Programmeren 1 (C#)

Gerwin van Dijken (gerwin.vandijken@inholland.nl)

Programma periode 1.1 (Programmeren 1)

01 (wk-36)	Inleiding / Visual Studio 2017/2019
02 (wk-37)	Sequentie
03 (wk-38)	Selectie
04 (wk-39)	Iteratie
05 (wk-40)	Array's
06 (wk-41)	Methoden
07 (wk-42)	Herhaling / oefententamen
08 (wk-43)	*roostervrije week*
09 (wk-44)	praktijktentamen *(computer opdrachten)*
10 (wk-45)	-

Iteratie

- Iteratie → herhaling

  lus / loop

- Meerdere keren uitvoeren van dezelfde code
  a) *vast aantal keren herhalen of*
  b) *zolang aan een bepaalde voorwaarde wordt voldaan*

3 soorten iteratie structuren

- **while loop**

  zolang (boolean) conditie True is, doe { … }

- **do-while loop**

  doe { … } zolang (boolean) conditie True is

- **for loop**

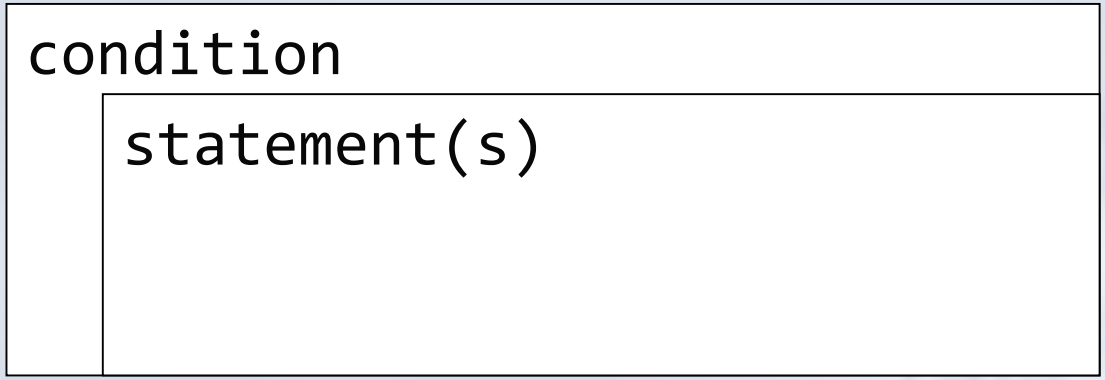
  doe een vast aantal keren { … }

while loop

pseudocode

```
while <condition>
      <statement(s)>
```

PSD



- statement(s) N keer uitgevoerd (N>=0, mogelijk 0 keer)

while loop

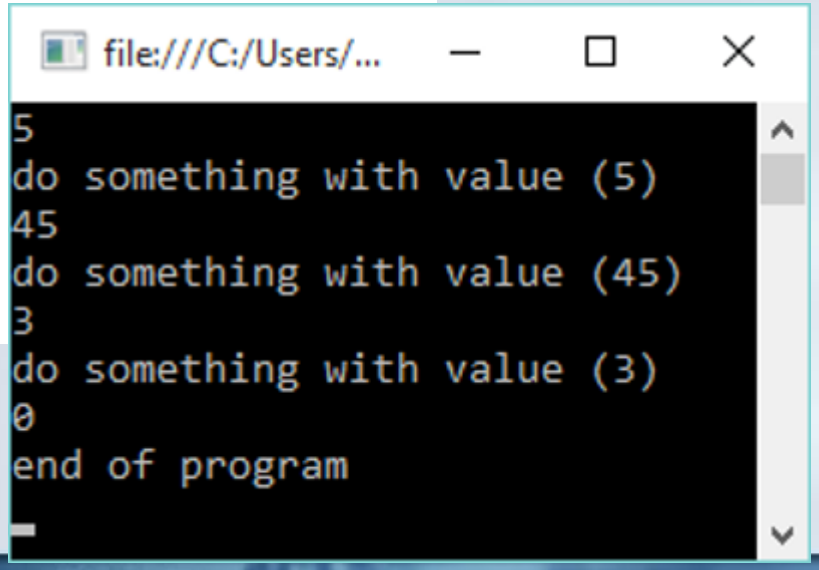
- *Testen op een specifieke (stop) waarde*

```
read value
while value <> 0
      // do something with value...
      read value
display "end of program"
```

