# .Net Essentials: C#

## Code conventies:

Gebruik bij Methods/Klassenamen: Pascal casing.

Graphical user interface, text

Description automatically generated with medium confidence

Gebruik bij variables en parameters + interfaces: Camel casing.

Text

Description automatically generated with medium confidence

## Hoofdstuk 1: Achtegrond

### Wat is een programma?

Logisch stappenplan (algoritme)

Bestaat uit:

* Opeenvolgende statements
* Herhalingen (loops)
* Selecties (if)

## Hoofdstuk 2: C#-Ontwikkelomgeving

IDE = Integrated Development Enviroment

### Creating a new project (in visual studio)

* Gebruik maken van WPF App (.NET framework)
* Project een juiste naam geven en opslaan op een vaste plaats.

### Werken binnen een project/solution:

* XAML
  + Toolbox (Common WPF controls):
    - Bevat alle design/ontwerp onderdelen van het programma zoals buttons etc.
    - Belangrijkste deel van het hele programma (in my opinion)
    - Ontwerp onderdelen worden geplaatst in de XAML  Ontwerpvenster
    - Al deze info zie je dan terugkomen in de window onderaan (gecodeerd)
  + Properties:
    - Aanpassen van de items uit de toolbox
    - Bliksempje == Eventhandler
  + Messagebox:
    - MessageBox.Show(“inhoud”);
  + Belangrijk om te onthouden:
    - ; na elke statement
    - Blokken: {…}
    - Vergeet geen name = “” in je XAML toe te voegen aan elk element.
    - Hogere versies van .NET zullen mogelijks niet draaien lagere versies.
    - Commentaar = //tekst
* XAML.cs
  + Hierin word de background-code geschreven.
  + Alle classes,loops en if statements worden hierin geschreven. (Verder info volgt).

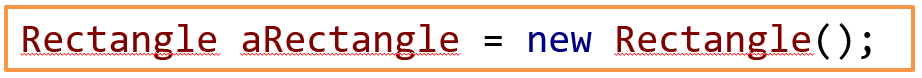
### Help (lokaal gebruik):

1. Help bovenaan de balk.
2. Add and Remove help content
3. Zoeken!

## Hoofdstuk 3: Objecten en methoden voor eenvoudige graphics

### Aanmaken van objecten:

Zoe goed als hetzelfde als java, het object is een instantie van een klasse.



Hiervan kunnen methodes worden opgeroepen en properties op kunnen worden gebruikt.



### Canvas:

Voor men een object kan laten tekenen moet men een canvas aanmaken.

Toolbox 🡪 Canvast 🡪 XAML canvas aanmaken (Naam niet vergeten!!)

Coordinaten worden gemeten vanaf linker bovenhoek canvas.

### Verschillende klasse-vormen:

Er bestaan al enkele vorm-klasse.

Bijvoorbeeld:

* Rectangle
* Ellipse
* Line
* …

### Aanmaken van een vorm:

* Width
* Height
* stroke 🡺 new SolidColorBrush(Colors.x) x == kleur die je wilt.
* Margin 🡺 new Thickness(x,x,0,0); (links, top, rechts,bottom) regelt de positie7
* Name

## Hoofdstuk 4: Variabelen en berekeningen

### Declareren van variabelen:

* Zinvolle namen
* Beginnen met letter of \_
* Geen spaties/symbolen of keywords
* camelCasing! (Zie vorig hoofdstuk)

Grotendeels hetzelfde als Java/python.

**Belangrijk!**

Int i = 3 + 3;

i krijgt de waarde 6, dit betekend dus niet i is gelijk aan 6.

#### Werken met strings en getallen:

Als men strings en getallen samen wilt voegen gaat dit niet uitkomen zoals bij python/java.

Calendar

Description automatically generated with medium confidence

### Conversiefuncties:

Omzetten van variabele naar een variabele van een ander soort.

Diagram

Description automatically generated

### String formatting - Getallen formatteren in een string:

$”Dit is een tekst met een eerste variabele {var1} en een tweede {var2}”

Dit mogen zowel getallen als andere dingen voorstellen.

### Nieuwe componenten:

* TextBox: invoer gebruiker met als property tekst
* Label: Niet wijzigbaar door gebruiker, Tekstuele informatie met als property content.
* TextBlock: property tekst, niet wijzigbaar door gebruiker.

