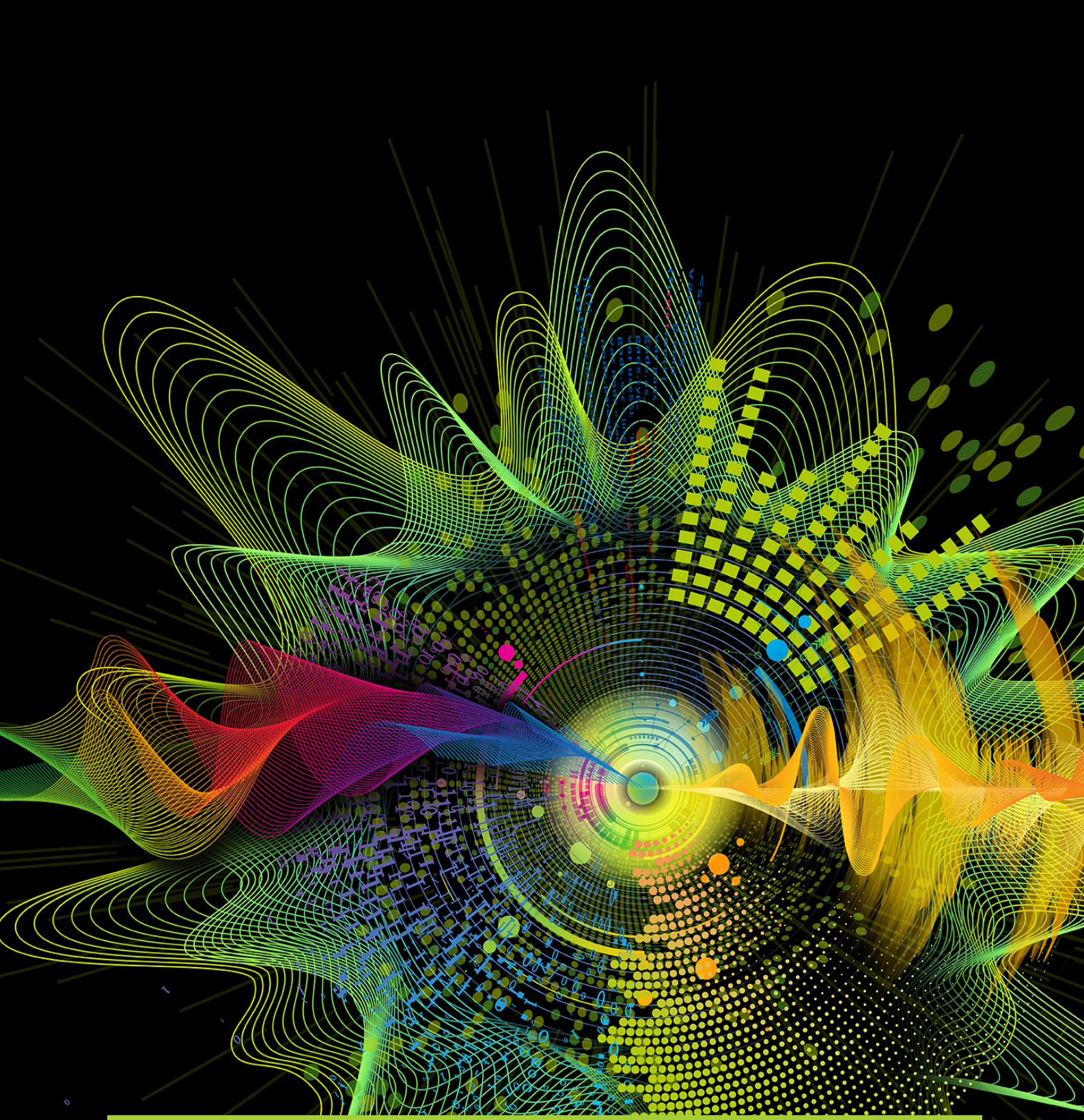


Programmeren

Module 7: C# 1

Leerjaar:	1
Crebonummer:	25604
Sector:	Software developer

INTRODUCTIE

Inmiddels heb je al behoorlijk leren programmeren in de taal JavaScript. Naast de taal JavaScript gaan we nu een tweede taal aanleren, namelijk C# (spreek uit: C-sharp). Dit is een programmeertaal die ontwikkeld is door Microsoft. Er zijn vrij veel overeenkomsten met de taal JAVA (dit is anders dan JavaScript). Destijds probeerde Microsoft de ervaren JAVA-programmeur te verleiden om met de Microsofttechnologie te gaan werken door een taal te ontwikkelen die heel herkenbaar was voor de JAVA-programmeur. Uiteindelijk zijn beide talen verder door ontwikkeld, maar er zijn nog steeds vrij veel overeenkomsten.

Wat voor JAVA geldt, geldt ook voor JavaScript. Je zal tijdens het werken aan deze module ook overeenkomsten zien tussen C# en JavaScript. Er is zelfs een hoofdstuk waarin we gegeven code in JavaScript gaan omzetten in C#.

De taal C# kan je voor heel veel doeleinden gebruiken. Je kan er bijvoorbeeld webapplicaties, desktopapplicaties, apps voor de telefoon en games mee maken. In deze module gaan we starten met de meest eenvoudige soort applicatie, namelijk de console-applicatie. Dit is een applicatie zonder een grafische user interface. Alle invoer en uitvoer is alleen tekstueel (zoals de computers uit het verre verleden)

LEERDOELEN

- Ik pas in C# de juiste syntax toe
- Ik declareer en initialiseer variabelen in C#
- Ik voer in C# bewerkingen op strings uit
- Ik maak berekeningen met variabelen in C#
- Ik gebruik beslisstructuren in C#
- Ik lees inputvelden uit in C#
- Ik maak gebruik van herhalingslussen ('for' en 'while') in C#
- Ik converteer variabelen naar een ander type (indien nodig) in C#
- Ik zet bestaande Javascriptcode om naar C#
- Ik debug C# code met behulp van breaktekens

TIJDSPLANNING

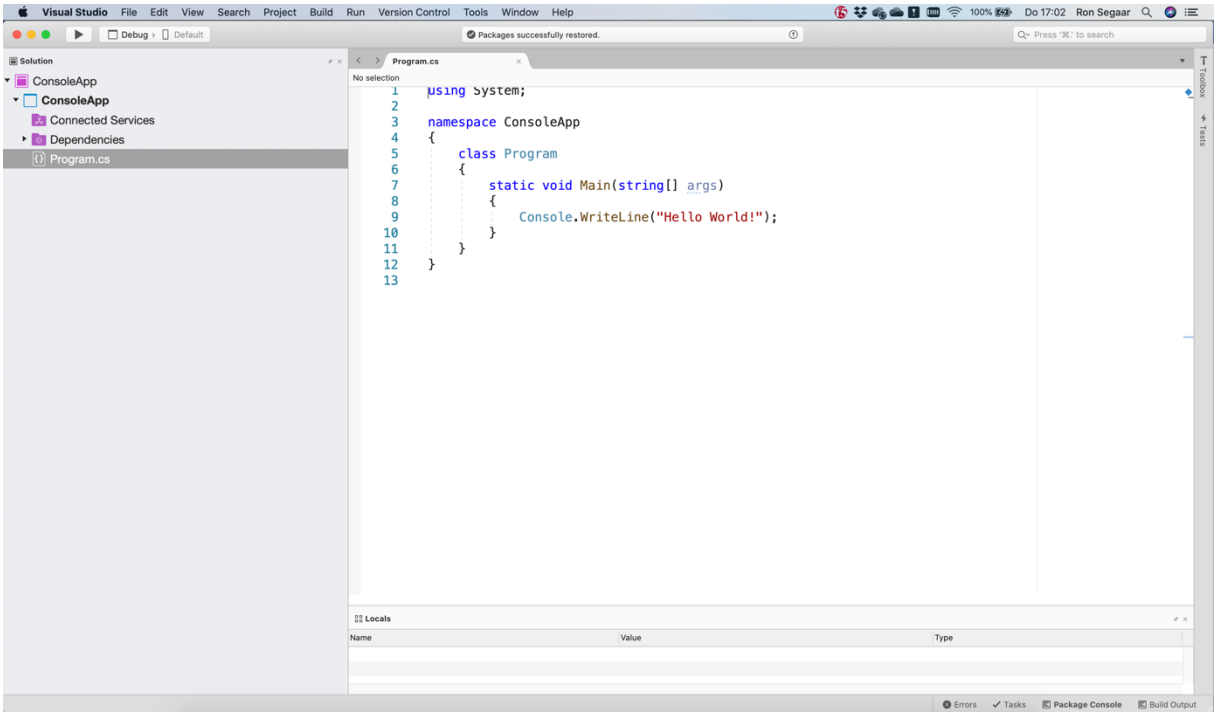
Week 1		
BASIS1, les 1	2 uur	1 De console-applicatie 2 Variabelen (declareren, Initialiseren, gebruiken) 3 Conversie en Input 4 Opdracht 1-1
BASIS1, les 2	2 uur	1 Beslisstructuren 2 Opdracht 2-1 3 Opdracht 2-2
Week 2		
BASIS1, les 3	2 uur	1 Herhalingslussen 2 Opdracht 3-1 3 Opdracht 3-2
BASIS1, les 4	2 uur	1 Debuggen 2 Opdracht 4-1 3 Opdracht 4-2 4 JavaScriptcode omzetten naar C# 5 Opdracht 4-3 6 Opdracht 4-4
Week 3		
BASIS1, les 5	2 uur	1 Bewerkingen op stringvariabelen 2 Opdracht 5-1 3 Opdracht 5-2
BASIS1, les 6	2 uur	1 Eindopdracht

LES 1 - 2 UUR – 1 DE CONSOLE-APPLICATIE

INLEIDING

Om goed te leren programmeren in C# kan je het beste starten met de meest eenvoudige type applicatie. Dit is de console-applicatie. Hierbij is er geen grafische user interface, maar gebeurt alle invoer en uitvoer met tekst.