while loop in C#

- *Testen op een specifieke (stop) waarde*

```
static void Main(string[] args)
{
  // read (first) value
  int value = int.Parse(Console.ReadLine());

  while (value != 0)    // 0 = stop value
  {
    Console.WriteLine("do something with value ({0})", value);

    // read (next) value
    value = int.Parse(Console.ReadLine());
  }

  Console.WriteLine("end of program");
}
```



while loop

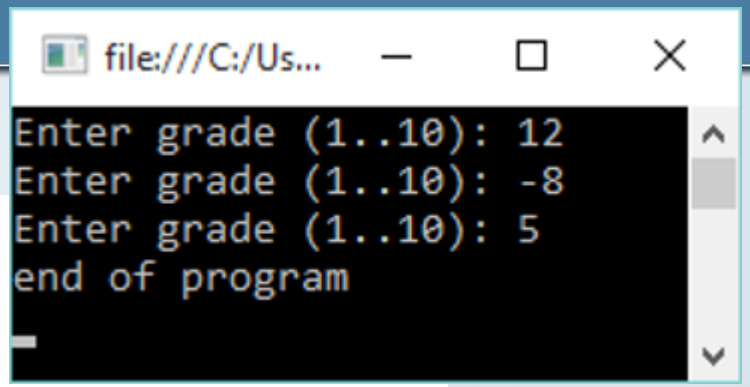
- *Testen op een boolean-waarde*

```
validGrade = false
while not validGrade
      display "enter grade (1..10)"
      read grade
      validGrade = (grade >= 1) AND (grade <= 10)
display "end of program"
```

while loop in C#

- *Testen op een boolean-waarde*

```
static void Main(string[] args)
{
  bool validGrade = false;
  int grade = 0;

  while (!validGrade)
  {
    // read grade
    Console.Write("Enter grade (1..10): ");
    grade = int.Parse(Console.ReadLine());

    // grade is valid?
    validGrade = (grade >= 1) && (grade <= 10);
  }

  // now we can do something with the (valid) grade...
  Console.WriteLine("end of program");
}
```

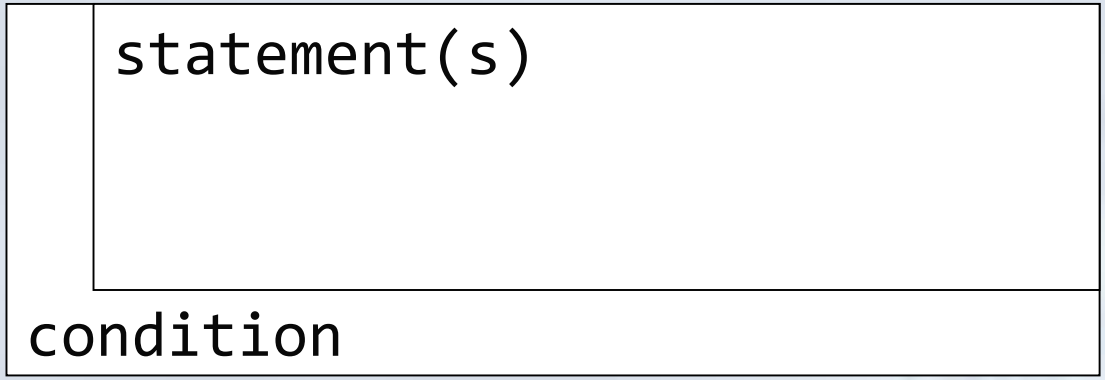


do-while loop

pseudocode

```
do
        <statement(s)>
while <condition>
```

PSD



- statement(s) N keer uitgevoerd (N>=1, minimaal 1 keer)

