



# Informatica Werktuigen

## Visual Basic .NET

# Doel van de les vandaag

## **Niet:**

Jullie in een kleine les Visual Basic .NET leren

## **Wel:**

Jullie een gevoel geven van wat Visual Basic .NET is

*Een programmeertaal leer je door ermee te werken!*

# Wat is Visual Basic .NET?

- Een **programmeertaal** die
  - Objectgeoriënteerd is (net als Python)
  - **Event-driven** is
  - Op **.NET framework** is gebaseerd
- Wordt gebruikt om snel programma's te ontwikkelen met een grafische interface
  - Vaak ten koste van performantie en uitbreidbaarheid
  - Vooral voor kleinere applicaties en prototypes

# Sequentieel Programmeren

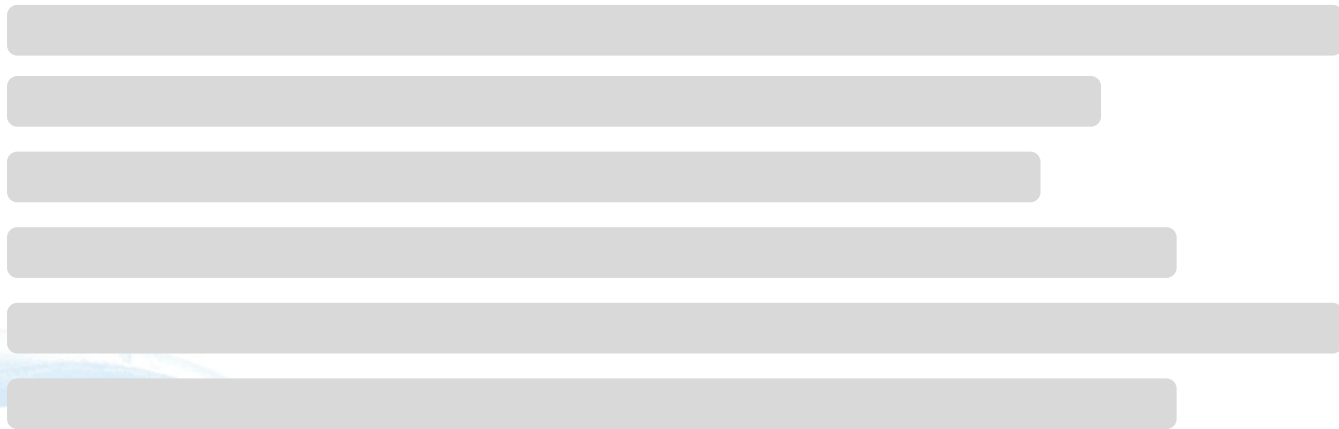
*Sequentieel* programmeren:

- Sturing van programma door een sequentiële reeks instructies
- Deze volgorde kan beïnvloed worden door de invoer