Op LinkedIn Learning of YouTube staan een aantal filmpjes voor het aanleren van C# met behulp van de console-applicatie (zie externe bronnen). Hier zal er een aantal keer naar verwezen worden.



In deze module zullen we gebruik maken van Visual Studio. Dit is een softwarepakket van Microsoft waarmee je (bijna) alle soorten applicaties in C# kan maken. Er is een versie beschikbaar voor zowel de Windows machine (Visual Studio Community 2019) als de Mac (Visual Studio for Mac).
LET OP: Deze versies zijn **NIET** dezelfde als Visual Studio Code.

WERKVORM

Bij een aantal oefeningen en opdrachten wordt er aanvullend een extra werkvorm aangeboden. Deze herken je aan de icoontjes aan het begin van de tekst.

Als je het lastig vindt om de opdracht direct te begrijpen bij het lezen ervan, kan je nog een extra instructiefilmpje bekijken.

Vind je het lastig om een begin te maken, dan kan je een project downloaden waarin al een start gemaakt is en waar in commentaarregels de vervolgstappen staan. Een voorbeeld:

```
static void Main(string[] args)
{
    //controle: Wat krijg je als je een int en een float bij elkaar optelt?

    int geheelGetal = 5;                        //maak een int variabele
    float gebrokengetal = 7.8f;                 //maak een float variabele

    //als je de optelling in een int variabele wil zetten krijg je een error
    //(de rode kringellijn)
    int antwoord = geheelGetal + gebrokengetal;

    //als je de optelling in een float variabele wil zetten gaat het goed:
    float antwoord2 = geheelGetal + gebrokengetal;

    //probeer op dezelfde manier het uitkomst van de volgende optellingen te bepalen:
    //int + double
    //float + double
    //string + int
    //float + string
    //double + string
    //bool + string
```

EXTERNE BRONNEN

Om Visual Studio te downloaden: https://visualstudio.microsoft.com/downloads/

In de volgende drie videofragmenten op LinkedIn Learning krijg je een indruk van de taal C# en Visual studio.

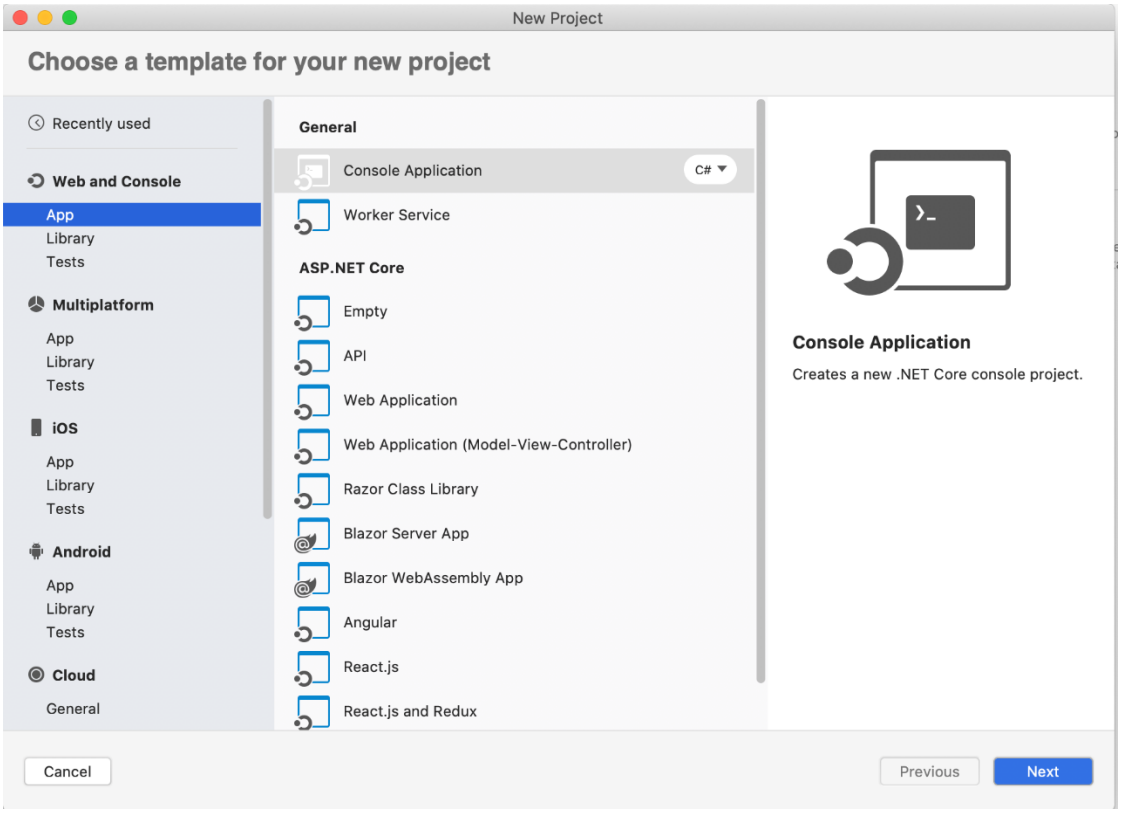
Intro: https://www.linkedin.com/learning/learning-c-sharp-3/welcome?u=36995692
Installatie: https://www.linkedin.com/learning/learning-c-sharp-3/install-the-environment?u=36995692
Eerste consoleproject: https://www.linkedin.com/learning/learning-c-sharp-3/a-hello-world-application?u=36995692

THEORIE

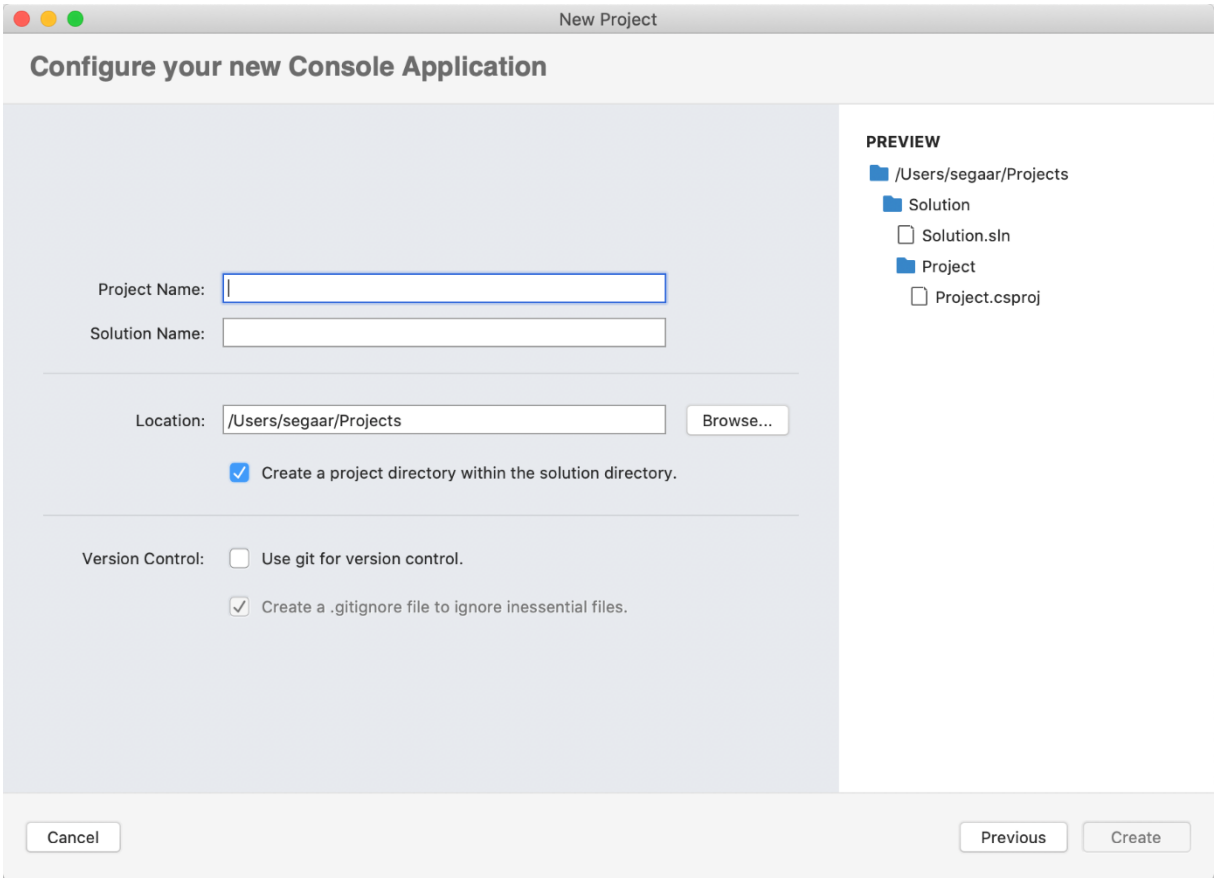
Bekijk eerst de drie videofragmenten die hierboven aangegeven staan. Deze geven een eerste indruk van een C#-console-applicatie en Visual Studio.