do-while loop in C#

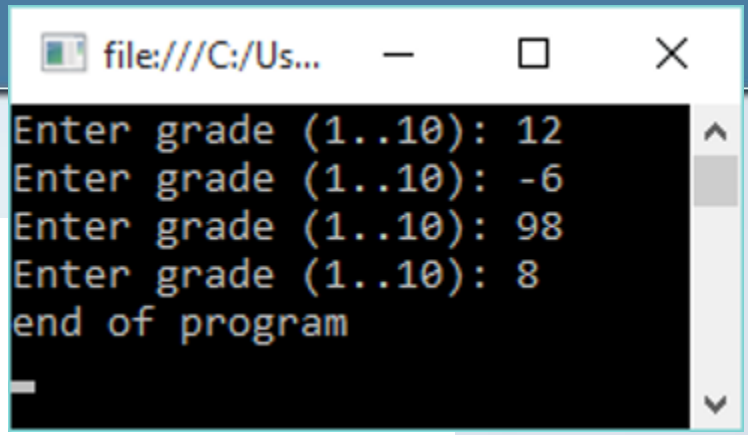
- Minstens 1 keer uitgevoerd

```
validGrade = false
do
      display "enter grade (1..10)"
      read grade
      validGrade = (grade >= 1) AND (grade <= 10)
while not validGrade
display "end of program"
```

do-while loop in C#

- Minstens 1 keer uitgevoerd

```
static void Main(string[] args)
{
  bool validGrade = false;
  int grade = 0;

  do
  {
    // read grade
    Console.Write("Enter grade (1..10): ");
    grade = int.Parse(Console.ReadLine());

    // grade is valid?
    validGrade = (grade >= 1) && (grade <= 10);
  } while (!validGrade);

  // now we can do something with the (valid) grade...
  Console.WriteLine("end of program");
}
```



Oefening – richtgetal

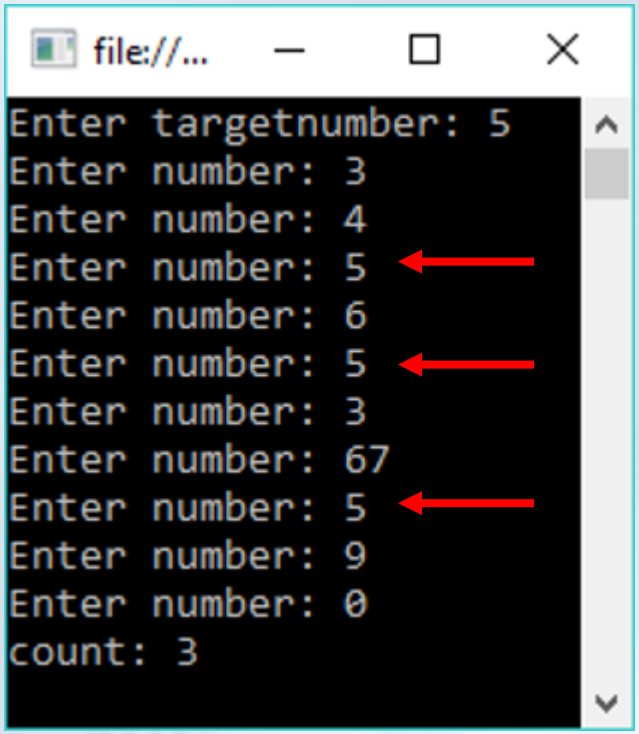
- Lees eerst een richtgetal. Lees vervolgens een (onbekend) aantal getallen (einde invoer is 0) en bepaal het aantal getallen dat gelijk is aan het richtgetal. Toon dit aantal.