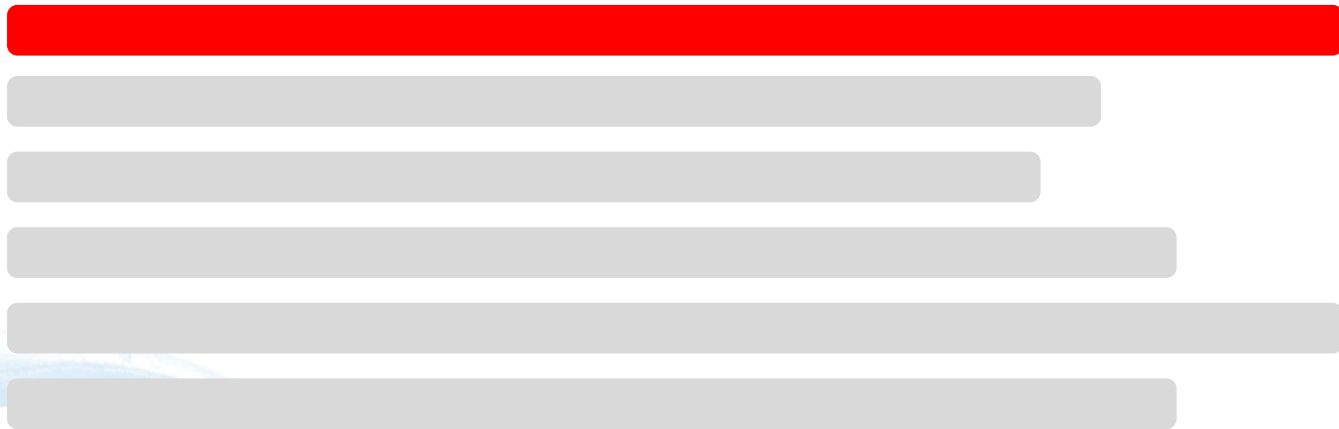
# Sequentieel Programmeren

## ***Sequentieel*** programmeren



# Sequentieel Programmeren

## *Sequentieel* programmeren



# Sequentieel Programmeren

## *Sequentieel* programmeren



# Sequentieel Programmeren

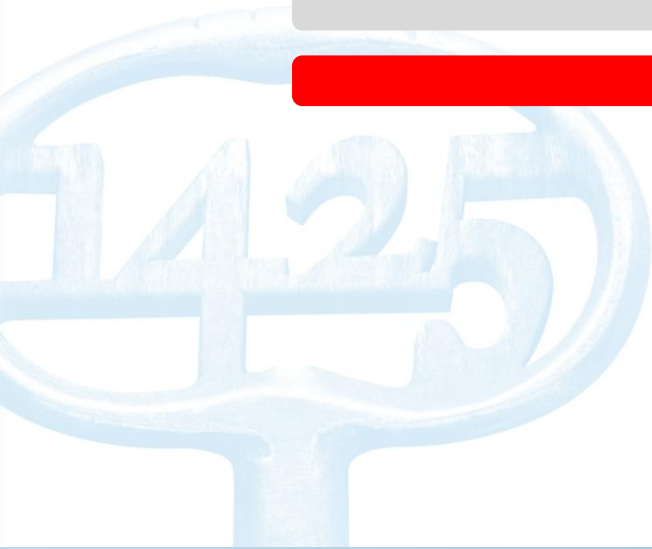
## ***Sequentieel*** programmeren





# Sequentieel Programmeren

## ***Sequentieel*** programmeren



# Sequentieel Programmeren

## ***Sequentieel*** programmeren



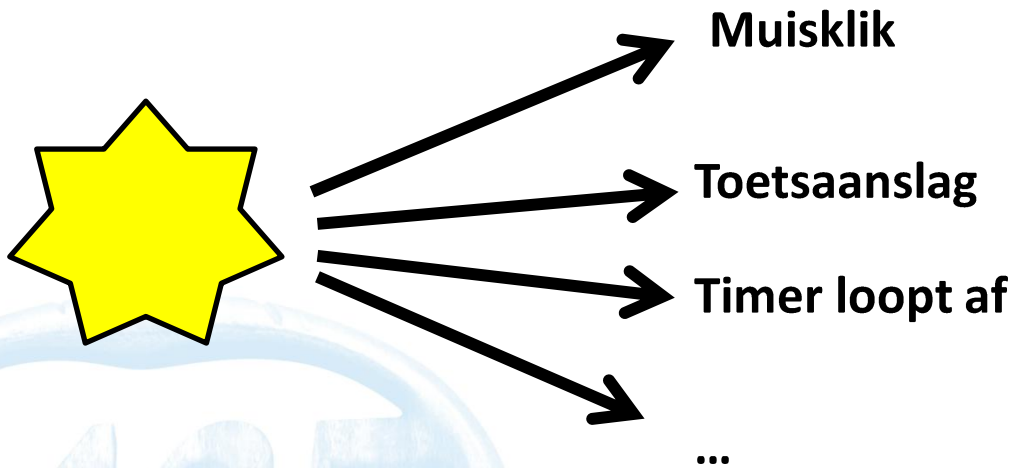
# Event-Driven Programmeren

## *Event-Driven* programmeren

- Centraal concept: de gebeurtenis
- De volgorde van uitvoering wordt bepaald door het voorkomen van events of **gebeurtenissen**
  - Bvb. Het klikken op een knop
  - Bvb. Het aflopen van een timer
- Bij elke gebeurtenis kan een **actie** worden geassocieerd

