

leren. durven. doen.



Leren Programmeren

ALGORITMISCH DENKEN

Inhoud

Basisbegrippen in de C# programmatie

- Keywords(gereserveerde woorden)
- Variabelen en constanten
- datatypes (numeriek, strings, tekst, datum ...)
- operatoren

leren. durven. doen.



Basisbegrippen

LEREN PROGRAMMEREN

Gereserveerde woorden in C# (keywords)

In de C# programmeertaal hebben bepaalde woorden een gereserveerd gebruik. Deze worden keyword genoemd. Deze keywords worden rechtstreeks door de compiler verstaan en gecompileerd. Deze worden kan je dus niet gebruiken als namen voor variabelen of constanten. Hieronder zie je een lijst van keywords in C#:

| | | | | | | |
|--------------------------|--------------------------|-------------------------|---------------------------|---------------------------|------------------------|-------------------------|
| abstract | const | extern | int | out | short | typeof |
| as | continue | false | interface | override | sizeof | uint |
| base | decimal | finally | internal | params | stackalloc | ulong |
| bool | default | fixed | is | private | static | unchecked |
| break | delegate | float | lock | protected | string | unsafe |
| byte | do | for | long | public | struct | ushort |
| case | double | foreach | namespace | readonly | switch | using |
| catch | else | goto | new | ref | this | virtual |
| char | enum | if | null | return | throw | void |
| checked | event | implicit | object | sbyte | true | volatile |
| class | explicit | in | operator | sealed | try | while |
| get | partial | set | value | where | yield | |

Variabelen versus literals

LEREN PROGRAMMEREN
BASISBEGRIPPEN

Variabelen en literals

Gebruik van gegevens

- Letterlijke gegevens (literals)
- Gegevens met een bepaalde naam (variabelen)

Variabelen dienen als placeholders voor het tijdelijk bijhouden van gegevens.

Variabelen hebben een:

- naam (identifier)
- data type (bepaald welke soort gegevens)

Variabelen – declaratie, initialisatie

(<https://www.learncs.org>)

Variabelen bevatten informatie

Onderscheid tussen soort informatie: wordt aangeduid door datatype voorbeelden

Tekst: datatype string

geheel getal: datatype integer

Juist(true)/fout(false): datatype boolean

- Variable wordt gedeclareerd, waarbij het datatype wordt aangeduid
- C# is een sterk getypeerde programmeertaal, in tegenstelling tot bv Javascript)
- Enkele voorbeelden van declaraties (met initialisaties) van variabelen van datatype int, string en bool

```
int myNumber = 3; // an integer
string myString = "Hello, World!"; // a string
bool myBoolean = true; // a boolean
```

Variabelen

Declareren van een variablen

Type name;

Initiële waarde toekennen aan variabelen

Type name = initieleWaarde;

Waarde kan een literal of variabelen zijn

`int getal1= 10;`

`int getal2= getal1;`

Variabele naam moet starten met een letter of een underscore en mag enkel letters, getallen of underscores bevatten. Een variabele moet ook uniek zijn binnen de scope waarin deze gedefinieerd staat

C# syntax

Puntkomma's (semicolons) versus komma's

```
int a, b, c;    // Declareer 3 variabelen
```

```
a = 5;    // Ken de waarde 5 toe aan variabele a
```

```
b = 6;    // Ken de waarde 6 toe aan variabele b
```

```
a = a + 5; // Verhoog de waarde in a met 5
```

```
c = a + b; // Ken de waarde van de som van a en b  
toe aan variabele c
```