In de derde video wordt gesproken over het .net (spreek uit: dotnet) en .net core (spreek uit: dotnet core) framework. Dit zijn de oude en nieuwe frameworks waarop C# werkt. C# heeft niet alleen een besturingssysteem nodig zoals Windows of MacOS, maar daarbij moet ook dit framework geïnstalleerd zijn (door Visual Studio te installeren weet je zeker dat ook het juiste framework geïnstalleerd is). De oude versie van dit framework .net is alleen beschikbaar voor het Windows platform. De nieuwe versie .net core , die wij gaan gebruiken, is open source en werkt op Windows, Linux en Mac.

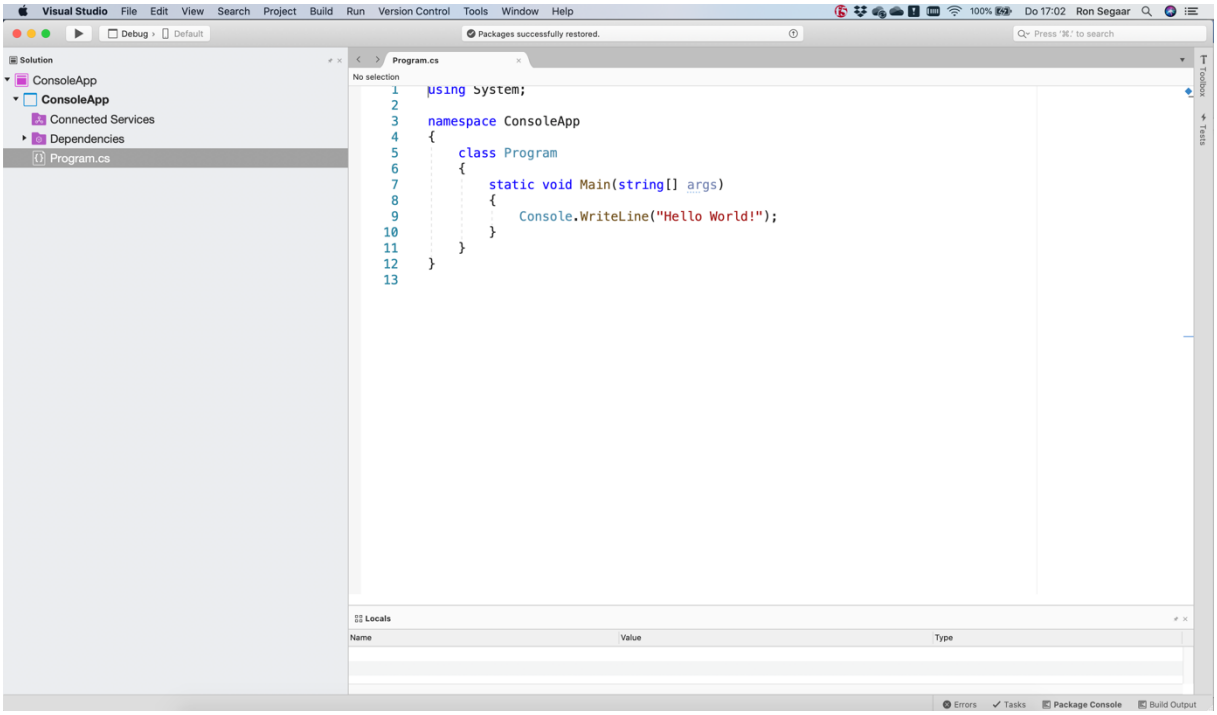
In de video's wordt gebruik gemaakt van de Windows versie. De schermen die je op een Mac ziet bij het aanmaken van een project (op een Mac heet dit een "new solution") zien er als volgt uit:



Kies het type project (Console Application).



Voer de naam van het project in en bepaal waar deze opgeslagen moet worden.



Het scherm na het starten van een consoleproject op een Mac.

De code die je krijgt na het aanmaken van een consoleproject ziet er als volgt uit:

```
1   using System;
2
3   namespace ConsoleApp
4   {
5       class Program
6       {
7           static void Main(string[] args)
8           {
9               Console.WriteLine("Hello World!");
10          }
11      }
12  }
```

De code bestaat uit een *class Program* met daarin de *Main* functie. Deze *Main* functie wordt automatisch uitgevoerd, zodra het programma opgestart wordt. De code die jij gaat schrijven (zolang je zelf nog geen functies maakt in C#) zet je dus tussen de accolades van regel 8 en 10.

OEFENINGEN

OEFENING 1A:

Maak nu zelf een consoleproject aan met de naam EersteProject (let op de hoofdletters). Pas de tekst die afgedrukt wordt aan in "Hallo Allemaal" en start het project eens op. Kijk goed wat er allemaal gebeurt.

OEFENING 1B:

Zoek via de Finder (Mac) of verkenner (Windows) de locatie op waar Visual Studio het project heeft opgeslagen. Bekijk de map die aangemaakt is en wat daarin is opgeslagen.

Let Op:

Als je het project wilt kopiëren of verplaatsen, moet je de gehele map gebruiken. De hele map met inhoud vormt gezamenlijk het project. Als je alleen de code-bestanden kopieert, moet je deze later handmatig in een nieuw project plakken. Dit is niet handig en veel werk.

Let Op 2:

Als je zelf de locatie bepaalt waar je het project opslaat, kies dan NOOIT voor een externe harde schijf (SSD) of een map die gedeeld is met een clouddienst zoals One drive. Visual Studio kan hier niet goed mee omgaan en dat geeft dus problemen

Sla het project dus altijd op de lokale schijf op (bijvoorbeeld op het bureaublad) en kopieer aan het einde van de les de gehele map naar jouw SSD of One drive.

LES 1 - 2 UUR – 2 VARIABELEN (DECLAREREN, INITIALISEREN, GEBRUIKEN)

INLEIDING

Bij JavaScript hebben we geleerd dat een variabele gedeclareerd moet worden met *let*:

```
let cijfer
```

Hiermee is de variabele aangemaakt. Het initialiseren (voor het eerst een waarde geven) werkt op de volgende manier:

```
cijfer = 8;
```

Er is wel een verschil met C#. In JavaScript kan je met *let* verschillende soorten variabelen declareren zoals getallen of teksten (strings). In C# moet je deze verschillen bij de declaratie al aangeven.

EXTERNE BRONNEN

Bekijk nu de video *Variabelen in C#.mp4*

THEORIE

Bekijk eerst de film met uitleg die hierboven staat aangegeven.
In dit stukje film worden vijf type variabelen in C# besproken:

- int gehele getallen voorbeelden: -4 -1 0 3 1001
- string tekenreeks (tekst) voorbeelden: "hallo" "@#raar" "123"
- float gebroken getal voorbeelden: 3.14 0.78945
- double gebroken getallen (met een grotere nauwkeurigheid)
- bool boolean die alleen true of false kan zijn.

Opmerking:
In C# bestaat ook de mogelijkheid om *var* te gebruiken voor de declaratie:

```
var tekst = "hallo";
```

Dit kan alleen als de variabele ook direct een waarde krijgt (initialisatie). Deze manier wordt als standaard gebruikt in de praktijk.

Toch raden wij dit voor nu af. Om meer bewust te worden van het datatype kan je het voorlopig beter zo doen:

```
string tekst;
tekst = "hallo";
```

of

```
string tekst = "hallo";
```

Variabelen van verschillende types worden in het computergeheugen op verschillende manieren opgeslagen. Daarom kan je ze ook niet ongestraft door elkaar gebruiken. Hieronder zie je een aantal voorbeelden die **niet** goed gaan.

