



Programmeren 1 (C#)

Week 1

Wiley Finch

Programmeren jaar 1

- Programmeren jaar 1: Visual Studio 2022 / C#
- Elk semester een (praktijk)examen + verplichte opdrachten
- Periode 1.1: **Programmeren 1** (4 ECs)
 - Programmeren 1 (100%)
 - Opdrachten programmeren 1 (0%)

(imperatief en objectgeoriënteerd programmeren, Console-toepassingen)
- Periode 1.2: **Programmeren 2** (4 ECs)
 - Programmeren 2 (100%)
 - Opdrachten programmeren 2 (0%)

(object-oriented programmeren, Console applications)

Programmeren jaar 1

- Periode 1.3: **Web programming 1** (3 ECs)

- Web programming 1 (100%)

- (object-oriented programming)*

- Project 'Databases' (3 ECs)

- Periode 1.4: **Web programming 2** (3 ECs)

- Web programming 2 (100%)

- (object-oriented programming)*

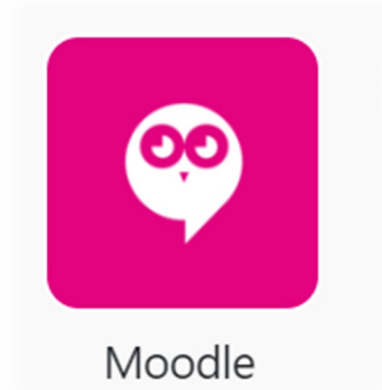
- Project 'Building Applications' (4 ECs)

Programma Periode 1.1 (Programming 1)

- 01 (wk 36) Introduction C# / Visual Studio 2022 (Community), basic problem solving
- 02 (wk 37) branching, methods
- 03 (wk 38) loops, basic version control setup
- 04 (wk 39) classes, enums, arrays
- 05 (wk 40) access control, fields/properties, values & references
- 06 (wk 41) interfaces, inheritance, version control
- 07 (wk 42) repetition / practice exam
- 08 (wk-43) *no classes*
- 09 (wk-44) exam (*practical, computer*)
- 10 (wk-45) -

Material

- Moodle course:
 - 2425 IT1.1 Web Design (*self enrollment*)
 - Powerpoint slides
 - wekelijkse opdrachten



Beoordeling

- Programmeren 1 (1924IN113A)
 - Praktijkexamen(computer)
 - Visual Studio 2022
 - Console-applicaties maken
- Opdrachten Programming 1 (1924IN113B)
 - week 2..5 (5x): Verplichte wekelijkse opdrachten, worden afgevinkt in (volgende) praktijklessen
 - week 1: Inleiding (oefeningen)

Programming 1	INF	1923IN113Z	0 van 4	
- Programming 1	INF	1918IN113A	100	Graded
- Assignments Programming 1	INF	1918IN113B	0	Sat/Unsat

Lessen

- Theorie lessen
 - theorie, probleemanalyse, oefeningen, demo's, vragen
- Praktijklessen
 - verplichte wekelijkse opdrachten
 - Microsoft Visual Studio 2022
 - Één solution elke week met alle opdrachten

Les 1

Inleiding tot programmeren

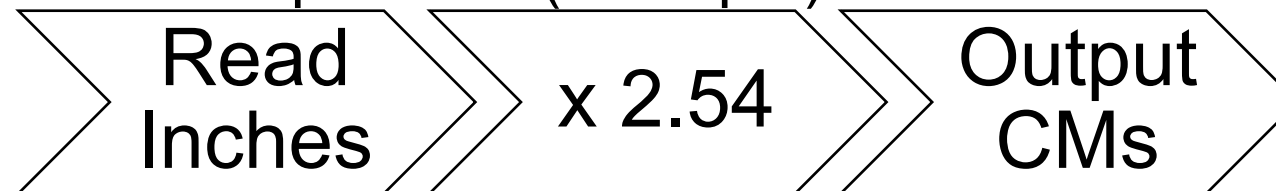
Software development

Proces van een softwareprogramma

Generic



Solution of a problem (example)