C# variabelen: Voorbeelden

Declaratie van variabelen:

```
int a, b;  
int x;  
int y;
```

Declaratie van variabelen en initialisatie:

```
int x = 10;  
int y = 20;
```

Toekenning van waarden:

```
x = 5; y = 6;
```

Berekeningen maken:

```
z = x + y;
```

Vaste waarden worden 'literals' genoemd

Vb Numerieke literals:

```
10.50  
1001
```

Vb Tekst-literal: "John Doe"

```
String naam = "John Doe";
```

Vb char-literal: 'a'

```
char letter = 'a';
```

Voorbeelden afdrukken van variable naar Console

Gebruik <https://www.learnncs.org> of <https://dotnetfiddle.net/>

- Tekst afdrukken naar console venster:
- **Console.WriteLine()** - > na afdrukken van de tekst wordt er naar een nieuwe lijn gesprongen
- **Console.Write()** -> na afdrukken van tekst is geen nieuwe lijn

Vb zonder variabele

```
Console.WriteLine("Welkom!");
```

Vb met variabele

```
string welkomTekst;  
welkomTekst = "Welkom!";  
Console.WriteLine(welkomTekst);
```

Oefening:

Druk de tekst « Hello World! » af in het console venster gebruik eens een variable en ook eens zonder variabele

Oefening

Druk de volgende tekst af (sterretjes) in het console venster

```
* * * * *  
*           *  
*           *  
*           *  
* * * * *
```

Oplossing 1

Zonder variabele

```
Console.WriteLine("*****");  
Console.WriteLine("*          *");  
Console.WriteLine("*          *");  
Console.WriteLine("*          *");  
Console.WriteLine("*          *");  
Console.WriteLine("*          *");  
Console.WriteLine("*****");
```

Oplossing 2

Met 2 variabelen

```
string lijnBuiten = "*****";  
string lijnBinnen = "*      *";  
  
Console.WriteLine(lijnBuiten);  
Console.WriteLine(lijnBinnen);  
Console.WriteLine(lijnBinnen);  
Console.WriteLine(lijnBinnen);  
Console.WriteLine(lijnBinnen);  
Console.WriteLine(lijnBinnen);  
Console.WriteLine(lijnBuiten);
```

vanaf C# 7.1 –Initialisatie met default

Vanaf C# 7.1 kunnen we met het keyword **default** een variabele (her)initialiseren met zijn default waarde

Bv:

```
bool isDirty = true;
Console.WriteLine("isDirty=" + isDirty);
isDirty = default;
Console.WriteLine("isDirty default = " + isDirty);
byte Shirts = 4;
string title = "Schoonmaakdienst";
Shirts = default; // 0;
title = default; // lege string
DateTime today = DateTime.Now;
Console.WriteLine("today = " + today);
today = default;
Console.WriteLine("today default=" + today);
```

Test deze code in <https://dotnetfiddle.net/> om de default keyword te kunnen gebruiken, moet je als compiler .NET Core 3.1 (Of Roslyn 3.4) kiezen

Variabelen declaration/initialization

```
static void Main()
{
    Console.WriteLine("=> Data Declarations:");
    // Local variables are declared and initialized as follows:
    // dataType varName = initialValue;
    int myInt = 0;
    string myString;
    myString = "This is my character data";
    // Declare 3 bools on a single line.
    bool b1 = true, b2 = false, b3 = b1;
    // Use System data type to declare a bool.
    System.Boolean b4 = false;
}
```


leren. durven. doen.



Constanten

**LEREN PROGRAMMEREN
BASISBEGRIPPEN**

Constance

Een constante heeft dezelfde eigenschappen als een variabele:

Enige verschil:

- Constante kan eenmaal hij is aangemaakt niet meer van waarde veranderen!
- Dus bij een constante zijn naam en waarde statisch.
- Constanten hebben vooral te maken met “proper” programmeren zoals later duidelijk wordt

Een constante wordt aangemaakt door gebruik van het sleutelwoord **const**

```
13 13 {  
14 - private const int _timeOutInSeconds = 1000;  
14 + private const int _timeOutInSeconds = int.MaxValue;
```



We do not need a 68 year timeout.