Een geheel getal (int) en een gebroken getal (float) bij elkaar optellen en daarna opslaan in een integer variabele:

```
int geheelGetal = 5;
float gebrokenGetal = 4.6f;
int antwoord = geheelGetal + gebrokenGetal;
```

Een geheel getal (int) en een tekst (string) bij elkaar optellen en daarna opslaan in een integer variabele:

```
int geheelGetal = 5;
string getalTekst = "123";
int antwoord = geheelGetal + getalTekst;
```

Dit laatste voorbeeld zou wel goed gaan als het resultaat in een string opgeslagen wordt:

```
int geheelGetal = 5;
string getalTekst = "123";
string antwoord = geheelGetal + getalTekst;
```

OEFENINGEN

OEFENING 1C:

Probeer nu met vergelijkbare stukjes code uit, welke variabelen bij elkaar opgeteld kunnen worden en wat het type is wat daar uitkomt. Vul de resultaten in de onderstaande tabel in (een aantal zijn al ingevuld). Geef door middel van een X aan als het helemaal niet kan.

De berekening is: variabele van TypeC = variabele van TypeA + variabele van TypeB

Wat Type C moet zijn vul je in de tabel in:

		Variabele van TypeB				
	+	int	string	float	double	bool
Variabele van TypeA	int	int		float		
	string	string	string			
	float			float		
	double				double	
	bool					X

Bovenstaande tabel kan je apart downloaden als Word-document: Oefening_1C.docx

Voor meer uitleg kan je nog het volgende filmpje bekijken:
Oefening_1C.mp4

In het project *Oefening_1C* is een begin gemaakt met de code en in commentaarregels staan de vervolgstappen aangegeven. Download voor dit project het bestand Oefening_1C.zip

LES 1 - 2 UUR – 3 CONVERSIE EN INPUT

INLEIDING

In het vorige hoofdstuk hebben we gezien dat er verschillende type variabelen zijn die we niet zomaar door elkaar kunnen gebruiken. Vaak heb je een omzetting nodig van het ene type naar het andere type. Dit noemen we een conversie. In dit hoofdstuk zullen we een aantal van deze conversies bekijken.

In dit hoofdstuk zullen we naast conversie ook bekijken hoe de gebruiker van een console-applicatie gegevens kan invoeren die in de applicatie gebruikt worden.

EXTERNE BRONNEN

Dit is een video waar een introductie over conversie behandeld wordt.
https://www.youtube.com/watch?v=ZO7pRhNd4Ow

THEORIE

Bekijk de hierboven genoemde video over conversie. In deze video wordt getoond hoe je een string naar een float converteert en andersom. De reden waarop dit nodig is, wordt ook aangegeven. Alle input van de gebruiker is in eerste instantie een string. Als je ermee wilt rekenen, dan moet je eerst de omzetting gebruiken.

Voor omzettingen van een string naar een getal zijn er de volgende mogelijkheden:

```
string getalTekst = "1234";                    //tekst die geconverteerd wordt
int intGetal = int.Parse(getalTekst);          //omzetting naar een int
float floatGetal = float.Parse(getalTekst);    //omzetting naar een float
double doubleGetal = double.Parse(getalTekst); //omzetting naar een double
```

LET OP:

Als de string een tekst bevat die niet in een getal is om te zetten (zoals "ABC") dan zal de applicatie stoppen en een error geven. Hou hier dus rekening mee. Hoe we dit op een nette manier kunnen oplossen zullen we later leren.

Voor omzettingen van een getal naar een string zijn dat de volgende mogelijkheden:

```
intGetal.ToString();
floatGetal.ToString();
doubleGetal.ToString();
```

Voor de getallen onderling kan het ook nodig zijn om de ene in de andere om te zetten. Om een int naar een float of double om te zetten hoeft niets bijzonders te gebeuren. Je kan een int gebruiken om een float of double variabele te vullen met een waarde. De omzetting gebeurt dan volledig automatisch:

```
int getal = 6;
float floatGetal;
floatGetal = getal;

double doubleGetal;
doubleGetal = getal;
```

Andersom, van een float of double naar een int, heeft wel extra aandacht nodig:

```
float pi = 3.14f;
int getal = (int) pi;
```

Het getal pi (3,14) dat op de eerste regel als float gedeclareerd is, wordt op de tweede regel naar een int gekopieerd. Dit lukt alleen door tussen haakjes int erbij te zetten. Omdat er in een int geen gebroken getallen opgeslagen worden, zullen de getallen achter de komma weggegooid worden (dus geen afronding). Het int variabele getal krijgt dus de waarde 3.

De omzetting van een float naar een double kan weer zonder problemen (omdat een float minder nauwkeurig is dan een double):

```
float pi = 3.14f;
double getal = pi;
```

De omzetting andersom moet weer met een extra notatie tussen haakjes.

```
double pi = 3.14;
float getal = (float) pi;
```

In de bekeken video heb je ook kunnen zien dat je een commando kunt gebruiken om gegevens (als een string), die door een gebruiker ingevoerd worden, in te lezen. Dit heb je nodig om de oefeningen en opdracht te kunnen maken.

OEFENINGEN

OEFENING 1D:

Maak een applicatie waar een gebruiker twee gehele getallen moet invoeren. Op de console wordt met tekst duidelijk aangegeven dat er een getal ingevoerd moet worden. Na het inlezen van de twee getallen geeft de applicatie twee waarden als resultaat, namelijk de vermenigvuldiging van de getallen en de deling. De deling moet er als een gebroken getal uitkomen. Op de console ziet het resultaat er als volgt uit (als de getallen 3 en 4 worden ingevoerd):

```
Voer het eerste getal in:
3
Voer het tweede getal in:
4
De vermenigvuldiging is: 12
De deling is: 0.75
```

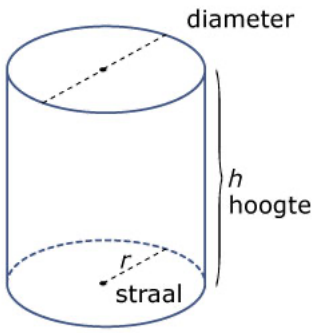
Voor meer uitleg kan je nog het volgende filmpje bekijken:
Oefening_1D.mp4

In het project *Oefening_1D* is een begin gemaakt van de code en in commentaarregels staan de vervolgstappen aangegeven. Download voor dit project het bestand Oefening_1D.zip

LES 1 - 2 UUR – 4 OPDRACHT 1-1

DE CILINDER

In deze opdracht maak je een applicatie die aan de gebruiker vraagt wat de hoogte en straal van een cilinder is. Voor de invoer kan de gebruiker een gebroken getal invoeren.



Na invoer van de hoogte en straal wordt de oppervlakte en de inhoud van de cilinder berekend en afgedrukt. De applicatie kan er als volgt uitzien:

```
Wat is de hoogte van de cilinder
4.5
Wat is de straal van de cilinder
2.3
De oppervlakte van de cilinder is: 98.26901547787759
De inhoud van de cilinder is: 74.78561001778687
```

Voor de berekening kan je gebruik maken van de volgende formules:

Oppervlakte Cilinder = oppervlakte Wand + 2 x oppervlakte Cirkel
Inhoud Cilinder = oppervlakte Cirkel x hoogte
Waarbij:
oppervlakte Wand = hoogte * omtrek Cirkel
oppervlakte Cirkel = PI x straal x straal
omtrek Cirkel = 2 x PI x straal

Het getal Pi (3,14.....) zit standaard in C#. Gebruik in je berekening hiervoor: `Math.PI`

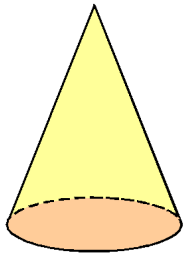
Voor meer uitleg kan je nog het volgende filmpje bekijken:
Opdracht1_1.mp4

In het project Opdracht1_1 is een begin gemaakt met de code en in commentaarregels staan de vervolgstappen aangegeven. Download voor dit project het bestand Opdracht1_1.zip

BENODIGDE KENNIS

- Ik declareer en initialiseer variabelen in C#
- Ik maak berekeningen met variabelen in C#
- Ik lees inputvelden uit in C#
- Ik converteer variabelen naar een ander type (indien nodig) in C#

UITBREIDING (EXTRA)



- Bereken ook de oppervlakte en inhoud van een kegel
- Zoek hiervoor zelf de formules op
- Vraag aan de gebruiker ook de hoogte en de straal

LES 2 - 2 UUR - 1 BESLISSTRUCTUREN

INLEIDING

Om beslissingen te kunnen nemen in een programma heb je beslisstructuren nodig. In de lessen JavaScript ben je deze al tegengekomen in de vorm van if en if-else structuren:

```
<script>
  let leeftijd = window.prompt("Hoe oud ben je?");
  if ( leeftijd >= 13 && leeftijd <= 18 ) {
    alert("Je bent een puber");
  } else {
    alert("Je bent geen puber");
  }
</script>
```

In C# is dit precies hetzelfde. Je kan dus gebruik maken van de kennis die je al opgedaan hebt.