### Getallen afkorten etc.:

* {expr:c}: omzetten naar een munteenheid
* {expr:d}: omzetten naar een geheel getal
* {expr:0.000}: een getal afronden/aanvullen met nullen tot 3 cijfers na de comma
* {expr:0.###}: een getal met maximaal 2 cijfers na de comma
* {expr:t}: korte tijdsnotatie

### Formateren van getallen:

Table

Description automatically generated with medium confidence

### Inputs:

Voor een input gaat men een textBox gebruiken en geen textblock!

## Hoofdstuk 5: Methoden en argumenten

Worden geschreven in de XAML.cs file

### Methodes:

Methodes werken basically hetzelfde als in java. Het nut hiervan is herhaling te vermijden.

Diagram

Description automatically generated

🡺Bekijk de oefening van het huisje.

#### Regels bij het maken van methodes:

* Kiest de parameters
* Type parameters
* Betekenisvolle namen

#### Gebruiken van een methode:

* Geeft de argumenten mee

### Klasses:

Bevatten x aantal methodes. Bv. MainWindow

Table

Description automatically generated

## Hoofdstuk 6: Objects

Er zijn een heel aantal pre-made objecten. Deze objecten worrden gemaakt in klasses.

Text

Description automatically generatedText

Description automatically generatedWerkt zo goed als hetzelfde als in java.

### Het random object:

De max-value is exclusief dus je moet altijd je max +1 doen.

### Sliders:

Content="{Binding ElementName=speedSlider, Path=Value, UpdateSourceTrigger=PropertyChanged}"

Zorgt ervoor dat de conteng van je label automatisch geupdate wordt.

### Text Description automatically generatedDe klasse dispatcherTimer:

Voorbeeld: klok

Eventhandler zorgt ervoor dat alles wat binnen de eventhandler staat om de ‘interval’ tijd wordt uitgevoerd.

## Hoofdstuk 7: Beslissingen (if-switch)

### If statements in C#:

Text

Description automatically generatedHetzelfde als in java.

}

* Else statements werken hetzelfde
* Operators werken hetzelfde
* Else if werkt hetzelfde

### Werking van een switch:

Diagram

Description automatically generated with low confidenceWerkt als volgt. Als dagnummer gelijk is aan “case optie1” dan word het volgende uitgevoert.

In dit geval staan de nummers voor mogelijke uitkomsten.

LET OP!

Je moet break gebruiken in een switch.

A picture containing text

Description automatically generatedHet is alsook aangeraden om een “default” te maken voor moest er een ongeldige waarden zijn ingegeven voor daynumber.

### Meerdere buttons die 1 event hebben:

Let hier op de ((button)sender)

Text

Description automatically generatedDeze neemt de content van die specifieke button.

## Hoofdstuk 8: Herhalingen (for-while)

### For loops:

Werkt hetzelfde als in java.

Text

Description automatically generated with low confidence

Belangrijk!

De lusvariabele (number) zou nooit mogen verandert worden in de lus zelf.

### While loop:

Werkt hetzelfde als in java.

Text

Description automatically generated

### Do while loop:

Werkt hetzelfde als een while loop enkel anders geschreven.

Graphical user interface, text

Description automatically generated with medium confidence

## Hoofdstuk 11: Overervingen

Werkt hetzelfde als in java.

Methodes kunnen overgenomen worden.

## Hoofdstuk 12: Brekeningen

### Wiskundige functies:

Table

Description automatically generated with low confidence

## Hoofdstuk 13: Gegevenssctructuren

### Control: Listbox

Toont een lijst met elementen. (gemakkelijk toevoegen/verwijderen van elementen in die lijst.

Graphical user interface, application

Description automatically generated

### Datastructure: List

Arraylist maken voor de objecten.

Hieraan de objecten toevoegen met “.add()”.

De lijst toevoegen aan een listbox 🡺

#### Problemen met listbox?

if (itemListBox.SelectedIndex == -1) return;

### Combobox:

Drop down lijst (Beetje een mix tussen TextBox en ListBox.

A picture containing shape

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

## Hoofdstuk 14: Arrays

Een array word gezien als een OBJECT dat andere objecten groepeert.