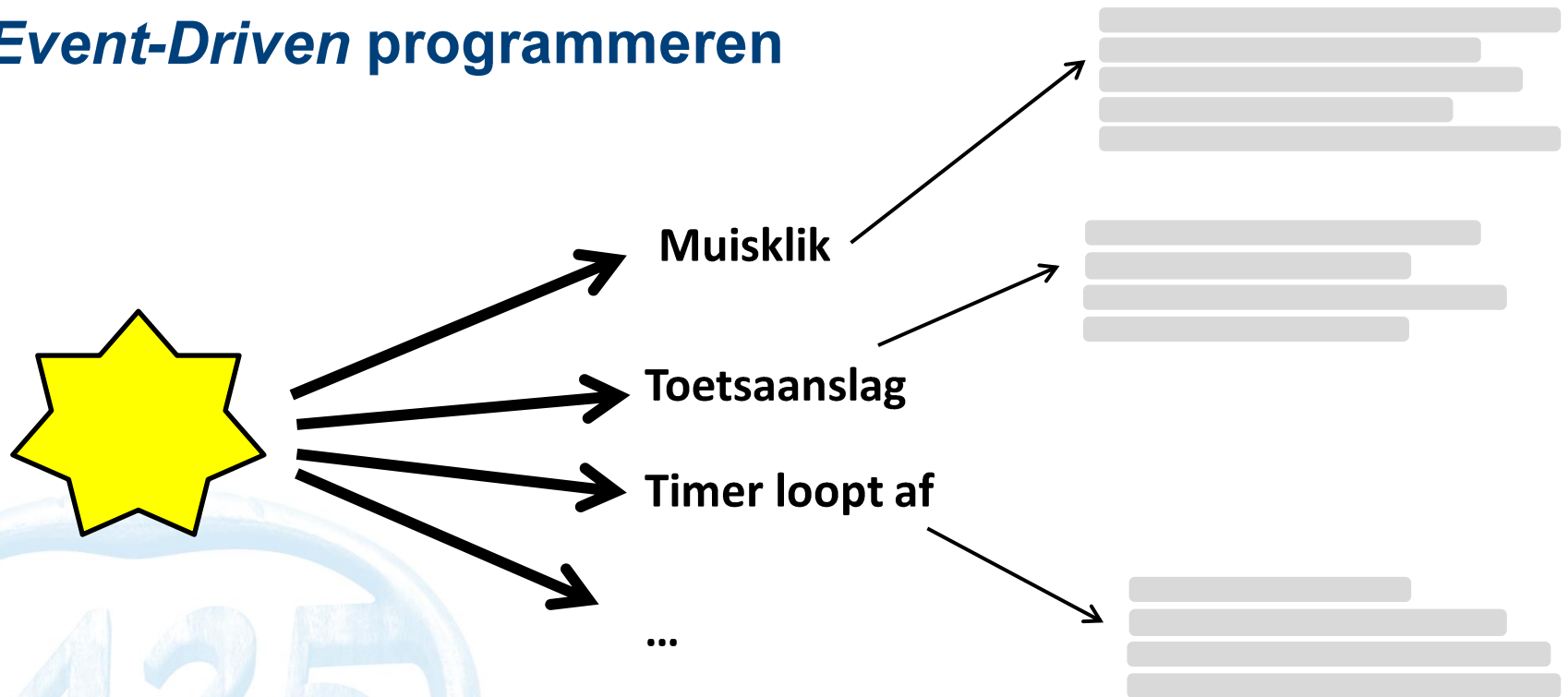
# Event-Driven Programmeren

## *Event-Driven* programmeren



# Event-Driven Programmeren

## *Event-Driven* programmeren



# Wat is het .NET framework?

Een softwareplatform dat allerlei functies, klassen en hulpmiddelen aanbiedt om het leven van de softwareontwikkelaar makkelijker te maken