EXTERNE BRONNEN

Dit is een video waarin de verschillende beslisstructuren in C# behandeld worden:
https://www.youtube.com/watch?v=_2CiUU-BIgQ

In deze video zie je nog een voorbeeld:
https://www.linkedin.com/learning/learning-c-sharp-3/work-with-conditions?u=36995692

In deze video zie je hoe je &&- en ||-tekens kan gebruiken om combinaties van vergelijkingen in een if-statement te maken:
https://www.youtube.com/watch?v=SHH85xHEH9I

THEORIE

In C# bestaat net zoals bij JavaScript het *if* statement. Deze kan gebruikt worden in combinatie met het *else* statement. Hoe dit in C# gebruikt kan worden, kan je zien in de video die hierboven genoemd is.

Als je deze video bekijkt, let dan goed op de posities waar de accolades { en } neergezet worden. Voor C# zijn daar andere afspraken voor dan bij JavaScript. Visual Studio zal deze posities ook automatisch afdwingen (door de accolades op de juiste plek te zetten).

In de video zie je ook hoe in C# het switch statement gebruikt kan worden en in welke situatie deze handig kan zijn.

In de derde video zie je hoe je meerdere vergelijkingen kan maken binnen een if statement. Hiervoor worden de &&- en ||-tekens gebruikt. Hieronder zie je nog een voorbeeld met meerdere vergelijkingen:

```
Console.WriteLine("voer je leeftijd in");
string leeftijdString = Console.ReadLine();
int leeftijd = int.Parse(leeftijdString);

if (leeftijd < 1 || (leeftijd >= 35 && leeftijd <= 45 ) || leeftijd > 80)
{
    Console.WriteLine("Je bent nog een baby of al behoorlijk oud of op een gemiddelde leeftijd");
}
```

Let goed op het stukje code waarin gecontroleerd wordt of de leeftijd tussen de 35 en 45 jaar zit. Dit deel staat apart tussen haakjes, omdat deze controle als eerste uitgevoerd moet worden.

Als bovenstaande code niet duidelijk is, bekijk dan de volgende video:
if-statement_uitleg.mp4

OEFENINGEN

OEFENING 2A:

Maak een applicatie waar een gebruiker twee gehele getallen moet invoeren. Op de console wordt met tekst duidelijk aangegeven dat er een getal ingevoerd moet worden. Na het inlezen van de getallen wordt gecontroleerd welke van de twee het grootst is. Dit getal wordt afgedrukt. Als de ingevoerde getallen precies gelijk aan elkaar zijn, wordt dit ook weergegeven. De output ziet er als volgt uit:

```
Voer een getal in
5
Voer een tweede getal in
7
getal-2: 7 is groter dan getal-1: 5
```

Of als de getallen gelijk zijn:

```
Voer een getal in
6
Voer een tweede getal in
6
getal-1 en getal-2 zijn gelijk
```

OEFENING 2B:

Maak een applicatie waar een gebruiker drie gehele getallen moet invoeren. Op de console wordt met tekst duidelijk aangegeven dat er een getal ingevoerd moet worden. Na het inlezen van de getallen wordt eerst gecontroleerd of één van deze getallen buiten de grenzen van 1 t/m 10 valt (dus: is een getal kleiner dan 1 of groter dan 10). Als dat zo is, wordt een foutmelding afgedrukt. Als alle getallen wel binnen de grenzen vallen, worden het grootste en het kleinste getal afgedrukt. De uitvoer ziet er als volgt uit:

```
Voer een getal in
5
Voer een tweede getal in
7
Voer een derde getal in
2
Het grootste getal is 7
Het kleinste getal is 2
```

Of als een getal buiten de grenzen valt:

```
Voer een getal in
3
Voer een tweede getal in
7
Voer een derde getal in
12
Een getal valt buiten de grenzen van 1 .. 10
```

Voor meer uitleg kan je nog het volgende filmpje bekijken:
Oefening_2B.mp4

In het project Oefening_2B is een begin gemaakt met de code en in commentaarregels staan de vervolgstappen aangegeven. Download voor dit project het bestand Oefening_2B.zip

LES 2 - 2 UUR – 4 OPDRACHT 2-1

MAANDEN VAN HET JAAR

In deze opdracht maak je een applicatie die aan de gebruiker vraagt om een getal in te voeren van 1 tot en met 12 (1 en 12 doen ook mee). Dit getal geeft aan welke maand van het jaar het is. Bij invoer van bijvoorbeeld het getal 1 zal de applicatie aangeven dat het januari is.
Indien het getal buiten de grenzen van 1 t/m 12 valt, wordt er een foutmelding gegeven.

De applicatie kan er als volgt uitzien:

```
Voer een getal in van 1 .. 12
11
november
```

Of bij invoer buiten de grenzen:

```
Voer een getal in van 1 .. 12
34
dit getal geeft geen maand van het jaar aan
```

Voor meer uitleg kan je nog het volgende filmpje bekijken:
Opdracht2_1.mp4

In het project Opdracht2_1 is een begin gemaakt met de code en in commentaarregels staan de vervolgstappen aangegeven. Download voor dit project het bestand Opdracht2_1.zip

BENODIGDE KENNIS

- Ik pas in C# de juiste syntax toe
- Ik declareer en initialiseer variabelen in C#
- Ik gebruik beslisstructuren in C#
- Ik lees inputvelden uit in C#
- Ik converteer variabelen naar een ander type (indien nodig) in C#

UITBREIDING (EXTRA)

- Probeer deze opdracht te maken met het switch statement

- Voeg ook de dagen van de week (1 t/m 7) toe, waarbij afgedrukt wordt of het maandag, dinsdag , enzovoort is.

LES 2 - 2 UUR – 4 OPDRACHT 2-2

ALFABETISCHE VOLGORDE

In deze applicatie vraag je aan de gebruiker om drie verschillende woorden in te voeren. De applicatie zet deze woorden in alfabetische volgorde en drukt ze daarna achter elkaar af (op 1 regel). Om woorden met elkaar te vergelijken, kan je de volgende code gebruiken:

```
if (woord1.CompareTo(woord2) > 0)
```

Met het CompareTo commando vergelijk je twee strings. Als er een positief getal uitkomt, is woord1 groter dan woord2 (verderop in het alfabet). Als er een negatief getal uitkomt, is woord1 kleiner dan woord2.

De applicatie kan er als volgt uitzien:

```
Voer een woord in:
fiets
Voer een tweede woord in:
auto
Voer een derde woord in:
aap
in alfabetische volgorde: aap auto fiets
```

Voor meer uitleg kan je nog het volgende filmpje bekijken:
Opdracht2_2.mp4
In deze video wordt een oplossing aangegeven die wel goed werkt en het eenvoudigst is om te begrijpen, maar die niet de snelste oplossing is. De docent kan eventueel nog een snellere oplossing uitleggen.

In het project Opdracht2_2 is een begin gemaakt met de code en in commentaarregels staan de vervolgstappen aangegeven. Download voor dit project het bestand Opdracht2_2.zip

BENODIGDE KENNIS

- Ik pas in C# de juiste syntax toe
- Ik declareer en initialiseer variabelen in C#

- Ik gebruik beslisstructuren in C#
- Ik lees inputvelden uit in C#
- Ik converteer variabelen naar een ander type (indien nodig) in C#

UITBREIDING (EXTRA)

- Probeer deze opdracht ook te maken met 4 woorden
- Probeer deze opdracht te maken met 6 woorden. (Je mag hierbij ook andere middelen gebruiken zoals herhalingslussen of functies. Zoek zelf op hoe je die kan gebruiken)

LES3 - 2 UUR - 1 HERHALINGSLUSSEN

INLEIDING

Net zoals in JavaScript kunnen we in C# gebruik maken van herhalingslussen. Net als bij de beslisstructuren lijkt de code heel erg veel op die van JavaScript. Ook hier hebben we een for-lus en een while-lus. Er zijn nog andere mogelijkheden, maar die zullen we niet in deze module behandelen.

EXTERNE BRONNEN

Op de volgende pagina's van de site van W3schools zie je voorbeelden van de for-lus en while-lus:
https://www.w3schools.com/cs/cs_for_loop.asp
https://www.w3schools.com/cs/cs_while_loop.asp

In deze video zie je nog een voorbeeld van een while-lus:
https://www.linkedin.com/learning/search?keywords=c%23%20loops&u=36995692

In deze video wordt een while-lus omgezet naar een for-lus:
https://www.linkedin.com/learning/search?keywords=c%23%20loops&u=36995692

In deze video wordt gebruik gemaakt van een for-lus binnen een for-lus:
https://www.linkedin.com/learning/search?keywords=c%23%20loops&u=36995692

THEORIE

Om een goed beeld te krijgen van hoe de verschillende herhalingslussen gebruikt worden in C# kan je het beste eerst de bovenstaande drie video's bekijken. Let daarbij goed op de verschillen met de taal JavaScript. Je zal zien dat die alleen zitten in de declaratie van de variabelen en de positie van de accolades.