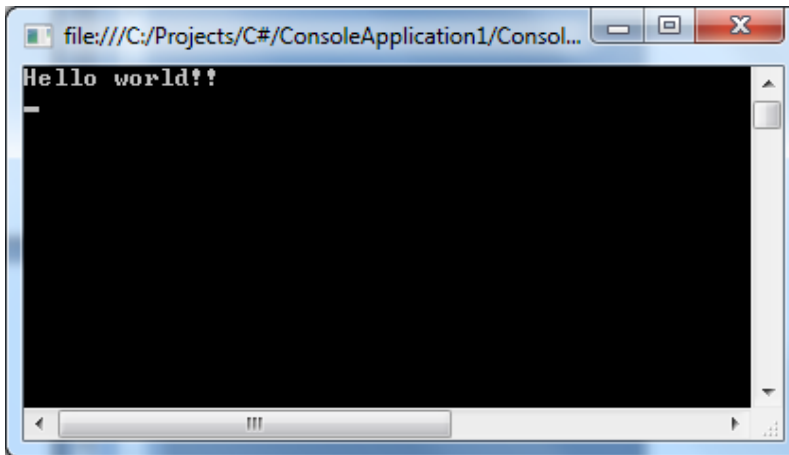
Development environment

- Voor het ontwikkelen van software: IDE
(Integrated Development Environment)
- IDE-applicaties bevatten verschillende functies/tools
(highlighting text, intellisense, debugging, version control, graphical designer, ...)
- C# → **Visual Studio 2022 (Community)**, Visual Studio Code

Soort programma's

- Periode 1 en 2
 - Console applicaties
- Periode 3 en 4
 - Web applicaties

Console applicaties



Console application

We beginnen met het programmeren van
Console applicaties (Periode 1.1 – 1.2)

Visual Studio 2022 (Community) - demo



(hoe maak je een Console-applicatie)

Variabelen

- Een variabele is een benoemde locatie waar je iets kunt opslaan (zoals een geheel getal, tekst,...)
- Naam van een variabele:
 - begint met een letter of _
 - gevolgd door een letter, cijfer of _
 - Er kunnen geen gereserveerde trefwoordnamen worden gebruikt (if, for, while, ...)
 - Conventie: begin met een kleine letter

```
int number, _answer, b52, 1ofakind, false;
```

Data types

- Een variabele heeft een bepaald data type
- Mogelijke waarden voor een variabele zijn afhankelijk van het data type
- Primitieve data types (build in):
 - byte / short / **int** / long
 - **float** / **double** / decimal
 - **char** / **string**
 - **bool** (true / false)

```
int number, _answer;  
string name;  
char choice;  
bool found;  
float length;
```

Visual Studio 2022 (Community) - demo



(hoe variabelen te declareren)

int, double, boolean

De belangrijkste (meest gebruikte) datatypes zijn:

- int

Gehele getallen:

0, -25, 123,000,000

- double

Drijvendekomma waarden:

3.14, -25.555555

- boolean

true or false

Operators for int

+	add	3 + 4	(7)
-	subtract	5 - 2	(3)
*	multiply	4 * 5	(20)
/	whole division (<i>quotient</i>)	10 / 3	(3)
%	modulo (<i>remainder</i>)	10 % 3	(1)

```
int height1, height2, totalHeight;

height1 = 75;
height2 = 200;
totalHeight = height2 + height1;    // 275
totalHeight = height2 / height1;    // 2
totalHeight = height2 % height1;    // 50
```

Operators for double

+	add	$3.1 + 4.0$
-	subtract	$5 - 2.1$
*	multiply	$4.0 * 5.1$
/	<i>normal</i> division	$10 / 3$

```
double p, q, r, s, t;  
p = 3.1 + 4.0; // 7.1  
q = 5 - 2.1;   // 2.9  
r = 4.0 * 5.1; // 20.4  
s = 10 / 3;    // 3 (!)  
t = 10.0 / 3;  // 3.33333333
```

Data type bool / boolean

- Het primitieve data type boolean kan slechts 2 verschillende waarden bevatten: true or false
- Normaal gesproken krijg je een boolean (waarde) als gevolg van het vergelijken van 2 waarden
- Vergelijken is mogelijk met relationele operatoren

Relationele operators

- > groter dan
- < kleiner dan
- >= groter dan of gelijk aan
- <= kleiner dan of gelijk aan
- == gelijk aan
- != niet gelijk aan

Een boolean voorbeeld (1)

```
int maxStudents = 30;  
int nrOfStudenten = 24;  
bool result;  
  
result = (nrOfStudenten < maxStudents);  
result = (nrOfStudenten <= maxStudents);  
result = (nrOfStudenten != maxStudents);
```

```
int maxStudents = 30;  
int nrOfStudenten = 24;  
bool result;  
  
result = (nrOfStudenten < maxStudents);      // true  
result = (nrOfStudenten <= maxStudents);      // true  
result = (nrOfStudenten != maxStudents);      // true
```

Een boolean voorbeeld (2)

```
int maxStudents = 30;  
int nrOfStudenten = 24;  
bool result;  
  
result = (nrOfStudenten == maxStudents);  
result = (nrOfStudenten >= maxStudents);  
result = (nrOfStudenten > maxStudents);
```

```
int maxStudents = 30;  
int nrOfStudenten = 24;  
bool result;  
  
result = (nrOfStudenten == maxStudents);    // false  
result = (nrOfStudenten >= maxStudents);    // false  
result = (nrOfStudenten > maxStudents);     // false
```