# Wat is het .NET framework?

Biedt de ontwikkelaar:

- Een **softwarebibliotheek**, die het ontwikkelen van een gebruikersinterface, communiceren met een database en ontwikkeling van websites vereenvoudigt (en veel meer)
- Programma's laat uitvoeren in een zgn. *Common Language Runtime*-omgeving, een **virtuele machine** (cfr. Java) die onder andere geheugenbeheer en foutafhandeling verzorgt

# Praktisch





# In VB .NET programmeren

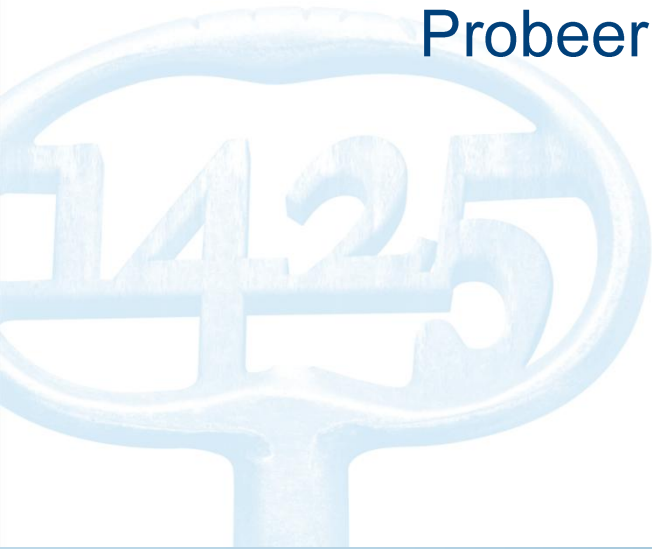
- Gebruik maken van Visual Studio ontwikkelomgeving
  - Snelle ontwikkeling van grafische interface
  - Broncode binden aan de grafische interface
- Voor jullie twee belangrijke versies:
  - Express versie
  - Professionele versie
- Let op: Editie 2013 is al beschikbaar, maar in computerlokalen word nog editie 2012 gebruikt!
  - Gebruik ook de **2012** editie om je practicum in te ontwikkelen!

# Visual Studio installeren

- Wil je het installeren op je eigen computer, volg dan de instructies op Toledo (volgen later)
  - Kan enkel op Windows
- Visual studio .NET staat geïnstalleerd in de computerklassen van gebouw 200C
  - Tijdens normale uren kan je hier terecht als er hier geen oefenzittingen worden gegeven
  - Hier kan je dus e.v.t. ook je practicum maken
  - Ook in het centrum (Dekenstraat) kan je hiervoor terecht

# Opmerking

- Meestal wordt Visual Basic .NET ook gewoon **Visual Basic** genoemd
  - Maar: bij opzoeken op Internet kan je soms code uit Visual Basic 6 of eerder terugvinden
  - Deze is niet volledig compatibel met .NET code
  - Kan dus zijn dat er compilatiefouten zijn  
Probeer eens een andere bron



# Inleiding tot de programmeertaal



# Demonstratie 1 – Hallo wereld



# Demonstratie 1 – Hallo wereld

- Form of formulier is het werkscherm van de grafische interface (GUI)
- Controls:
  - Elementen in het formulier
    - Knoppen, Lijsten, Labels, Textboxes, ...
  - Hebben allemaal een reeks properties of eigenschappen
    - Sommige zijn voor alle Controls geldig
      - Naam, Positie binnen het formulier, ...
    - Andere zijn specifiek bedoeld voor een type Control
      - Bijvoorbeeld vooruitgang bij een ProgressBar
- Op deze Controls gebeuren Events (gebeurtenissen)
  - Bvb Knoppen: Click, TextBoxes: OnTextChanged, ...