In de laatste video zie je een herhalingslus (for-lus) binnen een herhalingslus. Dit hebben we ook al bij JavaScript gebruikt. In het voorbeeld hieronder maken we hiervan gebruik om een vierkant, dat uit sterretjes bestaat, te tekenen.

```
for (int y = 1; y < 10; y++)
{
    for (int x = 1; x < 10; x++)
    {
        Console.Write("* ");
    }
    Console.WriteLine();
}
```

In de binnenste for-lus wordt er elke keer een * afgedrukt (met een extra spatie om de tussenruimte in horizontale richting gelijk te maken aan die bij de verticale richting). Dit wordt gedaan met een Console.Write commando. Hierdoor wordt wel de tekst afgedrukt, maar wordt nog niet naar de volgende regel gesprongen. Het volgende sterretje komt er dus achter te staan.

Na de uitvoering van de binnenste herhalingslus, wordt met het commando Console.WriteLine() naar de volgende regel gesprongen. In dit commando zit geen nieuwe tekst die afgedrukt wordt. De buitenste herhalingslus zal er nu voor zorgen (als y nog geen 10 geworden is) dat de binnenste herhalingslus nogmaals uitgevoerd wordt. Er wordt dus nog een keer een regel met sterretjes afgedrukt. De uiteindelijke uitkomst wordt dan:

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

OEFENINGEN

OEFENING 3A:

Maak gebruik van het voorbeeld hierboven en pas de code aan, zodat alleen de buitenkant van het vierkant getekend wordt:

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

Tip: Bedenk goed bij welke waarden van x en y er een sterretje afgedrukt moet worden of een spatie.

OEFENING 3B:

Pas oefening 3A aan, zodat de gebruiker het aantal sterretjes, die in de lengte en breedte van het vierkant getekend moet worden, invoert. De uitvoer van het programma gaat er dan zo uitzien:

```
Voer een getal in:
7
* * * * * * *
*           *
*           *
*           *
*           *
*           *
* * * * * * *
```

LES 3 - 2 UUR – 2 OPDRACHT 3-1

VIERKANT MET DIAGONALEN

In deze opdracht maak je een programma zoals in de oefeningen hierboven. Het programma tekent een vierkant, waarin ook de diagonalen getekend worden. De gebruiker geeft weer de lengte van een zijde op (dus het aantal sterretjes).

De applicatie kan er als volgt uitzien:

```
Voer een getal in:
10
* * * * * * * * * *
* *             * *
*   *         *   *
*     *     *     *
*       * *       *
*       * *       *
*     *     *     *
*   *         *   *
* *             * *
* * * * * * * * * *
```

Tip: Een sterretje voor de diagonaal van linksboven naar rechtsonder wordt getekend als x en y (zie voorbeeld code bij de theorie) gelijk zijn. Bedenk zelf welke waarden x en y moeten hebben voor de andere diagonaal.

Voor meer uitleg kan je nog het volgende filmpje bekijken:
Opdracht3_1.mp4

In het project Opdracht3_1 is een begin gemaakt met de code en in commentaarregels staan de vervolgstappen aangegeven. Download voor dit project het bestand Opdracht3_1.zip

BENODIGDE KENNIS

- Ik pas in C# de juiste syntax toe
- Ik gebruik beslisstructuren in C#
- Ik lees inputvelden uit in C#
- Ik maak gebruik van herhalingslussen ('for' en 'while') in C#

UITBREIDING (EXTRA)

- Probeer er een huisje van te maken door er een dak boven te tekenen.

```
Voer een getal in:
8
      * *
    *     *
  *         *
*             *
* * * * * * * *
* *         * *
*   *     *   *
*     * *     *
*     * *     *
*   *     *   *
* *         * *
* * * * * * * *
```

Tip: Gebruik extra herhalingslussen.

LES 3 - 2 UUR – 3 OPDRACHT 3-2

KLEINSTE GETAL, GROOTSTE GETAL EN GEMIDDELDE

In deze opdracht maak je een applicatie, waarin elke keer opnieuw aan de gebruiker gevraagd wordt om een getal in te voeren. Na het invoeren van het getal worden het kleinste getal, het grootste getal en het gemiddelde van alle tot dan toe ingevoerde waarden afgedrukt. De herhaling stopt als de gebruiker -1 als getal invoert. Hierbij krijgt de gebruiker wel een melding dat de applicatie beëindigd is.

Het geheel kan er als volgt uitzien:

```
voor een getal in:
23
Het kleinste getal is: 23
Het grootste getal is: 23
Het gemiddelde is: 23
voor een getal in:
576
Het kleinste getal is: 23
Het grootste getal is: 576
Het gemiddelde is: 299,5
voor een getal in:
29
Het kleinste getal is: 23
Het grootste getal is: 576
Het gemiddelde is: 209,3333
voor een getal in:
12
Het kleinste getal is: 12
Het grootste getal is: 576
Het gemiddelde is: 160
voor een getal in:
-1
Bedankt voor het gebruiken van deze applicatie
```

Voor meer uitleg kan je nog het volgende filmpje bekijken:
Opdracht3_2.mp4

In het project Opdracht3_2 is een begin gemaakt met de code en in commentaarregels staan de vervolgstappen aangegeven. Download voor dit project het bestand Opdracht3_2.zip

BENODIGDE KENNIS

- Ik maak berekeningen met variabelen in C#
- Ik gebruik beslisstructuren in C#
- Ik maak gebruik van herhalingslussen ('for' en 'while') in C#

UITBREIDING (EXTRA)

- Zorg dat de getallen die ingevoerd worden, alleen tussen de 0 en 100 mogen zitten. Als er een te groot getal wordt ingevoerd, wordt er een foutmelding afgedrukt.

LES4 - 2 UUR - 1 DEBUGGEN

INLEIDING

Het werk van een programmeur bestaat naast het schrijven van code ook voor een groot deel uit het oplossen van fouten. De problemen uit de code halen (die altijd wel ergens ontstaan) wordt debuggen genoemd. In een ontwikkelomgeving zoals Visual Studio worden hiervoor extra mogelijkheden geboden. In dit hoofdstuk gaan we deze mogelijkheden bekijken en gaan we bestaande problemen oplossen.

EXTERNE BRONNEN

In de volgende video kan je zien hoe je in Visual Studio kan debuggen: VS-debuggen.mp4

THEORIE

In Visual Studio hebben we in het voorgaande al gezien dat bepaalde fouten aangegeven worden door rode kringellijntjes onder de code: `aantal++;`

In zo'n situatie weet je al dat je iets moet aanpassen. Dit is meestal een typefout of een variabele die niet gedeclareerd is (zoals hierboven). Dit noemen we syntax fouten.

Er kunnen zich ook nog geheel andere soorten fouten voordoen. Als de code opgestart wordt, werkt de applicatie wel, maar geeft niet de verwachte resultaten. Dan is er geen syntaxfout gemaakt, maar een structuurfout. Ook deze moeten we kunnen oplossen. Hiervoor zijn breaktekens bedacht, waarmee je de code tussentijds kan laten pauzeren. Op een pauzemoment kan je als programmeur de waarden van de variabelen controleren op juistheid.

Hoe het debuggen in Visual Studio gebruikt kan worden, kan je zien in de video die hierboven genoemd is. Bekijk nu dus eerst de video.

OEFENING 4A:

Download en open het project met de naam Oefening_4A. Dit project kan je vinden in het bestand Oefening_4A.zip.

In dit programma worden de lengte, breedte en hoogte van een blok ingevoerd door de gebruiker. Dit mogen gebroken getallen zijn. Na invoer worden de oppervlakte en de inhoud van het blok afgedrukt. Er zitten alleen een aantal fouten in. Haal alle fouten uit deze code, zodat de applicatie werkt zoals gewenst.

OEFENING 4B:

Download en open het project met de naam Oefening_4B. Dit project kan je vinden in het bestand Oefening_4B.zip.

Dit programma zal na opstarten het volgende moeten afdrukken:

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

Dit gebeurt helaas niet omdat er meerdere fouten in het programma zitten. Haal alle fouten uit deze code, zodat de applicatie werkt zoals gewenst.

LES 4 - 2 UUR – 3 OPDRACHT 4-1

DELEN DOOR

Download en open het project met de naam Opdracht4_1. Dit project kan je vinden in het bestand Opdracht4_1.zip.

In dit programma wordt aan de gebruiker gevraagd om een getal in te voeren. Dit getal wordt daarna door 2 gedeeld, waarna de uitkomst wordt getoond. Daarna wordt het getal door 3 gedeeld en wordt de uitkomst weer getoond. Hetzelfde met 4, 5, 6 enzovoort. Dit loopt zo door totdat de uitkomst kleiner wordt dan 1.