Constanten

- Een constante heeft een waarde die niet kan veranderen
- Een constante wordt op dezelfde manier verklaard als een variabele, voorafgegaan door het woord 'const'
 - aan een constante moet onmiddellijk een waarde worden toegekend
 - conventie: eerste letter en elk volgend woord begint met een hoofdletter (PascalCase)

```
const int MinAge = 15;  
const int MaxAge = 60;  
const decimal VatRate = 0.21M;  
const float Length;
```


Visual Studio 2022 (Community) - demo



(hoe constanten te definiëren)

Operator voorrang

– Prioriteit hoog naar laag:

→ + (unair) en - (unair)

→ * / %

→ + (binair) en - (binair)

– Gelijke prioriteit (evalueren) van links naar rechts

– Met haakjes verander je de prioriteit

```
int x1, x2, x3, x4, x5;
```

```
x1 = 3 + 4 * 2;      // 11
```

```
x2 = (3 + 4) * 2;    // 14
```

```
x3 = 9 / 4;          // 2
```

```
x4 = 9 % 4;          // 1
```

```
x5 = 40 / 5 * 2;     // 16
```

Some extra mathematical operators


- `++` unary (1) verhoging met 1
- `--` afnemen met 1
- `+=` binair (2) optellen en toewijzen
- `-=` aftrekken en toewijzen
- `*=` vermenigvuldigen en toewijzen
- `/=` verdelen en toewijzen
- `%=` modulo/rest en toewijzen

```
int x1 = 1, x2 = 2, x3 = 3;  
x1++;           // 2  
x2--;           // 1  
x3 += 4;        // 7
```

char en string

- char kan 'één' teken bevatten
- string kan "meerdere" tekens bevatten

```
char letter = 'g';  
char letter2 = "x";  
string word = "multiple characters";  
string word2 = 'blablabla';
```

 `struct System.Char`

Represents a character as a UTF-16 code unit.

Too many characters in character literal

Visual Studio 2022 (Community) - demo



(hoe operatoren toe te passen op int/dubbele variabelen)

Inleiding – Analyse / Ontwerp

Programmeren: software/programma schrijven om een probleem op te lossen

Programma: set instructies, verwerkt door een computer

- Welke gegevens hebben we nodig?
- Wat moeten we met deze gegevens?
- Welke instructies?
- Instructies in welke volgorde? - ...

Inleiding – Analyse / Ontwerp

Ontwerp: beschrijf instructies/algorithmen in pseudocode

Pseudocode: informele beschrijving van een computerprogramma

"Een algoritme is een eenduidige specificatie (set instructies) van hoe een klasse van problemen moet worden opgelost".
(Wiki, z.d.)

Fasen van softwareontwikkeling

1) Analyse

Welke gegevens zijn nodig?

Waar kan ik deze gegevens vandaan halen?

Wat moet er met deze gegevens gebeuren?

2) Ontwerp

Algoritme beschrijven met pseudocode
(*test: controleren op logische fouten*)

3) Implementatie

Pseudocode omzetten in een programma (instructies)
(*test: controleer op syntaxisfouten en logische fouten*)

An example - analysis

(1/4)

- *Calculate and display the cost of fuel, tanked at a gas station*
- Analysis
 - What data is needed?
 - price of one liter of fuel
 - number of liters tanked
 - What has to be done with this data?
 - multiply number of liters with the price of one liter

An example - analysis

(2/4)

– *Calculate and display the cost of fuel, tanked at a gas station*

(define a set of instructions)

1. read price of one liter fuel
2. read number of liters tanked
3. calculate the amount by multiplying the number of liters with the price of one liter
4. display the amount

An example - analysis

(3/4)

– *Calculate and display the cost of fuel, tanked at a gas station*

(now use variables and operators)

1. read pricePerLiter
2. read numberOfLiters
3. $\text{amount} = \text{numberOfLiters} * \text{pricePerLiter}$
4. display amount

*Always give variables
meaningful names!!*

An example - analysis

(4/4)

– *Calculate and display the cost of fuel, tanked at a gas station*

(use a constant, if possible)

1. PricePerLiter = 1.79 ← constant
2. read numberOfLiters
3. amount = numberOfLiters * PricePerLiter
4. display amount

An example – design (pseudocode)

- One of the 3 basic structures in a (imperative) programming language is ‘sequence’ (*instructions in sequential order*)

```
PricePerLiter = 1.79  
read numberOfLiters  
amount = numberOfLiters * PricePerLiter  
display amount
```

An example – Console App

(1/6)

