

The image features a bright yellow background with a dark brown vertical bar on the left. In the center, there is a white, cloud-like shape. Inside this shape, the text 'C#' is written in a large, bold, dark brown font. Below the white shape, the word 'FUNDAMENTALS' is written in a very large, bold, dark brown font, spanning across the width of the image.

C#

FUNDAMENTALS

SESSIE 1 / 4

VOORSTELRONDE

- Naam
- Woonplaats
- Achtergrond studie/werkervaring
- Familie/partner/kinderen/huisdieren
- Hobby's
- Etc

PLAN DAG 1

- Wat is C#?
- Wat is .NET?
- Programmeren
- Eerste programma
- Variabelen
- Typen
- Primitieve typen vs objecten
- Enums
- Logische en vergelijkingsoperatoren
- If / else
- Switch
- While
- For

PLAN DAG 2

- Object georiënteerd programmeren
- Methoden
- Scope
- Encapsulation
- Toegangsbeperking
- Static / sealed / dynamic
- Inheritance
- Polymorphisme
- Klassen
- Abstracte klassen
- Interfaces
- Partial
- Assemblies

PLAN DAG 3

- Ruimte uitloop dag 2
- Constanten
- Arrays
- Collections
- Foreach
- In- en uitvoer van gegevens

PLAN DAG 4

- Debugging
- Exceptions
- Unittesting
- Eventuele uitloop
- Samenvatting
- Ruimte voor vragen en herhaling
- Casus die alles samenvat

WAT IS C#?

- General purpose programming language
- Object Oriented
- Ontwikkeld door Microsoft rond 2000

WAT IS .NET?

- Software framework
- In eerste instantie bedoeld voor Windows applicaties
- Nu breder toepasbaar met o.a. webapplicaties
- Framework kan aangesproken worden door C#, VB.NET, C++ en F#
- Ontwikkeld door Microsoft

Framework and core:

- <https://www.youtube.com/watch?v=OkeM7XVwEdA&feature=youtu.be>

QUIZ QUESTIONS

True or false:

- .NET werkt alleen samen met C#.
- Common Intermediate Language (CLI) maakt het mogelijk dat een VB component met een C# component kan communiceren
- Framework Class Libraries is the .NET virtual machine

QUIZ QUESTIONS

True or false:

- .NET werkt alleen samen met C#.
 - Common Intermediate Language (CIL) maakt het mogelijk dat een VB component met een C# component kan communiceren
 - Framework Class Libraries is the .NET virtual machine
- False
 - True
 - False, dat is the CLR: common language runtime

INSTALLEREN VISUAL STUDIO

- Visual C# express
- Of
- Visual Studio Community edition
- Of
- Betaalde versie van VS