Dit programma zal na opstarten het volgende moeten afdrukken:

```
Voer een getal in
13
13 gedeeld door 2 is 6,5
13 gedeeld door 3 is 4,333333
13 gedeeld door 4 is 3,25
13 gedeeld door 5 is 2,6
13 gedeeld door 6 is 2,166667
13 gedeeld door 7 is 1,857143
13 gedeeld door 8 is 1,625
13 gedeeld door 9 is 1,444444
13 gedeeld door 10 is 1,3
13 gedeeld door 11 is 1,181818
13 gedeeld door 12 is 1,083333
13 gedeeld door 13 is 1
13 gedeeld door 14 is 0,9285714
```

Dit gebeurt helaas niet, omdat er meerdere fouten in het programma zitten. Haal alle fouten uit deze code, zodat de applicatie werkt zoals gewenst.

BENODIGDE KENNIS

- Ik maak berekeningen met variabelen in C#
- Ik debug C# code met behulp van breaktekens

UITBREIDING (EXTRA)

- Geef in de code met behulp van commentaarregels aan welke fouten je hebt gevonden

LES 4 - 2 UUR – 4 OPDRACHT 4-2

KERSTBOOM

Download en open het project met de naam Opdracht4_2. Dit project kan je vinden in het bestand Opdracht4_2.zip.

In dit programma wordt een kerstboom bestaande uit sterretjes getekend.

Dit programma zal na het opstarten het volgende moeten afdrukken:

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

Dit gebeurt helaas niet, omdat er meerdere fouten in het programma zitten. Haal alle fouten uit deze code, zodat de applicatie werkt zoals gewenst.

BENODIGDE KENNIS

- Ik debug C# code met behulp van breaktekens

UITBREIDING (EXTRA)

- Geef in de code met behulp van commentaarregels aan welke fouten je hebt gevonden

LES4 - 2 UUR - 4 JAVASCRIPTCODE OMZETTEN NAAR C#

INLEIDING

Inmiddels heb je kennis gemaakt met de taal JavaScript en de taal C#. Je hebt als het goed is, ook gezien dat er veel overeenkomsten zijn. Het grootste verschil zit in het declareren en het gebruiken van variabelen. Ook de invoer van gebruikers en uitvoer naar het scherm zijn anders, omdat de C#-applicaties van een ander type (console-applicatie) zijn dan de Javascript (web) applicaties. In dit hoofdstuk gaan we een aantal JavaScript-applicaties omzetten naar C#-applicaties.

In dit hoofdstuk vind je geen theorie of oefeningen. We gaan meteen door met de opdrachten.

LES 4 - 2 UUR – 5 OPDRACHT 4-3

GOEDEDAG

Zet de volgende JavaScriptcode om in C#:

```
<script>
  let datum = new Date();      // geeft datum/tijd nu
  let uur = datum.getHours(); // haalt het uur uit let datum

  if (uur >= 6 && uur < 12)
  {
    document.write("Goedemorgen");
  }
    else if(uur >= 12 && uur < 18)
  {
    document.write("Goedemiddag");
  }
    else if(uur >= 18 && uur < 23)
  {
    document.write("Goedeavond");
  }
    else
  {
    document.write("Goedenacht");
  }
</script>
```

De regel om de datum van vandaag te bepalen is in C#: `DateTime datum = DateTime.Now;`

De code om het uur uit de datum te halen is: `datum.Hour;`

BENODIGDE KENNIS

- Ik zet bestaande Javascriptcode om naar C#

LES 4 - 2 UUR – 6 OPDRACHT 4-4

HET NUMMER

Zet de volgende JavaScriptcode om in C#:

```
var text = "";
var i = 0;
while (i < 5) {
  text += "<br>The number is " + i;
  i++;
  if (i === 3) {
    break;
  }
}
document.getElementById("demo").innerHTML = text;
```

Om in C# naar een volgende regel te kunnen springen, gebruik je geen <br> in de string, maar \n
De tekst wordt niet afgedrukt op een webpagina, maar op de console.

BENODIGDE KENNIS

- Ik zet bestaande Javascriptcode om naar C#

LES5 - 2 UUR - 1 BEWERKINGEN OP STRING VARIABELEN

INLEIDING

Tot nu toe hebben we stringvariabelen voornamelijk gebruikt bij het invoeren van data door de gebruiker van de applicatie. Daarna werd de stringvariabele direct geconverteerd naar een integer of float om ermee te kunnen rekenen. Er is echter nog veel meer mogelijk met stringvariabelen. Een voorbeeld dat we al tegengekomen zijn, is:

```
if (woord1.CompareTo(woord2) > 0)
```

Met het CompareTo commando kunnen we twee woorden met elkaar vergelijken. Ook hebben we al stringvariabelen aan elkaar gekoppeld met plustekens ertussen. In dit hoofdstuk gaan we gebruik maken van andere mogelijkheden met stringvariabelen:

- Bepalen van de lengte van een string
- Interpolatie (waarden van variabelen invoegen in een string)
- Special characters (hiervan zijn we de \n al tegengekomen)
- Substring (een deel van de string gebruiken)
- Trim (spaties aan het begin of eind verwijderen)
- Hoofdletters en kleine letters aanpassen

EXTERNE BRONNEN

De volgende links verwijzen naar een W3schools pagina, waar een aantal mogelijkheden met strings behandeld worden: https://www.w3schools.com/cs/cs_strings.asp

In de volgende video worden de volgende mogelijkheden besproken: Trim, Substring, hoofdletter en kleine letters:

https://www.linkedin.com/learning/search?keywords=C%23%20string&u=36995692

In deze korte video wordt nog het nut van interpolatie getoond:

https://www.linkedin.com/learning/search?keywords=C%23%20string&u=36995692

THEORIE

In bovenstaande externe bronnen, wordt de basis van stringbewerkingen besproken. In de eerste video wordt een speciaal venster gebruikt (zoals de inspector-console in de browser bij JavaScript) waar de commando's direct toegepast kunnen worden en direct een uitkomst geven. In de code hieronder zie je dat deze in een programma iets anders gebruikt moeten worden:

```
string tekst = "   Dit is een tekst MET Rare HOOftletters   ";
tekst = tekst.Trim();

string tekst2 = tekst.ToLower();

int aantalTekens = tekst2.Length;

Console.WriteLine($"De lengte van de tekst na het Trim commando is {aantalTekens}");
Console.WriteLine(tekst);
Console.WriteLine(tekst2);
```

Deze code zal het volgende afdrukken:

```
De lengte van de tekst na het Trim commando is 38
Dit is een tekst MET Rare HOOfdLetters
dit is een tekst met rare hoofdletters
```

Wat opvalt, is dat in deze code elke keer voor de bewerking op de string een = teken staat met daar weer een variabele voor. Het resultaat van de bewerking wordt, in de variabele die voor het = teken staat, opgeslagen. Op de tweede regel is dat dezelfde variabele waarop je de bewerking uitvoert. Je past daarmee de variabele (in dit geval tekst) zelf aan.

Op de derde regel wordt op de variabele tekst het ToLower commando uitgevoerd (alle letters worden kleine letters). Het resultaat wordt in een nieuwe variabele opgeslagen (tekst2). De variabele tekst verandert nu niet. Dat zie je aan het resultaat als deze afgedrukt wordt (een na laatste coderegel).

Als we de lengte van een string bepalen, komt daar een integer variabele uit. Dat zie je op de vierde coderegel.

OEFENING 5A:

Neem bovenstaande code over en pas deze zodanig aan dat in de onderste afgedrukte regel niet meer de tekst *met rare* staat. Deze wordt vervangen door het woord *zonder*:

```
De lengte van de tekst na het Trim commando is 38
Dit is een tekst MET Rare HOOfdLetters
dit is een tekst zonder hoofdletters
```

Tip: gebruik het Substringcommando om de regel in twee stukken op te delen, waarin de tekst *met rare* niet meer voorkomt.

OEFENING 5B:

In de theorie is nog niet besproken dat je met het commando IndexOf precies de positie van een woord in een string kan bepalen. Kijk maar eens naar het volgende voorbeeld:

```
int positie1 = tekst2.IndexOf("met");
```

In deze code wordt de positie bepaald van het woord *met* in de variabele *tekst2* uit de vorige oefening. Het getal dat hier uitkomt is 17 en dit wordt opgeslagen in de variabele *positie1.*

In deze oefening pas je nu de code aan van oefening 5a, waarbij je gebruik maakt van het IndexOf commando om te bepalen hoe de regel in delen opgesplitst moet worden.