- Create a new project via (menu) File | New | Project...
select Console App

Project name

ConsoleApp-example

Location

C:\Users\gerwin.vandijken\source\repos

Solution name ⓘ

Programming1-week2

☐ Place solution and project in the same directory

1. Give the project a name

2. Give the solution a name

An example – Console App

- Make sure you have .NET 8.0 selected as Framework
- Make sure to check “Do not use top-level statements”

Additional information

Console App C# Linux macOS Windows Console

Framework ⓘ

.NET 8.0 (Long Term Support) ▼

☒ Do not use top-level statements ⓘ

☐ Enable native AOT publish ⓘ

An example – Console App

(2/6)

– *Calculate and display the amount of tanked fuel*

```
class Program
{
    static void Main(string[] args)
    {
        // 1. read number of litres tanked
        // ...


        // 2. calculate the amount
        // ...

        // 3. display the amount
        // ...
    }
}
```


An example – Console App

(3/6)

– Step 1: input / read



```
class Program
{
    static void Main(string[] args)
    {
        // 1. read number of litres tanked
        Console.WriteLine("Enter number of litres tanked: ");
        string input = Console.ReadLine();


        // 2. calculate the amount
        // ...

        // 3. display the amount
        // ...
    }
}
```

An example – Console App

(4/6)

- Data type conversion: from **string** to **double**



```
static void Main(string[] args)
{
    // 1. read number of litres tanked
    Console.Write("Enter number of litres tanked: ");
    string input = Console.ReadLine();

    // convert string to double (data type conversion)
    double nrOfLitres = double.Parse(input);
    // or: double nrOfLitres = Convert.ToDouble(input);


    // 2. calculate the amount
    // ...

    // 3. display the amount
    // ...
}
```

An example – Console App

(5/6)

– Step 2: calculate amount



```
class Program
{
    const double PRICELITRE = 1.79;

    static void Main(string[] args)
    {
        // 1. read number of litres tanked
        Console.Write("Enter number of litres tanked: ");
        string input = Console.ReadLine();

        // convert string to double (data type conversion)
        double nrOfLitres = double.Parse(input);

        // 2. calculate the amount
        double amount = nrOfLitres * PRICELITRE;

        // 3. display the amount
        // ...
    }
}
```

An example – Console App

(6/6)

– Step 3: display

Add (useful)
comments to your
code.


```
class Program
{
    const double PRICELITRE = 1.79;

    static void Main(string[] args)
    {
        // 1. read number of litres tanked
        Console.Write("Enter number of litres tanked: ");
        string input = Console.ReadLine();

        // convert string to double (data type conversion)
        double nrOfLitres = double.Parse(input);

        // 2. calculate the amount
        double amount = nrOfLitres * PRICELITRE;

        // 3. display the amount
        Console.WriteLine("The amount is: " + amount);
    }
}
```



An example - result



A screenshot of a Windows command prompt window. The title bar shows the file path: file:///C:/Users/Gerwin van Dijken/Documents/l... The window contains the following text: "Enter number of litres tanked: 35.5" followed by "The amount is: 63.545". A cursor is visible on the line below the output.

```
file:///C:/Users/Gerwin van Dijken/Documents/l...
Enter number of litres tanked: 35.5
The amount is: 63.545
```

– Wait for user input:

```
static void Main(string[] args)
{
    // ...

    // wait for user to enter a (random) key
    Console.ReadKey();
}
```

Declaration / initialization var/const

```
class Program
{
    // declaration and initialization of a constant
    const double PRICELITRE = 1.79;

    static void Main(string[] args)
    {
        // declaration of variables
        string input;
        double nrOfLitres;

        // 1. read number of litres tanked
        Console.Write("Enter number of litres tanked: ");
        input = Console.ReadLine();

        // convert string to double (data type conversion)
        nrOfLitres = double.Parse(input);

        // ...
    }
}
```

Exercise ‘Value Added Tax’ (VAT)

- Read a price. Determine the VAT (21%) for this price. Print the price, VAT value, and the total price (price + VAT).

Exercise 'Value Added Tax' - pseudocode

```
VatRate = 0.21
```

```
read price
```

```
vat = price * VatRate
```

```
totalPrice = price + vat
```

```
display price, vat, totalPrice
```


Exercise 'leakage'

- *A tap is leaking a certain amount of water (cubic centimeter per hour). How many liters of water is lost in one week?*
- Analyse the problem and write pseudocode for the solution.

Exercise 'leakage' - pseudocode

```
HoursInWeek = 168
```

```
CcInLiter = 1000
```

```
read ccPerHour
```

```
ccPerWeek = ccPerHour * HoursInWeek
```

```
litersPerWeek = ccPerWeek / CcInLiter
```

```
display litersPerWeek
```

Homework

- (practical class) Programming 1
 - week 1 exercises (Moodle)