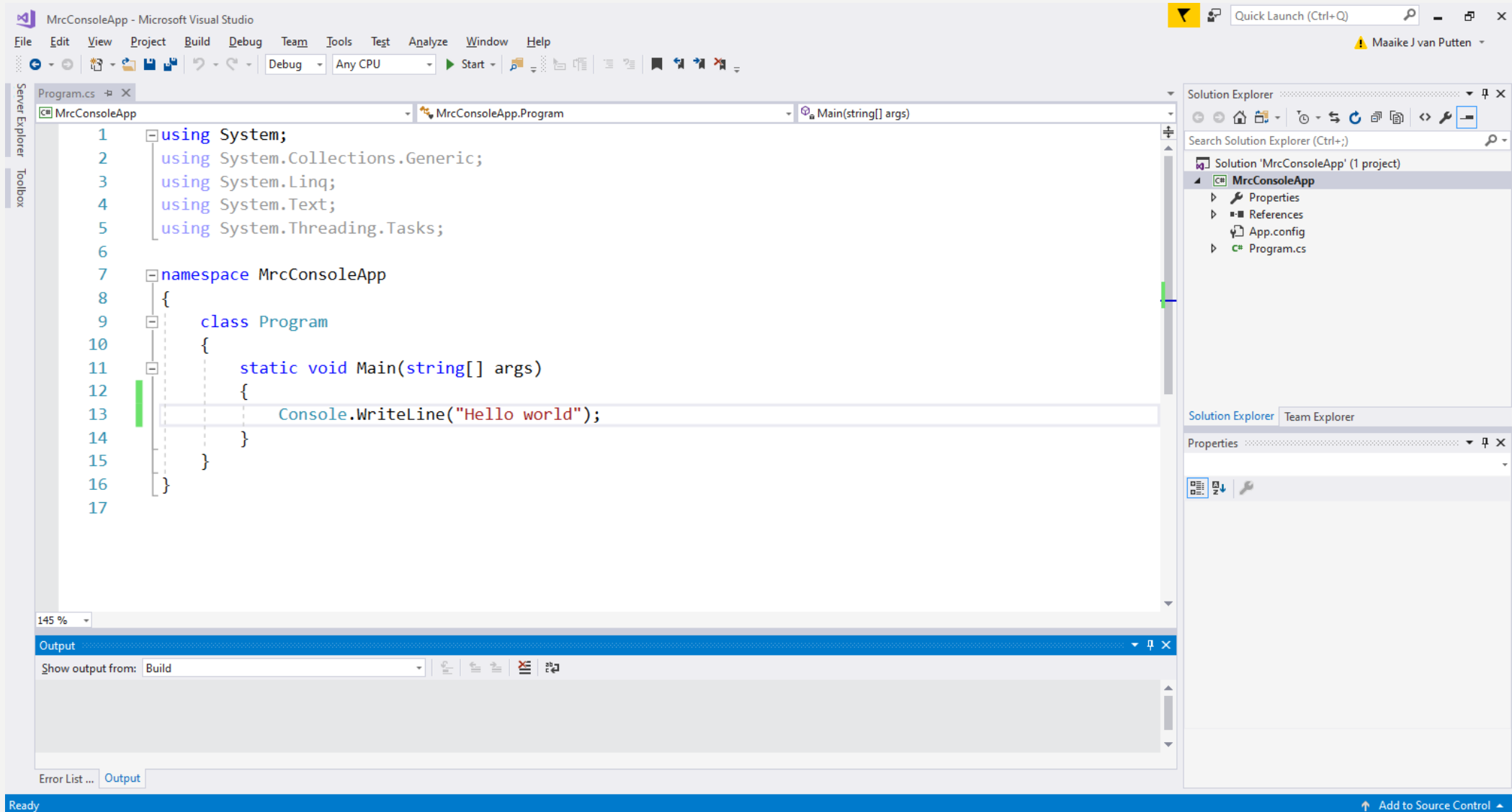
RONDELEIDING DOOR DE IDE

- Toolbox
- Code editor / designer
- Solution explorer
- Properties

PROGRAMMEREN

- Schrijven van instructies voor een computer om een bepaalde taak uit te voeren
- We schrijven code in de code editor
- Hulpmiddelen voor leesbare code:
 - IntelliSense, suggesties voor wat je wil typen
 - Zorg voor leesbare code:
 - Kies logische namen
 - Houd je aan de conventies voor namen
 - Spring je code voldoende in
 - Syntax hulp: dmv kleurtjes kun je zien of het goed gaat
 - Refactoring: aanpassen van code die al geschreven is. Bijv. een logischere naam geven.

HELLO WORLD!



OEFENING

- Maak je eigen hello world applicaties

NAMESPACES

- Een soort mapje voor code bestanden
- Wordt gebruikt om grote hoeveelheden code te structureren
- Staan bovenaan een file met “using” ervoor
- Namespaces kunnen in andere namespaces zitten. Daarmee wordt dan verwezen door middel van een punt ertussen
- Opper namespace is “global” (global::System)
- Namespaces beginnen met een hoofdletter

CONSOLE APPLICATION VS WINDOWS FORMS

- Console application
 - No interface
 - Run in cmd prompt
- Windows forms:
 - Applicatie met grafische gebruikersinterface
 - Standaard bouwblokken voor gebruikersinterface

VARIABLES

- Variabelen bestaan uit drie onderdelen:
 - Type
 - Naam
 - Inhoud
- Voorbeeld:
 - `int x = 5;`
 - `string name = "Maaike";`
 - `double getal = 5.6;`
- Declareren en initialiseren van variabelen
 - Declareren: type en naam geven.
 - Initialiseren: waarde toekennen.
- `string name = "Maaike";`
- `int number = 3;`

MEEST GEBRUIKTE TYPEN

Type	Description	Size (bytes)	.NET Type	Range
int	Whole numbers	4	System.Int32	-2,147,483,648 to 2,147,483,647
long	Whole numbers (bigger range)	8	System.Int64	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	Floating-point numbers	4	System.Single	+/-3.4 x 10 ³⁸
double	Double precision (more accurate) floating-point numbers	8	System.Double	+/-1.7 x 10 ³⁰⁸
decimal	Monetary values	16	System.Decimal	28 significant figures
char	Single character	2	System.Char	N/A
bool	Boolean	1	System.Boolean	True or false
DateTime	Moments in time	8	System.DateTime	0:00:00 on 01/01/0001 to 23:59:59 on 12/31/9999
string	Sequence of characters	2 per character	System.String	N/A

OEFENING

- Maak een variabele voor je naam, geef deze het juiste type en een logische naam
- Maak een variabele voor je leeftijd, geef deze het juiste type en een logische naam
- Maak een variabele voor je lievelingsgetal gedeeld door je minst favoriete getal, geef deze het juiste type en een logische naam

READLINE

- Met `Console.ReadLine()` pauzeert het programma en wacht het op input in de cmd line

OEFENING

- Ga terug naar je hallo wereld applicatie
- Zorg ervoor dat de gebruiker gevraagd wordt om een naam
- Daarna verschijnt: Hallo “ingevoerde naam”
- `Console.ReadLine()`

OBJECTS

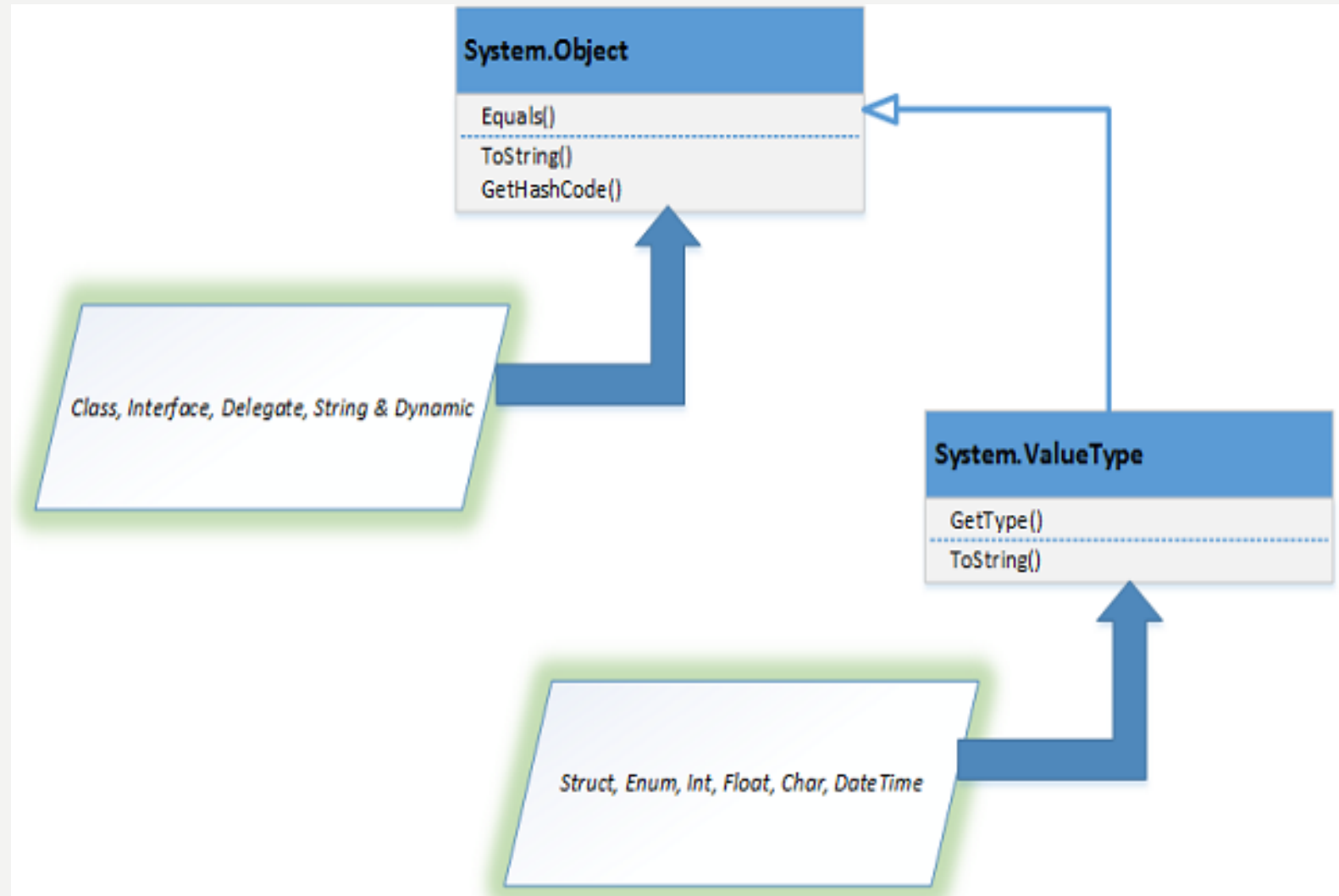
- Object references
- Verwijst naar een instantie van een klasse
- Object referenties (reference types) hebben geen waarde in het geheugen, maar alleen de geheugenlocatie van de instantie van de klasse
- Een referentie kan wijzen naar een object van het gespecificeerde type
- Een referentie kan gewijzigd worden naar een ander object van hetzelfde type
- Een referentie kan wijzen naar een nieuw object mbv new keyword

OBJECT REFERENCE

- References kunnen als waarde “null” hebben
 - `string name = null; //prima`
 - `int x = null; //compileert niet`
 - `int? x = null; //compileert wel, nu is het een nullable`
- References die niet naar null wijzen kunnen gebruikt worden om methodes/properties aan te roepen
 - `name.Length; //prima (ervan uitgaand dat het niet null meer is)`
 - `int y = 4;`
 - `y.Length; //compileert niet`
- Naamconventie: primitives lower case, class names upper case
- **Metafoor:**
 - Als je een appel hebt in de vorm van een object reference, dan heb je een sleutel van een kluisje. En in dat kluisje ligt de appel. Als de inhoud van het kluisje wijzigt, blijft de sleutel hetzelfde. Als je de sleutel wijzigt, dan wijst de sleutel naar een ander kluisje, waardoor een hele andere inhoud mogelijk is.

OBJECT REFERENCE VS VALUE TYPES

- <https://www.infoworld.com/article/3043992/a-deep-dive-value-and-reference-types-in-net.html>



VERANDEREN VAN DATA TYPE

- Met behulp van Convert.To..... kan het ene type omgezet worden in het andere.

- Bijv:

```
int x = 546;
```

```
string tekst = Convert.ToString(x);
```

OEFENING

- Bepaal het aantal posities (de lengte) van het getal 4567895 mbv C#

BOOLEANS

- True or false

```
bool myFirstBoolean = true;
```

LOGISCHE EN VERGELIJKINGSOPERATOREN

- < kleiner dan
- <= Kleiner dan of gelijk aan
- > groter dan
- >= groter dan of gelijk aan
- == gelijk
- != niet gelijk
- AND (&&)
- OR (||)

OEFENING

- Schrijf de logica voor het volgende scenario, true wanneer:
- Je jonger bent dan 3 jaar
- Of ouder bent dan 65
- Tussen de 31 en 41 bent én een museumjaarkaart hebt

- Schrijf de logica voor het volgende scenario, false wanneer:
- Het niet regent
- Het wel regent, maar ook hard waait
- Je niet een artistieke show gaat opvoeren waarvoor je een paraplu nodig hebt
- Bonus: verzin bij beiden de vraag

IF-STATEMENT

- Condities maken
- Als x, doe dit
- Anders doe dat

- Als x, doe dit
- Als y, doe zus
- Anders doe zo

```
if(x < 300)
{
    Console.WriteLine("x is kleiner dan 300");
}
else
{
    Console.WriteLine("x is groter dan 300");
}
```

IF-ELSE IF-ELSE STATEMENTS

- Soms ook meerdere mogelijkheden
- Bijvoorbeeld hiernaast:
 - Als x Kleiner is dan 300, dan...
 - Anders al x gelijk is aan 300, dan...
 - Anders....

```
if(x < 300)
{
    Console.WriteLine("x is kleiner dan 300");
} else if (x == 300)
{
    Console.WriteLine("x is 300");
}
else
{
    Console.WriteLine("x is groter dan 300");
}
```


OEFENING IF STATEMENTS

- Bijvoorbeeld toegangsprijs museum
- Gebruik readline om een leeftijd te krijgen
- Bepaal de toegangsprijs (WriteLine result)
- - als jonger dan 3 dan gratis
- - als tussen 3 en 18 dan 10,-
- - als ouder dan 18 dan 15,-

```
if(x < 300)
{
    Console.WriteLine("x is kleiner dan 300");
} else if (x == 300)
{
    Console.WriteLine("x is 300");
}
else
{
    Console.WriteLine("x is groter dan 300");
}
```

OEFENING: VERKEERSLICHT

- Schrijf in if statements de logica die bij een stoplicht in je hoofd gebeurt.
- Zoals: rood, stoppen
- En: oranje en dichtbij >> rijden, oranje en ver weg >> stoppen/gas los

ENUM

- Instead of class, we use enum to define an enum ;)
- Starts with capital letter, same as class.
- Basic enum has comma separated values.
- Enums cannot start with numbers
- Enum values are best written UPPERCASE

2 references

```
enum WeekDay
{
    MAANDAG,
    DINSDAG,
    WOENSDAG,
    DONDERDAG,
    VRIJDAG,
    ZATERDAG,
    ZONDAG
}
```

ENUM – BASIC USAGE

Weekdag dag;

dag = Weekdag.WOENSDAG;

Wat denk je dat er gebeurt als:

int dagnummer = (int)dag;

EXERCISE: VERKEERSLICHT

- Maak een enum voor de waarden van een verkeerslicht (ROOD, GROEN, ORANJE)
- Zorg dat er ook een waarde is voor buiten gebruik
- Gebruik de waarde van de enum in eerder gemaakte if / else statement

SWITCH STATEMENT

- Decision making with a single variable in expression.
- Value of case must be same variable type as expression.
- When case value is equal to expression value, code after the case
- declaration will be executed UNTIL a break statement is reached.
- Optional: default case
- If none of the cases are true, the default is executed.

```
int day = 4;
switch (day)
{
    case 1:
        Console.WriteLine("Monday");
        break;
    case 2:
        Console.WriteLine("Tuesday");
        break;
    case 3:
        Console.WriteLine("Wednesday");
        break;
    case 4:
        Console.WriteLine("Thursday");
        break;
    case 5:
        Console.WriteLine("Friday");
        break;
    case 6:
        Console.WriteLine("Saturday");
        break;
    case 7:
        Console.WriteLine("Sunday");
        break;
    default:
        Console.WriteLine("Huh?");
        break;
}
```

EXERCISE: VERKEERSLICHT

- Schrijf een switch statement voor wat je doet bij rood, groen, oranje en defect stoplicht.