LES 5 - 2 UUR – 2 OPDRACHT 5-1

AANTAL LETTERS A, E OF I

In deze opdracht maak je een applicatie die aan de gebruiker vraagt om een zin in te voeren. Na invoer van deze zin wordt door de applicatie bepaald hoeveel keer de letter A, de letter E en de letter I in de zin voorkomt. Het maakt niet uit of de letter als hoofdletter of als kleine letter in de zin staat. De uitvoer kan er als volgt uitzien:

```
Voer een volledige zin in
Dit is een zin die ingevoerd is om deze applicatie te testen
aantal keer A is: 2
aantal keer E is: 11
aantal keer I is: 8
```

Voor meer uitleg kan je nog het volgende filmpje bekijken:
Opdracht5_1.mp4

In het project Opdracht5_1 is een begin gemaakt met de code en in commentaarregels staan de vervolgstappen aangegeven. Download voor dit project het bestand Opdracht5_1.zip

BENODIGDE KENNIS

- Ik pas in C# de juiste syntax toe
- Ik voer in C# bewerkingen op strings uit
- Ik gebruik beslisstructuren in C#
- Ik maak gebruik van herhalingslussen ('for' en 'while') in C#

UITBREIDING (EXTRA)

- Zorg ervoor dat de klinkers die tegen elkaar aanstaan niet meer meegeteld worden bij het aantal keer A, E of I, maar als combinatie geteld worden. Dus in de zin "wie is er een aap" krijg je het resultaat:
    - 1x een I
    - 1x een e
    - 3x een combinatie (van wie, een en aap)

LES 5 - 2 UUR – 3 OPDRACHT 5-2

WOORDEN UIT DE ZIN

In deze opdracht maak je een applicatie die aan de gebruiker vraagt om een zin in te voeren. Na invoer van deze zin wordt door de applicatie bepaald uit welke afzonderlijke woorden de zin opgebouwd is. Elke woord wordt op een nieuwe regel afgedrukt. Elk woord dat afgedrukt wordt, begint met een hoofdletter en de rest zijn kleine letters. De uitvoer kan er als volgt uitzien:

```
Voer een zin in:
Dit is weer een hele leuke zin
Dit
Is
Weer
Een
Hele
Leuke
Zin
```

Voor meer uitleg kan je nog het volgende filmpje bekijken:
Opdracht5_2.mp4

In het project Opdracht5_2 is een begin gemaakt met de code en in commentaarregels staan de vervolgstappen aangegeven. Download voor dit project het bestand Opdracht5_2.zip

BENODIGDE KENNIS

- Ik pas in C# de juiste syntax toe
- Ik voer in C# bewerkingen op strings uit
- Ik gebruik beslisstructuren in C#
- Ik maak gebruik van herhalingslussen ('for' en 'while') in C#

UITBREIDING (EXTRA)

- Zorg dat de woorden op alfabetische volgorde worden afgedrukt.

LES6 - 2 UUR - 1 EINDOPDRACHT

CIJFER LEGGEN

In deze opdracht wordt aan de gebruiker gevraagd om vier cijfers met een waarde van 1 t/m 6 in één keer in te voeren. De gebruiker voert deze cijfers dus op één regel in, met elke keer een spatie ertussen. Het maakt niet uit of de gebruiker de cijfers als getal of als tekst invoert. Na de invoer worden de getallen als een soort blokje afgedrukt op het scherm. Hieronder zie je twee voorbeelden met verschillende invoer van de gebruiker:

```
voer vier getallen in van 1..6 gescheiden door een spatie (als getal of als tekst):
3 vijf 2 vier
****************************
*     **     **     **     *
*  3  **  5  **  2  **  4  *
*     **     **     **     *
****************************
```

```
voer vier getallen in van 1..6 gescheiden door een spatie (als getal of als tekst):
een 3 zes twee
****************************
*     **     **     **     *
*  1  **  3  **  6  **  2  *
*     **     **     **     *
****************************
```

Je hoeft geen rekening te houden met foutieve invoer.

Je kan het volgende stappenplan volgen:

- Splits de invoer in afzonderlijke woorden
- Controleer of een woord tekst bevat (een, twee, drie, vier , vijf of zes)
- Zo ja, zet dan een bepaalde integer variabele op de waarde die bedoeld is
- Indien dit niet zo is, dan is het een getal geweest. Converteer dit naar een integer
- Druk de getallen af met de sterretjes er omheen

BENODIGDE KENNIS

- Ik pas in C# de juiste syntax toe
- Ik declareer en initialiseer variabelen in C#
- Ik voer in C# bewerkingen op strings uit
- Ik gebruik beslisstructuren in C#
- Ik lees inputvelden uit in C#
- Ik maak gebruik van herhalingslussen ('for' en 'while') in C#
- Ik converteer variabelen naar een ander type (indien nodig) in C#

UITBREIDING (EXTRA)

- Controleer of er te veel spaties ingevoerd zijn (bijvoorbeeld twee achter elkaar)
- Contoleer of er andere getallen dan 1 t/m 6 ingevoerd zijn of een verkeerde tekst. Geef een foutmelding als de invoer niet correct is geweest i.p.v. het afdrukken van de blokjes.

BEOORDELING

De module wordt beoordeeld op de leerdoelen. Per opdracht staat (bij de opdracht) aangegeven om welk(e) leerdoel(en) het gaat. Een leerdoel van met de volgende waardering beoordeeld worden:

O = onvoldoende
M = matig (je doet iets nog niet helemaal goed)
V = voldoende
G = je doet het extra goed, bijvoorbeeld door een extra onderdeel te maken.

LEERDOELEN

De leerdoelen zijn:

- Ik pas in C# de juiste syntax toe
- Ik declareer en initialiseer variabelen in C#
- Ik voer in C# bewerkingen op strings uit
- Ik maak berekeningen met variabelen in C#
- Ik gebruik beslisstructuren in C#
- Ik lees inputvelden uit in C#
- Ik maak gebruik van herhalingslussen ('for' en 'while') in C#
- Ik converteer variabelen naar een ander type (indien nodig) in C#
- Ik zet bestaande Javascriptcode om naar C#
- Ik debug C# code met behulp van breaktekens
-

WERKPROCESSEN

De werkprocessen zijn:

- B1-K1-W3 : Realiseert (onderdelen van) software
- B1-K1-W4 : Test software

De werkprocessen zijn gekoppeld aan de leerdoelen. De volgende koppelingen zijn in deze reader opgenomen:

B1-K1-W3 :

- Ik pas in C# de juiste syntax toe
- Ik declareer en initialiseer variabelen in C#
- Ik voer in C# bewerkingen op strings uit
- Ik maak berekeningen met variabelen in C#
- Ik gebruik beslisstructuren in C#
- Ik lees inputvelden uit in C#
- Ik maak gebruik van herhalingslussen ('for' en 'while') in C#
- Ik converteer variabelen naar een ander type (indien nodig) in C#
- Ik zet bestaande Javascriptcode om naar C#

B1-K1-W4 :

- Ik debug C# code met behulp van breaktekens

Op de volgende pagina staat het gebruikte beoordelingsformulier afgedrukt.

BEOORDELINGSFORMULIER

Module					
Naam beoordelaar					
Naam student					
Datum					
Leerdoelen:	Beoordeeld bij opdrachten :	Punten:			
		O	M	V	G
Ik pas in C# de juiste syntax toe	2_1 Maanden van het jaar 2_2 Alfabetische volgorde 3_1 Vierkant met diagonalen 5_1 Aantal letters A, E of I 5_2 Woorden uit de zin 6 Eindopdracht				
Ik declareer en initialiseer variabelen in C#	1_1 Cilinder 2_1 Maanden van het jaar 2_2 Alfabetische volgorde 6 Eindopdracht				
Ik voer in C# bewerkingen op strings uit	5_1 Aantal letters A, E of I 5_2 Woorden uit de zin 6 Eindopdracht				
Ik maak berekeningen met variabelen in C#	1_1 Cilinder 3_2 kleinste getal, grootste getal en gemiddelde 4_1 Delen door				
Ik gebruik beslisstructuren in C#	2_1 Maanden van het jaar 2_2 Alfabetische volgorde 3_1 Vierkant met diagonalen 3_2 kleinste getal, grootste getal en gemiddelde 5_1 Aantal letters A, E of I 5_2 Woorden uit de zin 6 Eindopdracht				
Ik lees inputvelden uit in C#	1_1 Cilinder 2_1 Maanden van het jaar 2_2 Alfabetische volgorde 3_1 Vierkant met diagonalen 6 Eindopdracht				
Ik maak gebruik van herhalingslussen ('for' en 'while') in C#	3_1 Vierkant met diagonalen 3_2 kleinste getal, grootste getal en gemiddelde 5_1 Aantal letters A, E of I 5_2 Woorden uit de zin 6 Eindopdracht				
Ik converteer variabelen naar een ander type (indien nodig) in C#	1_1 Cilinder 2_1 Maanden van het jaar 2_2 Alfabetische volgorde 6 Eindopdracht				
Ik zet bestaande Javascriptcode om naar C#	4_3 Goedendag 4_4 Het nummer				

Ik debug C# code met behulp van breaktekens	4_1 Delen door 4_2 Kerstboom				

Opmerking: