

SOFTWARE ONTWIKKELEN

AGENDA VOOR VANDAAG

- Lesdoelen
- Ternary Operator
- Lazy Evaluation
- Static Classes

LESDOELEN

Na de les van vandaag:

- Herken je de ternary operator, en kunt hem toepassen
- Begrijp je het concept Lazy Evaluation
- Ben je nauwkeuriger, kritischer en consistentter als het gaat om naamgeving.
- Weet je hoe je commentaar in zet om je code te verbeteren
- Kun je een statische klasse schrijven

VORIGE LES

Waarom vraag ik eigenlijk iedere les wat we vorige les ook alweer gedaan hadden?

TERNARY OPERATOR

Een compactere manier om een if/else te schrijven.

Reeds geleerd

```
// Bepaal het hoogste getal
int input1 = 2;
int input2 = 6;

// Classic C# ifs
int max;
if(input1 > input2)
{
    max = input1;
}
else
{
    max = input2;
}
```

Alternatief

```
// Bepaal het hoogste getal
int input1 = 2;
int input2 = 6;

// Ternary operator
int max = input1 > input2 ? input1 : input2;
```

conditie

waarde
bij true

waarde
bij false

TERNARY OPERATOR - OEFENING

Herschrijf en gebruik een ternary operator

```
// Bepaal of de student een voldoende heeft
double cijfer = 6;

string uitslag;

if(cijfer >= 5.5)
{
    uitslag = "voldoende";
}
else
{
    uitslag = "onvoldoende";
}

Console.WriteLine($"Student heeft een {uitslag}");
```

LAZY EVALUATION

```
// Voorbeeld lazy evaluation
string[] ingredienten = ["bodem", "tomatensaus", "tomaat", "ananas", "salami"];
string vorm = "rond";

bool isHetEenPizza = heeftDeJuisteVorm(vorm) && bevatVerbodenIngredienten(ingredienten);

Console.WriteLine($"Is het een pizza: {(isHetEenPizza ? "Ja" : "Nee")}");

bool heeftDeJuisteVorm(string vorm)
{
    return vorm == "rond";
}

bool bevatVerbodenIngredienten(string[] ingredienten)
{
    // Wanneer de vorm niet rond is, wordt deze functie niet uitgevoerd.
    return !ingredienten.Contains("ananas");
}
```

NAAMGEVING

Basisregels voor C#



snake_case



PascalCase ➡ methodes, klasses, properties, constanten



camelCase ➡ variabelen, argumenten

* bij private fields in klasses startend met een _



SCREAMING_SNAKE_CASE

NAAMGEVING – WAAROM BELANGRIJK?



NAAMGEVING - VOORBEELDEN

Niet duidelijk	Duidelijk
<code>bool containsNumbers = true;</code>	<code>bool numbers = true;</code>
<code>var temp = getData();</code>	<code>var userInput = getUserInput();</code>
<code>theTotalAmountOfAllTheProductsPurchased</code>	<code>totalAmount;</code>
<code>x</code>	<code>coordinateX;</code>
<code>int tmpr;</code>	<code>int temperature;</code>
<code>var customer1; var customer2;</code>	<code>// Consider using a list or array</code>
<code>var customre;</code>	<code>var customer;</code>

COMMENTAAR

Commentaar in code gebruik je om je toekomstige zelf of je collega te helpen.

- Documentatie van complexe logica
- Uitleg van het waarom (niet het wat).
- Hou het beknopt
- Vermijd meningen of persoonlijke frustraties 🖱️

```
/*  
    If you're trying to optimize this routine and it fails (most surely)  
    please increase this counter as a warning for the next person:  
  
    total_hours_wasted_here: 254  
*/
```

COMMENTAAR

```
// We schrijven iets naar de console  
Console.WriteLine("iets");
```

Voorkom dat je de code herhaalt



COMMENTAAR IN C#

Standaard

```
// Single line comment

/*
   This is a comment block
*/
```

XML documentation

```
/// <summary>
/// Bepaalt of het op dit moment vriest.
/// </summary>
/// <param name="huidigeBuitenTemperatuur">De huidige buiten temperatuur in graden celcius</param>
/// <returns>Een boolean die aangeeft of het vriest</returns>
1 reference
public static bool VriestHet(int huidigeBuitenTemperatuur)
{
    return huidigeBuitenTemperatuur <= 0;
}
```



```
bool Natuurkunde.VriestHet(int huidigeBuitenTemperatuur)
Bepaalt of het op dit moment vriest.
Returns:
    Een boolean die aangeeft of het vriest
Natuurkunde.VriestHet(0);
```

STATISCHE KLASSE

Een statische klasse is een klasse waarvan alle variabelen en methodes ook statisch zijn. Er kan geen instantie van worden gemaakt.

```
// Je hebt al statische klassen gebruikt (Math)  
var hoogsteGetal = Math.Max(1,2);
```

Ideaal voor functies die je wilt hergebruiken en volledig generiek zijn.

STATISCHE KLASSE

access
modifier

keyword

naam

```
/// <summary>
/// Bevat elementaire natuurkundige regels in de vorm van functies.
/// </summary>
0 references
public static class Natuurkunde
{
    /// <summary>
    /// Bepaalt of het op dit moment vriest.
    /// </summary>
    /// <param name="huidigeBuitenTemperatuur">De huidige buiten temperatuur in graden celcius</param>
    /// <returns>Een boolean die aangeeft of het vriest</returns>
    0 references
    public static bool VriestHet(int huidigeBuitenTemperatuur)
    {
        return huidigeBuitenTemperatuur <= 0;
    }
}
```

EINDOEFENING

Maak een rekenmachine in een statische klasse . De rekenmachine heeft de volgende functionaliteiten:

- Optellen van twee getallen
- Bepalen of een getal even of oneven is (denk aan de ternary operator)

Commentarieer je code met XML documentation en denk aan je naamgeving!

VOORBEREIDEN

- Lijsten!