teller2:5
Teller:2, Teller2:1
Teller:2, Teller2:2
Teller:2, Teller2:3
Teller:2, Teller2:4
Teller:2, Teller2:5
Teller:3, Teller2:1
Teller:3, Teller2:2
Teller:3, Teller2:3
Teller:3, Teller2:4
Teller:3, Teller2:5
Druk op een toets om door te gaan. . .
```

# Geneste loops tellen

- Vermenigvuldig het aantal keer iedere loop wordt uitgevoerd met mekaar.

- Bv: 

```
for (int i = 0; i < 10; i++)  
{  
    for (int j = 0; j < 5; j++)  
    {  
        Console.WriteLine("Hallo");  
    }  
}
```

- Outer loop= 10x
- Inner loop = 5x
  - “Hallo” zal 50keer op scherm verschijnen (5x10)

# Deze?

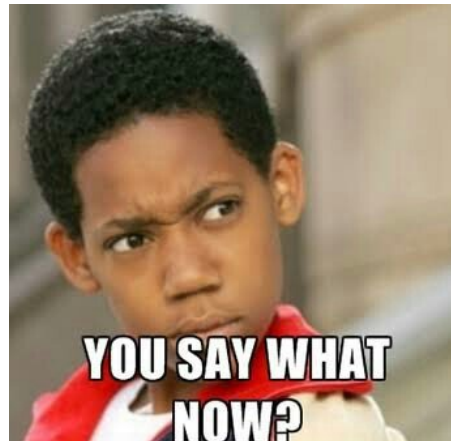
```
for (int i = 0; i < 3; i++)  
{  
    for (int j = 0; j < 10; j+=2)  
    {  
        for (int k = 0; k <= 6; k++)  
        {  
            Console.WriteLine("Hallo");  
        }  
    }  
}
```

- Loop 1 (outer loop): 3x
- Loop 2 (middelste): 5x (Let op de j+=2)
- Loop 3 (inner): 7x (Let op de k<=6 )
  - Totaal dus  $3 \times 5 \times 7 \Rightarrow$  **105 keer**

# Hoe vaak zal de inner code worden uitgevoerd?

```
int teller= 0;  
for (int j = 0; j < 4; j++)  
{  
  
    while (teller < 10)  
    {  
        Console.WriteLine("Hallo");  
        teller++;  
    }  
}
```

- Oplossing: 10 keer!



# Typische fout: inner teller niet reseten

- For doet dat voor ons tijdens setup gratis, maar een while niet! Dus moeten we manueel dit doen:
- Deze zal 40 keer op scherm getoond worden nu:

```
int teller = 0;
for (int j = 0; j < 4; j++)
{
    teller = 0;
    while (teller < 10)
    {
        Console.WriteLine("Hallo");
        teller++;
    }
}
```

break