[Reply](#) · [Edit](#) · [Delete](#) · [Create task](#) · [A moment ago](#)

Constante

Voorbeelden:

const float PI = 3.1415;

const char constChar = 'b';

Belangrijk: een constante moet onmiddellijk een waarde krijgen tijdens zijn creatie (declaratie) = initialisatie.

Voorbeeld Constanten

```
const string dogName = "Kirby";  
const int numBarks = 3;  
const string foodKind = "cookies";  
const double foodVolume = 1.2;
```

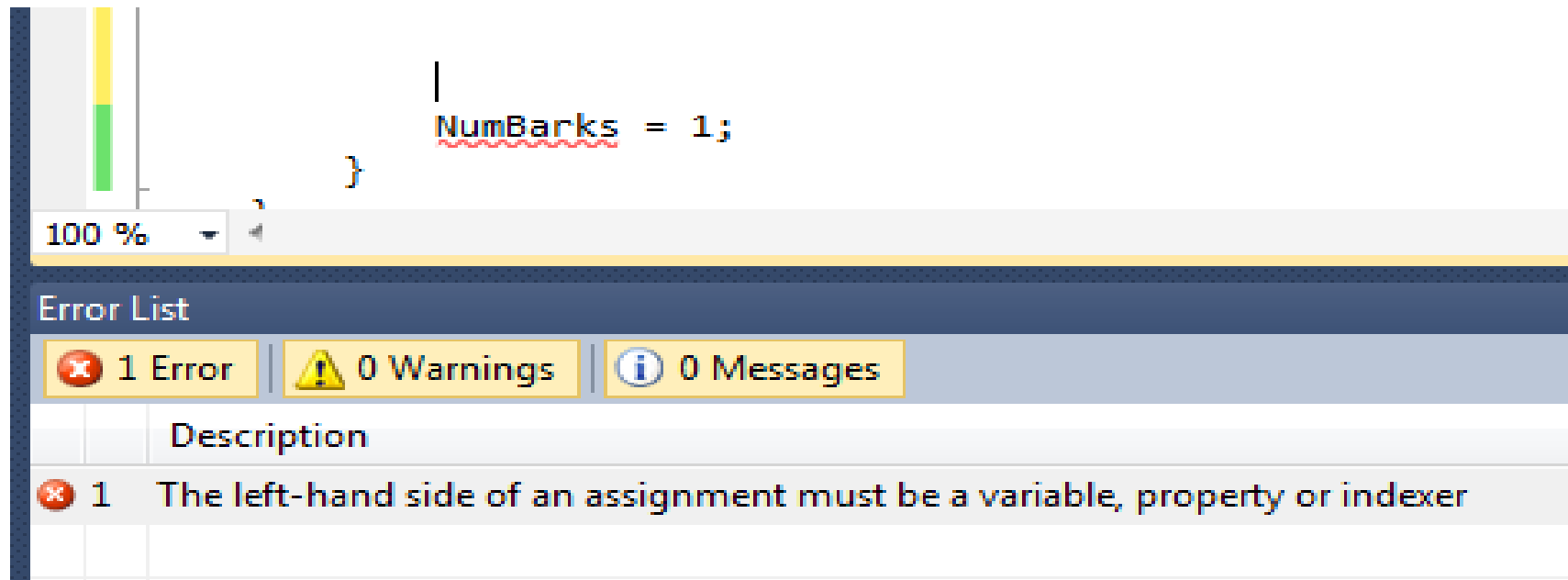
```
Console.WriteLine("When dog " + dogName + " barks " + numBarks + " times," +  
    " he gets " + foodVolume + " kg of " + foodKind);
```

Waarde van Constante kan niet veranderen

Voorbeeld:

```
const int numBarks = 3;
```

~~numBarks = 1;~~ -> compiler geeft foutmelding



Variabelen en constanten



Oefeningen

- **Oefening-variabelenEnConstanten.pdf**

Vragen?

REFERENTIES

[HTTPS://WWW.LEARNCS.ORG/](https://www.learncs.org/)