Oefening – richtgetal

```
read targetNumber
count = 0
read number
while number <> 0
      if number = targetNumber
            count = count + 1
      read number
display count
```

Oefening – C# code

```
static void Main(string[] args)
{
    Console.Write("Enter targetnumber: ");
    string input = Console.ReadLine();
    int targetNumber = int.Parse(input);

    int count = 0;

    // read (first) number
    Console.Write("Enter number: ");
    int number = int.Parse(Console.ReadLine());

    while (number != 0)
    {
        if (number == targetNumber)
        {
            count++;
        }

        // read (next) number
        Console.Write("Enter number: ");
        number = int.Parse(Console.ReadLine());
    }

    Console.WriteLine("count: {0}", count);

    Console.ReadKey();
}
```



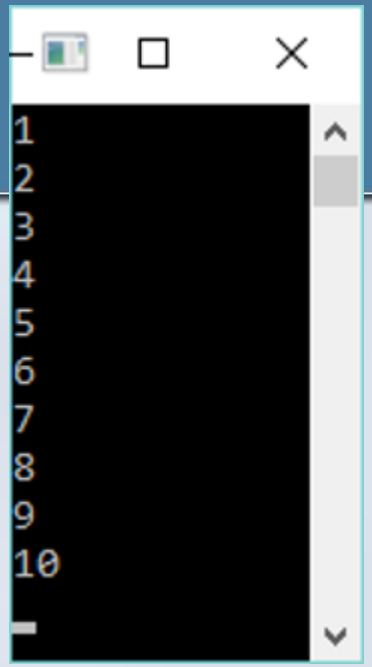
for loop

```
for i = 1 to 10
      <statement(s)>
```

for loop in C#

- Vast aantal keer herhalen

```
static void Main(string[] args)
{
  // repeat 10 times
  for (int i = 1; i <= 10; i++)
  {
    Console.WriteLine(i);
  }
}
```



3. update

2. condition

1. initialisation

for loop in C#

- for loop vs while loop

```
static void Main(string[] args)
{
  // repeat 10 times
  for (int i = 1; i <= 10; i++)
  {
    Console.WriteLine(i);
  }
}
```

```
static void Main(string[] args)
{
  // 1. initialisation
  int i = 1;

  // 2. condition
  while (i <= 10)
  {
    Console.WriteLine(i);

    // 3. update
    i++;
  }
}
```

for loop in C#

- Wat is de uitvoer?

```
static void Main(string[] args)
{
  // repeat 10 times
  for (int i = 1; i <= 10; i++)
  {
    Console.Write(i + " ");
  }
}
```

```
1 2 3 4 5 6 7 8 9 10
```

```
static void Main(string[] args)
{
  // repeat 10 times
  for (int i = 0; i < 10; i++)
  {
    Console.Write(i + " ");
  }
}
```

```
0 1 2 3 4 5 6 7 8 9
```

```
static void Main(string[] args)
{
  for (int i = 2; i <= 20; i += 2)
  {
    Console.Write(i + " ");
  }
}
```

```
2 4 6 8 10 12 14 16 18 20
```

```
static void Main(string[] args)
{
  for (int i = 1; i <= 20; i += 3)
  {
    Console.Write(i + " ");
  }
}
```

```
1 4 7 10 13 16 19
```

