

ModProg – I

1.1-2: Programmeren

computer processor en geheugen
opdracht processor verandert geheugen
methode opdr.→methode
klasse opdr.→meth.→klasse
namespace opdr.→meth.→klas.→namespace
variabele geheugenplaats met een naam
object variabelen→object

1.3 Programmeerparadigmas

declaratief functie-based
declaratief-functioneel legt funct. verbanden
declaratief-logisch propositiologica
imperatief opdracht-based (gebiedend)
enkel imperatief bv: Fortran, Basic
procedureel imperatief + methoden
procedureel bv: Python, PHP
object-georiënteerd procedureel + objecten
object-georiënteerd bv: C#, Java, C++

1.5 Vertalen

assembler vertaalt 1 sourcecode voor 1 processor
compiler ... vertaalt 1 sourcecode voor alle processors
interpreter voert sourcecode direct uit
Java compiler+interpreter →bytecode
C# compiler+compiler →intermediate language

2. C# programma's

opbouw broncode class{methode{opdracht}}}
class public/private
class-members (public/private)(static)(type/void)
method body declaratie/opdracht
declaratie reserveert geheugen
declaratie `int x; string naam;`
opdracht aanroep/toekenning
opdracht-aanroep roept andere methode aan
opdracht-aanroep `Console.WriteLine(x);`
opdracht-toekenning verandert het geheugen
opdracht-toekenning `x = 1; naam.Text = "a";`
property eigenschap van een object
subklasse subversie van bestaande klasse
subklasse `class scherm : Form`
constructormethode maakt **new** object
constructormethode methode met naam van class

this object dat door methode bewerkt wordt
public bruikbaar in andere classes
property van object `naam.Length`
niet-static methoden werkt op object
niet-static methoden `naam.ToUpper();`
static class ipv object
static properties `Color.Yellow`
static methoden `Console.WriteLine(x)`
Main altijd **static**
constructor nooit **static**
eventhandler `this.Paint+=teken;`
eventhandler toekenning (geen aanroep)

3. Teken

library `using System.Drawing;`
Paint-event (object o, EventArgs pea)
`pea.Graphics` property van pea
`pea.Graphics.DrawLine()` tekentlijnen
`DrawLine,Rectangle,Ellipse` gebruiken Pen
`FillRectangle,Ellipse` gebruiken Brush
`DrawString` gebruikt Brush

3.3-5 Berekeningen

expressie alle code met een waarde
expressie ... `int, string, object-waarde, constructor new`
`new Form()` heeft **Form**-object als waarde
expressie kun je *uitrekenen* en heeft *waarde*
opdracht kun je *uitvoeren* en heeft *effect*
expressies kunnen deel uitmaken van opdracht
modulo % `8%3=2`
const onaanpasbare declaratie
var declaratie met automatische typebepaling

Created by Raoul Grouls, 2017

Released under the MIT license.