# Typering

- In Python is er geen expliciete typering van variabelen
  - “*Dynamic typing*”
- Visual Basic .NET vereist dit wel:
  - We moeten aangeven van welk **type** een variabele zal zijn en ons hieraan houden!
  - Dit is veiliger, maar restrictiever dan Python
  - Allerhande datatypes ondersteund:
    - Getallen, kommagetallen, strings, karakters, booleans, ...

# Demonstratie 2 – Een tweede versie





# Declaratie van variabelen

- Door middel van het **Dim** sleutelwoord
  - Bijvoorbeeld **Dim leeftijd As Integer**
- Het datatype wordt aangegeven na **As**, veel voorkomende types zijn
  - Integer (getallen)
  - Double (kommagetallen)
  - String (tekstuele data)
  - Boolean (Waar / Onwaar waarde)
- Een **Lijst** (Array) wordt gedeclareerd door de lengte na de variabelenaam te plaatsen
  - Bijvoorbeeld **Dim leeftijd(10) As Integer**
  - Een matrix als **Dim matrix(5,5) As Integer**

# Algemene opmerkingen

- Wanneer we een element uit een lijst opvragen, gebruiken we ronde haakjes ( ) met de index, **geen** vierkante (Python)
- Bijvoorbeeld **huidigeLeeftijd = leeftijd(2)**
- Commentaar begint met ' en wordt in Visual Studio standaard **groen** gekleurd
  - Commentaar over verschillende regels (zoals **''''''** **''''''** in Python) bestaat niet in Visual Basic .NET
- VB .NET laat maar één instructie per regel toe

# Toekenning en Operatoren

- Toekenning

=

- Operatoren

- op **Integers**

+, -, \*, \ (gehele deling), **Mod** (rest, zoals % in Python)

- op **Booleans**

And, Or, Not

- op **Strings**

- **&** (concatenatie van twee strings)

Bijvoorbeeld: **mijnTekst = "Hallo" & " Wereld"**

- Vergelijkingsoperatoren:

<

>

<=

>=

<>

=

# Toekenning en Operatoren

- Toekenning

=

- Operatoren

- op **Integers**

+, -, \*, \ (gehele deling), **Mod** (rest, zoals % in Java)

- op **Booleans**

And, Or, Not

- op **Strings**

- **&** (concatenatie van twee strings)

Bijvoorbeeld:

**mijnTekst = "Hallo" & " Wereld"**

- Vergelijkingsoperatoren:

<

>

<=

>=

<>

=

!!

# Controlestructure: If Else

## Visual Basic .NET

- If invoer > 10 Then  
    MessageBox.Show("Te groot!")  
End If
- If invoer > 10 Then  
    MessageBox.Show("Te groot!")  
Else  
    MessageBox.Show("OK")  
End If

## Python

- if invoer > 10:  
    print 'Te groot!'
- if invoer > 10:  
    print 'Te groot!'  
else:  
    print "OK"

# Controlestructuren: Lussen

## Visual Basic .NET

- While `teller < 3`  
    `Console.WriteLine(teller)`  
    `teller = teller + 1`  
End While

## Python

- while `teller < 3`:  
    `print teller`  
    `teller = teller + 1`

### Uitvoer:

0  
1  
2

# Controlestructuren: Lussen (2)

## Visual Basic .NET

- For **teller = 0** To **2** Step **1**  
    **Console.WriteLine(teller)**  
Next

## Python

- for **teller** in range(**0,3,1**):  
    **print teller**

## Uitvoer:

0  
1  
2

# Functies en procedures

- In Visual Basic .NET is er een expliciet syntactisch verschil gemaakt tussen functies en procedures
  - Procedures worden aangegeven met het sleutelwoord **Sub** en **geven geen resultaat terug**

Bijvoorbeeld

```
Sub SchrijfUit (aantal As Integer, tekst As String)
    System.Console.WriteLine(aantal.ToString & tekst)
End Sub
```

Uitvoer voor oproep SchrijfUit(2, “ programmeertalen”):  
2 programmeertalen



# Functies en procedures (2)

- In Visual Basic .NET is er een expliciet syntactisch verschil gemaakt tussen functies en procedures
- Functies worden aangegeven met het sleutelwoord **Function** en geven **wél een resultaat** terug