Oefening - kwadraten

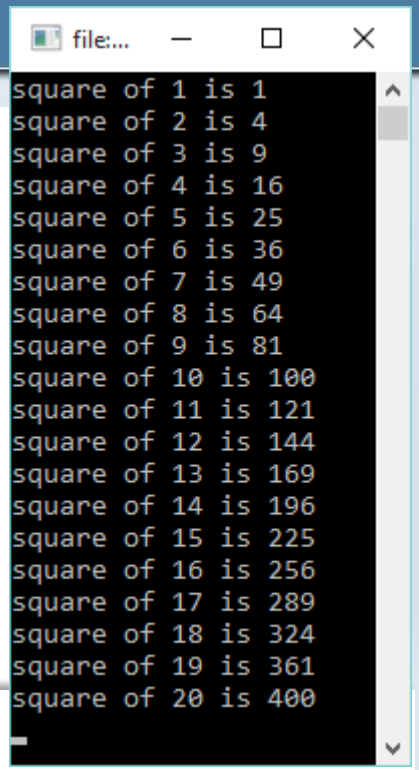
- Geef van de getallen 1 t/m 20 de kwadraten (gebruik hiervoor een lus).

Oefening - kwadraten

```
number = 1
while number <= 20
      square = number * number
      display square
      number = number + 1


for number = 1 to 20
      square = number * number
      display square
```

Oefening - kwadraten

```
static void Main(string[] args)
{
  int number = 1;
  while (number <= 20)
  {
    int square = number * number;
    Console.WriteLine("square of {0} is {1}", number, square);
    number++;
  }

  Console.ReadKey();
}
```




```
static void Main(string[] args)
{
  for (int number = 1; number <= 20;  number++)
  {
    int square = number * number;
    Console.WriteLine("square of {0} is {1}", number, square);
  }

  Console.ReadKey();
}
```

Geneste loop

```
while <condition>
      statement(s)
      while <condition>
            statement(s)


for i=1 to 10
      statement(s)              // 10x
      for j=1 to 20
            statement(s)        // 200x
```

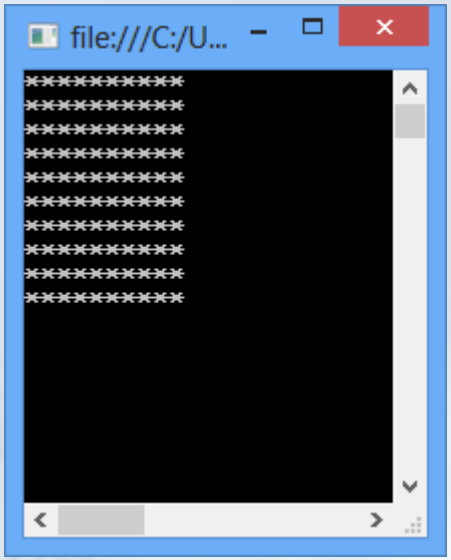
Geneste loop – voorbeeld 1

```
read number
while number > 0
      i = 1
      while i <= number
            display '*'
            i = i + 1
      read number


read number
while number > 0
      for i=1 to number
            display '*'
      read number
```

Geneste loop – voorbeeld 2

- Wat is hiervan de uitvoer?

```
static void Main(string[] args)
{
  for (int i = 1; i <= 10; i++)
  {
    for (int j = 1; j <= 10; j++)
    {
      Console.Write("*");
    }
    Console.WriteLine();
  }
}
```



Oefening – priemgetal

- Lees een getal en bepaal of het een priemgetal is (of niet).

Oefening – priemgetal

```
read number
isPrime = true
i = 2
while i < number AND isPrime
      if number modulo i = 0
            isPrime = false
      else
            i = i + 1

if isPrime
      display "prime number"
else
      display "no prime number"
```

Oefening – priemgetal

```
static void Main(string[] args)
{
  // read number
  int number = int.Parse(Console.ReadLine());

  // determine if number is a prime number
  bool isPrime = true;
  int i = 2;
  while ((i < number) && isPrime)
  {
    if ((number % i) == 0)
      isPrime = false;
    else
      i = i + 1;
  }

  if (isPrime)
    Console.WriteLine("{0} is a prime number", number);
  else
    Console.WriteLine("{0} is not a prime number", number);

  Console.ReadKey();
}
```

Huiswerk

- Lezen
  - Yellow Book → zie Moodle

- (praktijk) Programmeren 1 *(deze week)*
  - week 4 opdrachten