### Array creëren:

Graphical user interface, application

Description automatically generated

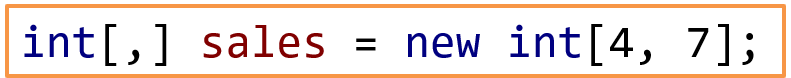


Belangrijk!

Arrays starten van 0.

## Hoofdstuk 15: Tweedimensionale arrays

### Tweedimensionale array declareren:

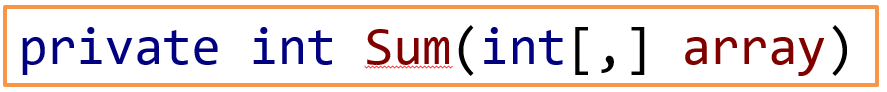


A picture containing text

Description automatically generated

* 4 = Rij-nummers (verticaal)
* 7 = Kolomnummers (horizontaal)

### Arrays als parameters:



### Grootte van een array opvragen:

* Arraynaam.GetLength(x)
  + X = 0 🡺 Aantal rijen
  + X = 1 🡺 Aantal kolommen
* Arraynaam.length 🡺 totaal aantal elementen

## Hoofdstuk 16: Bewerkingen met strings

### Diagram Description automatically generatedStandaard bewerkingen met strings:

Table

Description automatically generated with low confidence

Text

Description automatically generated with medium confidence

### Stringbuilder:

Zoals met java

## Hoofdstuk 17: Exceptions

Exception 🡪 er is een fout opgetreden

**Runtime-fouten:**

* Niet gedetecteerd door de compiler
* Onmogelijke operaties

**Logische fouten:**

* Oneindige lussen
* Programma produceert foute resultaten

### Try-Catch:

Als er in het programma een fout kan optreden.

Try{

Code die kan falen

}

Catch(exception){ code die uitgevoerd word als het faalt}

### Opgooien van een FormatException:

Throw new FormatException(“tekstje die je wilt weergeven bij de fout”);

### Finally:

Een stuk code dat altijd word uitgevoerd.

### Voorwaardelijke catch:

Deze catch word enkel uitgevoerd als iets anders voldoet aan een bepaalde waarde.

A picture containing text

Description automatically generated

### Exception classe:

public ExceptionName(string message) : base(message) { }

Text

Description automatically generated

## Hoofdstuk 18: Bestanden

System.IO

### Het aanmaken van een bestand:

Text

Description automatically generatedWriteline() sluit altijd af met een NewLine teken.

#### Schrijven naar een bepaalde locatie:

String folderpad = Environment.GetFolderPath(Enivronment.SpecialFolder.MyDocuments);

String filePath = System.IO.Path.Combine(folderpad,”naamvanjetekst.txt”);

StreamWriter writer = File.CreateText(filePath);

### Uitlezen van een bestand:

Text

Description automatically generatedLeest alle lijnen uit tot hij een lege lijn tegenkomt.

### OpenFileDialog:

Verkenner openen. 🡺 System.windows.forms;

Text

Description automatically generatedNormaal gezien moet je gewoon:

Text

Description automatically generatedDialog.showDialog(); doen om de verkenner te openen. En dialog.filename geeft je de geselecteerde file terug.

Graphical user interface, text, application, website

Description automatically generated

### GUI menu creëren:

Text, timeline

Description automatically generated

<Menu DockPanel.Dock="Top" Background="#FFDE665B">

<MenuItem Header="File" Foreground="White">

<MenuItem Header="\_Open" Name="openItem" Click="openItem\_Click" Foreground="Black"/>

<MenuItem Header="\_Save" Name="saveItem" IsEnabled="False" Foreground="Black" Click="saveItem\_Click"/>

<Separator/>

<MenuItem Header="E\_xit" Name="exitItem" Click="exitItem\_Click" Foreground="Black"/>

</MenuItem>

</Menu>

## Hoofdstuk 19: Consoleprogramma’s

Bekijk de ppt snel en dan is het easy.

## Hoofdstuk 20: OO

### Richtlijnen:

Graphical user interface, text, application

Description automatically generated

## Hoofdstuk 22: Interfaces

Word gebruikt voor de structuur van een programma.

### Aanmaken van een interface:

Text

Description automatically generated with low confidence