Bijvoorbeeld

```
Function GeefTekst (aantal As Integer, tekst As String) As String
```

```
    Return aantal.ToString & tekst
```

```
End Function
```

# Demonstratie 3 – Deelbaar door 3

- Gegeven een getal A, geef alle getallen kleiner dan A die deelbaar zijn door 3 terug
- Bijvoorbeeld
  - Invoer: 16
  - Uitvoer: 3 6 9 12 15
- In een tekstvak, met een Toon-knop en Wis-knop
  - Beide onbeschikbaar bij lege invoer



# Event-driven programmeren

- Om event-driven te programmeren, moet je aan een gebeurtenis een reeks acties kunnen binnen
  - In Visual Basic .NET wordt dit aangegeven door middel van de **Handles** instructie
  - Deze wordt gebonden aan een control (bijv. Een knop, een timer, een formulier, ...) door middel van de naam die je eraan gegeven hebt (kan bij de eigenschappen worden aangepast!)
  - Acties worden gegroepeerd in een subroutine

# Event-driven programmeren

**Private Sub** KlikAfhandeling(sender **As Object**, e **As EventArgs**) **Handles** knop.Click  
    ' Voer hier de acties uit die er moeten gebeuren  
**End Sub**



# Omgaan met tijd

## Timer:

- Zal, wanneer gestart, om de zoveel tijd een *Tick* gebeurtenis aan het .NET framework geven
- Deze gebeurtenis kan net als andere gebeurtenissen worden afgehandeld door acties uit te voeren



# Demonstratie 4 – Gebruik van een Timer



# Timer

- Gebruiken:
  - Trek de Timer uit de Toolbox naar het formulier
  - Verander de Interval-eigenschap in de tijd die je wil dat de timer afloopt
    - Opgelet: Na deze tijd begint de Timer opnieuw
  - Start de Timer door de Enabled-eigenschap op True te zetten
    - Stoppen analoog: waarde op False zetten

# Debugging





# Debugging

- Visual Studio biedt ondersteuning om op een makkelijke manier te **debuggen**
  - Debuggen = opsporen van softwarefouten
  - Wij zien vandaag slechts twee werktuigen hiervoor
  - De tweede oefenzitting zal hier dieper op ingaan



# Debugging – Debug.WriteLine

- Soms is het handig om de waarden van variabelen op een bepaald moment van de uitvoer uit te printen
  - Ook: rapporteren of we op een bepaald deel in de code terechtkomen (bijvoorbeeld na een lus – eindigt deze?)
- Het is echter niet de bedoeling dat de eindgebruiker dit te zien krijgt
- Debug printing:

`Debug.WriteLine("Hier in de code is de waarde " & variabele & "!!")`

- Enkel zichtbaar in ontwikkelmodus

# Debugging – Breakpoints

- WriteLine kan code zeer chaotisch maken
- Om dit te voorkomen kan je werken met **breakpoints**
  - Vereenvoudigt het zoeken naar fouten in de code



# Debugging – Breakpoints

Een **breakpoint** is een locatie in de code waar je, terwijl je het programma aan het testen bent, wil dat de uitvoer van het programma even pauzeert zodat je de waarden van alle variabelen kan bekijken



# Demonstratie - Breakpoint



# Praktische afspraken



# Praktisch

- Oefenzittingen
  - 2 Oefenzittingen om jullie te laten kennismaken met Visual Basic .NET
    - De eerste is een kennismaking met de programmeertaal
    - De tweede spitst zich toe op debugging
  - Vanaf volgende week in lokalen in 200C
  - Verplicht!



# Praktisch

- Practicum
  - Zal op Toledo beschikbaar worden ten laatste vrijdag 24 oktober
  - Deadline maandag 3 november zijn
  - Komt ook een individuele verdediging van
  - Lees de opgave **goed** na voor je begint!
  - Begin op tijd!





# Vragen?

