Hand-out

Programmeren

06. Loops

1. Inleiding

In dit hoofdstuk gaan we werken met loops. Loops of ook wel iteraties genoemd zijn herhalingen van bepaalde delen van je code. Als we bijvoorbeeld 100 keer een regel op het scherm willen tonen is het niet makkelijk als we de regel 100 keer in de code moeten zetten, zo zou je onnodig grote programma's krijgen en onleesbare code, hier komen herhalingen heel goed van pas.

2. Wat is een Loop

Een loop of een iteratie is wanneer we een computer opdracht herhalen. Deze herhaling kan op een aantal manieren tot stand komen. We kunnen kiezen voor een oneindige herhaling, dat is vaak niet handig omdat we dan het programma abrupt moeten af breken en de processor nog steeds herhalingen aan het uitvoeren is "onder water" tot dat de GC (Garbage Collector) dit opruimt. Meer over de Garbage Collector in een later hoofdstuk.

Wel kunnen we kiezen voor een herhaling die we een statisch aantal keer willen uitvoeren, hiervoor gebruiken we vaak een for of foreach loop. Of een herhaling die uitgevoerd word tot een bepaalde stelling waar of niet waar is en hier voor gebruiken we vaak een while of do-while loop.

In C# kennen we dus de volgende loop types:

- For : wanneer we een statisch aantal keer willen herhalen

Foreach : wanneer we door een List of Array willen lopen
 While : wanneer we willen herhalen zolang iets waar is
 Do While : wanneer we willen herhalen zolang iets waar is.

3. Herhaling met behulp van een variabele

Wanneer je een herhaling wil gaan uitvoeren is het handig om variabelen te gebruiker. Dit kan bijvoorbeeld een variabele zijn die het aantal keer herhalen aangeeft of je kunt een variabele gebruiken die bij iedere herhaling een optelt.

Bij een for loop heb je vaak het geval dat je beide hebt, een variabele die begint bij 0 en iedere iteratie een meer wordt, en dit doet zolang deze niet gelijk is aan een variabele die een statische waarde heeft. Voor deze variabele gebruiken we meestal het type int. Onthoudt goed, int is om aan te geven hoe vaak we iets willen doen, en double gebruiken we om aan te geven hoeveel we ergens van hebben.

4. Demo

In dit stuk gaan we klassikaal een demo maken. Het is de bedoeling dat je mee doet en zo stapsgewijs houvast krijgt op het onderwerp, maak daar waar nodig is aantekeningen in je schrift of in je kladblok. Als je klaar bent ga je zelf aan de slag dus let goed op!

5. Opdracht

Maak een programma dat vraagt om een getal onder de 100 en daar van de tafel op het scherm zet. Zorg deze keer ook dat het programma zelf in een loop staat zodat het programma je iedere keer vraagt of je nogmaals een tafel op het scherm wil tonen. Deze vraag moet beantwoord worden met j of n (j/n) en wanneer de gebruiker nee antwoord (n) moet het programma stoppen.

Twee extra eisen voor dit programma zijn:

- De loop waarin het programma draait moet een do-while loop zijn.
- De loop die de tafels op het scherm zet moet een for loop zijn.

Extra punten kunnen worden verdiend met de volgende optie :

- Het programma moet controleren of het ingevoerde getal onder de 100 zit en als dit NIET het geval is moet het de vraag stellen die hier boven benoemd is.

Als je klaar bent met deze opdracht ga je je werk samen met de docent bekijken. veel succes!

De code kun je vinden op GitHub (rep 06.-Loops)

6. Uitwerking

Demo Code:

```
1. class Program
2. {
3.
       static void Main(string[] args)
4.
5.
6.
           string strCharField = String.Empty;
7.
           int blnToggle = 0;
8.
9.
           do
10.
               char chrRandomCharacter = GetLetter();
11.
12.
               Console.ForegroundColor = ConsoleColor.Green;
13.
               Console.Write(chrRandomCharacter);
14.
               Thread.Sleep(3);
15.
16.
           while (true);
17.
18.
19.
       public static char GetLetter()
20.
21.
           string chars = "$%#@!*abcdefghijklmnopqrstuvwxyz1
                            234567890?;:ABCDEFGHIJKLMNOPQRSTUVWXYZ^&";
22.
23.
           Random rand = new Random();
24.
           int num = rand.Next(0, chars.Length - 1);
25.
           return chars[num];
26.
27.}
```