Object-georiënteerd Programmeren II

Prof. dr. Kris Luyten

Bram Van Deurzen

Dries Cardinaels

Wat mag je verwachten?

- Je wordt efficiënter als software ontwikkelaar
  - Beter structureren/modelleren
  - Moeilijkere problemen oplossen
- Je beheerst verschillende programmeertechnieken
- Je kan sneller nieuwe talen onder de knie krijgen
- Een hoog tempo met leerstof die van een goede programmeerkennis uitgaat
  - Java, C en C++

Wat verwachten wij?

- Zin om te programmeren en experimenteren
- Zin voor initiatief en zelfstandig werken
- Kunnen is even belangrijk als kennen (maar dat maakt kennen niet minder belangrijk)

Wat moet je doen?

- *Aanwezig zijn* in de lessen
- *Notities nemen tijdens lessen*
- **Oefenen, oefenen, oefenen – ook in de andere programmeervakken**
- Opfrissen vorige programmeervakken indien nodig
- Zelfwerkzaamheid & kritisch reflecteren

Wat mag je vooral niet doen?

- Alles laten liggen tot op het einde
- Vragen **alvorens zelf te zoeken**
- Denken dat het niet jouw fout is, want dat is het meestal wel…

Overzicht van het vak: thema's

- Programmeren en Programmeerparadigma's
- OO structuren en patronen
- Defensief programmeren (**!!!**)
- Memory management
- Meta-programming
- Closure
- Patronen
- …

Overzicht van het vak: examinering

- Schriftelijk examen: 70%
- “Challenges” (practica): 30%

Werking en Communicatie binnen het vak

(in tijden van COVID-19)

- bb.uhasselt.be
- Individuele Github account
  - voor je oefeningen en al je ander werk mbt dit vak. Gebruik opgegeven bestands- of foldernaam (e.g. **[Opdracht_01_01]**).
  - Onderwijsteam heeft ook toegang en volgt je mee op.
  - Hou ook een backwards planning/burndownchart bij voor al je studie-gerelateerde activiteiten (cfr. projectvaardigheden).
- Discord channel (invite link komt er nog aan):
  - Gebruik best een duidelijke nick name.
  - Zo houden we makkelijker contact ook buiten de les momenten.
  - Help elkaar, dat werkt vaak beter.
- Vragen worden ***niet*** *via email* gesteld: via Discord of tijdens de les.
- Er zijn Hoorcolleges, maar geen geplande responsies (enkel wanneer dat nodig blijkt).

Cursusmateriaal

- Ter beschikking gesteld per les / via het net
- Van externe bronnen (zoals websites) wordt duidelijk aangegeven wat je er van moet kennen, kunnen of beide.
- Geen aparte studieleidraad. Opdrachten staan in de slides

Vragen?

Programmeertalen

http://people.ku.edu/~nkinners/LangList/Extras/classif.htm

http://oreilly.com/news/graphics/prog_lang_poster.pdf

- Talen: Postscript, Oberon, Tcl/Tk, Fortran, Prolog, Pascal, Delphi, Python, Cobol, Modula, Ada, Rexx, ISO C, C#, Javascript, ANSI/ISO C++, Ruby, Self, Eiffel, PHP, Perl, ML, Lisp, Objective-C, VB, Scheme, Haskell, Caml, Smalltalk, Miranda, PL/1, Simula, Java...
- Belangrijkste klassen: imperatief, functioneel, object-georiënteerd en logisch

Programmeertalen

http://people.ku.edu/~nkinners/LangList/Extras/classif.htm

http://oreilly.com/news/graphics/prog_lang_poster.pdf

- Talen: Postscript, Oberon, Tcl/Tk, Fortran, **Prolog**, Pascal, Delphi, Python, Cobol, Modula, Ada, Rexx, ISO C, C#, Javascript, ANSI/ISO C++, Ruby, Self, Eiffel, PHP, Perl, ML, Lisp, Objective-C, VB, Scheme, Haskell, Caml, Smalltalk, Miranda, PL/1, Simula, Java...
- Belangrijkste klassen: imperatief, functioneel, object-georiënteerd en **logisch**

Logisch programmeren

- Programmeren gebaseerd op logica
- Prolog
  - Predicatenlogica
  - Declaratief (*wat* niet *hoe)*
  - Facts (feiten) en clauses (regels)
  - Gebruikt in o.m. expertensystemen, gegevens analyse en semantisch web

