# 4. Expressies en operatoren

H2 - De basisconcepten van C#

# Expressies

- Expressie (expression) is een zin die kan geëvalueerd worden om een bepaald resultaat te geven.
- Dit resultaat gebruiken we dan naar believen
- Bestaat uit 2 delen:
  - **Operators**: +, -,*,%, etc.
  - **Operands**: meestal literals of variabelen
- Voorbeelden van expressie:

```
2 + 3 * 4
-1 + 3
(2 + 3) * 4
2 * aantalKippen
```

# Expressie regels

- Regels bij uitwerken van expressies
  - Wordt van links naar rechts uitgewerkt.
  - Vermenigvuldiging en deling voorrang, vervolgens aftrekken en optellen.
  - Haakjes om voorrang te geven aan bepaald stuk

# Voorbeeld van expressie

```
int breedte = 15;
int hoogte = 20 * breedte;
```

# Numeric operators

- In volgorde van prioriteit/volgorde van berekening (belangrijkste boven)

| Operator | Gebruik |
|---|---|
| - | 'unary minus', gebruikt om negatieve getallen weer te geven.<br>Bv -1<br><br>Unary wil zeggen: maar één operand nodig |
| * | Vermenigvuldiging |
| / | Deling |
| + | Optelling |
| - | Aftrekking<br>Merk op: zelfde teken als unary minus, maar deze heeft 2 operands nodig.<br>Maw, compiler ziet zelf welke operator bedoeld wordt a.d.h.v. operands (links) en rechts van operator |

Er zijn nog operators, dit zijn echter de meest gebruikte.

# Modulator operator

- **Modulo operator: %**
  - Geeft rest weer na deling door getal.
  - Bijvoorbeeld:
    - 10%4  (spreek uit *“10 mod 4”*) geeft als resultaat 2
    - 13%3 => resultaat: 1
    - 6%2=> resultaat: 0

# Typisch gebruik van modulo-operator

- “M’n uberlevel paladin heeft 3261 diamanten gevonden. Hij kan deze in inventory-doosjes plaatsen waarin er telkens 500 passen. Hoeveel doosjes heeft hij nodig en hoeveel plek is er nog in het laatste doosje over als hij ze allemaal vult?”

```
int aantalDiamanten = 3261;
int doosGrootte = 500;

int aantalDozen = aantalDiamanten / doosGrootte;
int aantalInLaatsteDoos = aantalDiamanten % 500;
int overInLaatsteDoos = 500 - aantalInLaatsteDoos;

Console.Write("Aantal dozen = " + aantalDozen);
Console.WriteLine("met " + aantalInLaatsteDoos + " diamanten in laatste doos ");
Console.WriteLine("en dus " + aantalInLaatsteDoos + " diamantan nog over");
```

# Hoe verhoog je waarde van een variabele?

Stel, je hebt: int getal = 5;

Hoe verhoog ik deze met 1?

- 1° Huidige waarde van de variabele getal uitlezen
- 2° 1 bij deze waarde optellen
- 3° Resultaat terug in de variabele getal plaatsen
- Dus:
  - getal= getal +1;

# Hoe verwissel je de waarde van 2 variabelen?

Stel je hebt:

- int getalA= 6;
- int getalB= 8;

Je wil de waarden omwisselen.Je zal een extra reserve variabele nodig hebben:

- 1° Waarde eerste variabele in reserve variabele bewaren
- 2° Waarde eerste variabele overschrijven met waarde van tweede variabele
- 3° Waarde in tweede variabele overschrijven met waarde in reserve variabele