WHILE LOOP

- Lijkt op herhalende if-statement

```
while(x < 300)
{
    Console.WriteLine(x);
    x++;
}
```


OEFENING

- Zolang er ongelezen email is, ga je de volgende email lezen
- Kan 1 email zijn
- Kunnen er ook 100 zijn

DO...WHILE

- Bijna zelfde als while
- Doe het eerst 1 keer en daarna zolang de conditie geldig is

```
do
{
    Console.WriteLine("geef een getal groter dan of gelijk aan 300 op");
    x = Convert.ToInt32(Console.ReadLine());
} while (x < 300);
```

OEFENING

- Vraag de user om te kiezen tussen een appel of een peer
- Blijf dit vragen zolang de user iets anders kiest dan appel of peer

FOR LOOP - INTRODUCTIE

- For loop kent 2 varianten, we doen vandaag alleen de eerste:
- For met teller:

```
for ( teller ; conditie ; na afloop) {  
    // Do something  
}
```

FOR LOOP - TELLER

- Wanneer te gebruiken: Je weet vaak vooraf hoe vaak je moet lopen

- 1) Zet $i = 0$
- 2) $i < 10 = \text{true}$
- 3) Voer code uit
- 4) $i++$ (betekent $i = i + 1$)
- 5) $i < 10 = \text{true}$
- 6) Voer code uit
- 7) $i++$
- 8)
- 9) $i < 10 = \text{false}$
- 10) Stop

OEFENING

- Maak een for loop die voor iedere verjaardag die je gehad hebt print:
- En toen werd je x jaar!

OEFENING

- **Schrijf de code (zo kort mogelijk) om dit liedje in het console uit te printen**
- 10 Little monkeys jumping on a bed, one jumped up and bumped his head,
Mom called the Doctor and the doctor said....."no more monkeys jumping on a bed."
- 9 Little monkeys jumping on a bed, one jumped up and bumped his head.
Mom called the Doctor and the Doctor said....."no more monkeys jumping on a bed."
-
- 1 Little **monkey** jumping on a bed, **he** jumped up and bumped his head.
Mom called the Doctor and the Doctor said....."no more monkeys jumping on a bed."
- Bonus: maak de 10 een variable

Bonus: gebruik verschillende soorten loops om hetzelfde te bereiken!

BREAK / CONTINUE

Break:

- Je kunt uit een loop met behulp van break.

```
int x = 0;
while (true)
{
    x++;
    if(x % 2 == 0)
    {
        continue; //back to while()
    }
    Console.WriteLine("waarde van x " + x);

    if(x > 150)
    {
        break; //uit de loop
    }
}
```

Continue:

- Met behulp van continue kun je de huidige iteratie van de loop vroegtijdig beëindigen en doorgaan met de volgende

OEFENING

- Maak een while loop die aan de user vraagt om appel of peer
- Break zodra appel of peer is ingevoerd

- Maak een for loop die telt tot 100
- En enkel de even getallen print mbv continue

WRAP UP OEFENING

- We gaan een spelletje maken!
- De code neemt een getal in het hoofd tussen 0 en 500
- De gebruiker moet het getal raden
- Op de command wordt getoond of het getal hoger of lager moet zijn
- De gebruiker kan blijven raden totdat het juiste antwoord gegeven is

Tip:

- `Random r = new Random();`
- `int secret = r.Next(1,500);`

- Dan krijgt de gebruiker nog de volgende feedback:
 - Minder dan 5x: snel gedaan!
 - Tussen de 5 en 10x: netjes
 - Tussen de 10 en 15x: zo zo
 - Meer dan 15x: het duurde wel lang...
- Bonus: maak een enum van 4 bovenstaande opties en gebruik een switch statement

WRAP UP DAY 1

- Wat is C#?
- Wat is .NET?
- Programmeren
- Eerste programma
- Variabelen
- Typen
- Primitieve typen vs objecten
- Enums
- Logische en vergelijkingsoperatoren
- If / else
- Switch
- While
- For

NOG VRAGEN?

Stel ze via:

maaike.vanputten@brightboost.nl

Of WhatsApp:

0683982426