Logisch programmeren: Prolog

facts

```
parent(Frans, Eefje).
parent(Klaar, Eefje).
parent(Eefje, Salammbo).
parent(Eefje,Mattho).
parent(Gustave, Salammbo).
parent(Gustave,Mattho).
```

Clause definieert een relatie

```
grandparent(X,Y) :-
        parent(X,Z),
        parent(Z,Y).
```

Ondervragen van een Prolog programma

```
? – parent(Eefje, Mattho).
        yes
? – grandparent(Frans,Salammbo).
        yes
? – grandparent(Klaar, X).
        X = Salammbo.
        X = Mattho.
```

Logisch programmeren: Prolog

- Verschillende interpreters en compilers: SWI-Prolog, GNU-Prolog, Sicstus Prolog, Amzi! Prolog,…
- [niet verplicht] Zelf al wat meer leren? http://www.krisluyten.net/wp-content/uploads/2014/01/GPT-les8-prolog.pdf
  "Adventure in Prolog" is ook een uitstekende resource om mee te starten: http://www.amzi.com/AdventureInProlog/

Programmeertalen

http://people.ku.edu/~nkinners/LangList/Extras/classif.htm

http://oreilly.com/news/graphics/prog_lang_poster.pdf

- Talen: Postscript, Oberon, Tcl/Tk, Fortran, Prolog, Pascal, Delphi, Python, Cobol, Modula, Ada, Rexx, ISO C, C#, Javascript, ANSI/ISO C++, Ruby, Self, Eiffel, PHP, Perl, **ML**, **Lisp**, Objective-C, VB, **Scheme**, **Haskell**, Caml, Smalltalk, **Miranda**, PL/1, Simula, Java...
- Belangrijkste klassen: imperatief, **functioneel**, object-georiënteerd en logisch

Functioneel programmeren

- Gebruikt functies (maar dan die gebaseerd op Lambda calculus)
- Maar programma heeft geen *state*
- Geen gebruik van “destructive updates” (bijv. variabele updates) == *puur* functioneel
- Functionele programmeertalen zijn bijzonder geschikt om *lijsten* mee te verwerken en bewerken
- [niet verplicht] Interesse in meer? http://www.krisluyten.net/wp-content/uploads/2014/01/GPT-les2-haskell.pdf en http://www.krisluyten.net/wp-content/uploads/2014/01/GPT-les3-haskell.pdf

Programmeertalen

http://people.ku.edu/~nkinners/LangList/Extras/classif.htm

http://oreilly.com/news/graphics/prog_lang_poster.pdf

- Talen: Postscript, Oberon, Tcl/Tk, Fortran, Prolog, **Pascal**, Delphi, Python, Cobol, Modula, **Ada**, Rexx, **ISO C**, C#, Javascript, ANSI/ISO C++, Ruby, Self, Eiffel, PHP, Perl, ML, Lisp, Objective-C, VB, Scheme, Haskell, Caml, Smalltalk, Miranda, PL/1, Simula, Java...
- Belangrijkste klassen: **imperatief**, functioneel, object-georiënteerd en logisch

Programmeertalen

http://people.ku.edu/~nkinners/LangList/Extras/classif.htm

http://oreilly.com/news/graphics/prog_lang_poster.pdf

- Talen: Postscript, Oberon, Tcl/Tk, Fortran, Prolog, Pascal, Delphi, Python, Cobol, Modula, Ada, Rexx, ISO C, **C#**, Javascript, **ANSI/ISO C++**, Ruby, Self, Eiffel, PHP, Perl, ML, Lisp, **Objective-C**, VB, Scheme, Haskell, Caml, Smalltalk, Miranda, PL/1, Simula, **Java**...
- Belangrijkste klassen: imperatief, functioneel, **object-georiënteerd** en logisch

Programmeertalen

- **Opdracht_01_01** Wat is een programmeertaal? Omschrijf wat een programmeertaal is aan de hand van je huidige kennis!
- **Opdracht_01_02** Wat is een programmeerparadigma?

  Stijl van programmeren die invloed heeft op de manier waarop de uitvoering van de code gebeurt en die getypeerd wordt door de elementen die in de programmeertaal gebruikt worden (wel/geen variabelen, functies, objecten,...)

Multi-Paradigm Programmeertalen

- Programmeertalen die meerdere paradigma’ s ondersteunen
- Combinaties procedureel+OO (bijv. C++) en logisch+functioneel (bijv. Mercury) komen veel voor
- **Python**

Vragen?