Dus:

```
int temp = getalA;
getalA= getalB;
getalB= temp;
```

# Verkorte notatie van operators

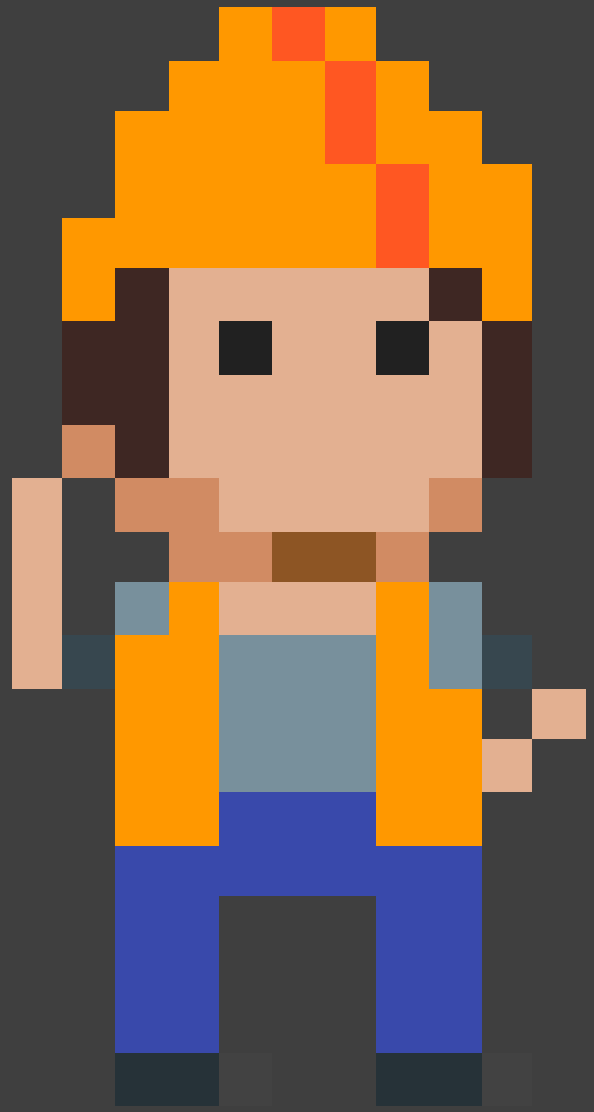
- Vooral handig wanneer je lange identifiers gebruikt

```
window_count = window_count + 1
```

```
window_count++
```

| Verkorte notatie | Volledige notatie |
|---|---|
| a++ | a = a + 1 |
| a += b | a = a + b |
| a -= b | a = a - b |
| a /= b | a = a / b |
| a *= b | a = a * b |

# Datatype van een expressie



# Datatype in expressies

- Datatypes in de expressie (van variabelen en literals) bepaald welke type als resultaat van een expressie wordt teruggegeven:
  - Integer + Integer zal integer als resultaat geven
  - Float + Float zal float als resultaat geven.
  - Etc.
- Dus:
  - 1 / 2 (integer gedeeld integer)


**Resultaat: 0 ( integer)**

# Gemengde expressies

- Indien meer dan 1 datatype in een expressie: volgorde van berekeningen en individuele datatypes bepaald finale datatype.
- Bij een (sub-)berekening met 2 verschillende datatypes kiest C# het 'grootste' (dus bv double ipv int)

- Vb:
  - `1/2.0` geeft de double 0.5 als resultaat
  - `4/3+ 3.2`
    - eerst krijgen we 4/3 => de int met waarde 1
    - Dan 1+3.2 => geeft de double 4.2

    - Resultaat van de totale expressie is dus een double (maar niet 4.5333... zoals dit uit rekenmachine zou komen)

# Const

# Magic numbers en const

- const (constant) voor variabele declaratie geeft aan dat deze variabele niet aangepast kan worden
- Identifier Meestal in all caps en underscore tussen de aparte woorden
  - Bv: `const double MAX_TEMP= 100.5;`

```
const double PI = 3.141592654;
// ...
circ = rad * 2 * PI;
```

```
const double G = 9.81; // m/s²
// ...
acceleration = mass * G;
```

# Oefeningen

6. Assume you have a variable declared as `int var1 = 3;`. What is the value of `22 % var1`?

   a. 0

   b. 1

   c. 7

   d. 21

7. Assume you have a variable declared as `int var1 = 3;`. What is the value of `22 / var1`?

   a. 1

   b. 7

   c. 7.333

   d. 21

# Demo time

- Expressies
- Waarden omwisselen
- Waarde verhogen
- Const
- Expressie resultaten