

***SOFTWARE ONTWIKKELEN***



# AGENDA VOOR VANDAAG

- Lesdoelen
- Lastige opdrachten?
- Herhalingsstructuren – While loop
- C# functionaliteiten (Math, Random)
- Leerstrategie: verwerking

# ***LESDOELEN***

Na de les van vandaag:

- Begrijp je het concept 'while-lus' en kun je deze toepassen in C#
- Ken je een aantal C# functionaliteiten (Math, Random) en kun je deze toepassen

# ***LASTIGE OPDRACHTEN - QUIZES?***

- Feedback n.a.v gemaakte quizzes.
- Waar liep je tegen aan?
- Wat viel op?



# ***VORIGE LES***

- Neem de vraag die je jezelf gesteld had en schrijf een zo volledig mogelijk antwoord op
  - Is dit antwoord beter dan die van de vorige les?

# ***WHILE - LOOP***

Een while-loop in C# gebruik je wanneer je niet van tevoren weet hoe vaak een codeblok moet worden herhaald, terwijl een for-loop beter geschikt is als je het aantal iteraties vooraf kunt bepalen (denk aan het aantal maanden in een jaar).

Kun je een voorbeeld bedenken?

# ***WHILE – LOOP (DISCUSSIE)***

Stel je voor dat je een programma schrijft om te vragen om een correcte gebruikersinvoer, zoals een wachtwoord. Je weet niet van tevoren hoe vaak de gebruiker het verkeerde wachtwoord zal invoeren, hoe zou deze pseudo-code er uit zien?

# WHILE - LOOP

```
string correctPassword = "geheim";
string userInput = "";

while(userInput != correctPassword)
{
    Console.WriteLine("Voer je wachtwoord in:");
    userInput = Console.ReadLine();

    if(userInput != correctPassword)
    {
        Console.WriteLine("Fout wachtwoord, probeer opnieuw.");
    }
}

Console.WriteLine("Correct wachtwoord ingevoerd!");
```



# WHILE – LOOP (OPDRACHT)

Maak een programma waarin de gebruiker een getal moet raden dat door de computer willekeurig is gekozen tussen 1 en 100. De gebruiker mag maximaal 10 pogingen doen om het getal te raden. Na elke poging moet het programma aangeven of het geraden getal te hoog, te laag, of correct is.

Neem in eerste instantie een vast getal in de code, bijvoorbeeld 50.

Welke stappen zijn er in pseudo-code?

```
Raad het getal tussen 1 en 100:  
Jouw gok: 50  
Te laag! Je hebt nog 9 pogingen over.  
  
Jouw gok: 75  
Te hoog! Je hebt nog 8 pogingen over.  
  
Jouw gok: 60  
Correct! Je hebt het geraden in 3 pogingen!
```

# ***RANDOM KLASSE***

De Random klasse in C# wordt gebruikt om willekeurige getallen te genereren.

Dit is handig voor het "Raad het getal"-spel, omdat de computer een willekeurig getal tussen 1 en 100 moet kiezen dat de gebruiker moet raden.

Pas je programma aan.

```
Random random = new Random();  
int randomNumber = random.Next(1, 101); // Getal tussen 1 en 100
```

# MATH KLASSE

De Math klasse in C# biedt een breed scala aan wiskundige functies. Wat voorbeelden:

```
// Math.Abs(x) - Absolute waarde
int negatiefGetal = -5;
int absoluteWaarde = Math.Abs(negatiefGetal);
Console.WriteLine("De absolute waarde van -5 is: " + absoluteWaarde);

// Math.Max(x, y) - Grootste van twee getallen
int getal1 = 3;
int getal2 = 7;
int grootste = Math.Max(getal1, getal2);
Console.WriteLine("Het grootste getal tussen 3 en 7 is: " + grootste);

// Math.Min(x, y) - Kleinste van twee getallen
int kleinste = Math.Min(getal1, getal2);
Console.WriteLine("Het kleinste getal tussen 3 en 7 is: " + kleinste);
```



# ***DO ... WHILE - LOOP***

De **do...while**-lus is vergelijkbaar met een normale **while**-lus, maar met één belangrijk verschil: een **do...while**-lus voert de code minstens één keer uit voordat de voorwaarde wordt gecontroleerd. Dit komt omdat de voorwaarde pas na de uitvoering van het codeblok wordt geëvalueerd.

```
do
{
    // Code om uit te voeren
} while (voorwaarde);
```

# ***DO ... WHILE - LOOP***

```
int getal;

// Do...while-lus
do
{
    Console.WriteLine("Voer een getal in tussen 1 en 10:");
    getal = int.Parse(Console.ReadLine());

} while (getal < 1 || getal > 10); // Voorwaarde: getal moet tussen 1 en 10

Console.WriteLine("Je hebt een correct getal ingevoerd: " + getal);
```

# CHOOSE OR LOSE!

- Een eenvoudig menu voor een rekentool biedt
- De gebruiker kan kiezen uit verschillende rekenkundige bewerkingen
- De gebruiker moet 2 getallen invoeren
- Na de bewerking moet het programma vragen of er nog een bewerking uitgevoerd moet worden.

## do...while of while?

Kies een bewerking:

1. Optellen
2. Aftrekken
3. Vermenigvuldigen
4. Delen
5. Afsluiten

> 1

Voer het eerste getal in: 10

Voer het tweede getal in: 5

Resultaat: 10 + 5 = 15

Wil je nog een bewerking uitvoeren? (ja/nee)

> ja

Kies een bewerking:

1. Optellen
2. Aftrekken
3. Vermenigvuldigen
4. Delen
5. Afsluiten



# ***LESDOELEN BEHAALD?***

- Begrijp je het concept 'while-lus' en kun je deze toepassen in C#
- Ken je een aantal C# functionaliteiten (Math, Random) en kun je deze toepassen

# ***ASYNCHROON***

- Algoritmen - Herhalingen & Branching
- Herhalingsstructuren - Nesting