

# PROGRAMMEREN 2

# INLEIDING

# OVERZICHT

- Kennismaking
- Inleiding tot dit vak
- Doelstellingen, leerstof en evaluatie
- Praktische zaken en tips

# WELKOM!

## KENNISMAKING

# OVER MIJ

- ir. Thomas Vranken
- Computerwetenschappen aan de KU Leuven
- Educatieve master Wetenschappen en Technologie aan de KU Leuven
- Software engineer bij Twipe Mobile Solutions
- Lesgever bij CodeFever (wanneer er tijd is)



# OVER JULLIE

- Naam?
- Specialisaties binnen de lerarenopleiding?
- Andere relevante ervaringen voor dit vak? Jobstudent, werkervaring, voorgaande opleidingen, ...

# POSITIONERING IN OPLEIDING

# LINK MET ANDERE VAKKEN

Programmeren staat centraal in het domein van informatica. Binnen de lerarenopleiding van de UCLL zijn er 2 vakken die zich hierop richten:

- Programmeren 1 - 5 SP - 1e fase
- Programmeren 2 - 7 SP - 2e fase

Uit de **ECTS-fiche**:

*"Dit opleidingsonderdeel bouwt expliciet voort op Programmeren 1 in de eerste opleidingsfase en vooronderstelt dat de student dit met succes heeft doorgemaakt. Evenwaardige competententies verworven in een andere context voldoen desgevallend ook."*

# LINK MET ANDERE VAKKEN

Eens je kan programmeren, kan je dit benutten in andere vakken. Bijvoorbeeld:

- Webdesign en multimedia
- Applicatiesoftware
- Andere onderwijsvakken?
- ...



# STUDIEBELASTING EN AANPAK

7 SP  $\approx$  175 - 210 uren

- ~45u les
- 125 - 160 uren zelfstandig werk:
  - Project
  - Examen

# DOELSTELLINGEN, LEERSTOF EN EVALUATIE

WAT JE OP HET EINDE KENT EN KUNT

# DOELSTELLINGEN

**Verdieping** kennis en beheersing van programmeerconcepten en -technieken uit Programmeren 1:

- Algoritmen ontwerpen en beschrijven
- Controlestructuren in algoritmen en programma's
- Datastructuren in Python (en daarbuiten)
- Iteraties (herhalingen)
- Recursie
- Objectgericht ontwerpen en programmeren

# DOELSTELLINGEN (2)

Noties bezitten van enkele fundamentele aspecten van algoritmiek, programmeren en informaticawetenschappen:

- Enkele klassieke algoritmes en oplossingswijzen
- Efficiëntie, complexiteit, berekenbaarheid
- Grafen- en automaten theorie (als er tijd voor is)
- Principes van kunstmatige intelligentie (als er tijd voor is)

Programmeren (in Python) kunnen plaatsen in een context:

- Programmeren en programmeertalen vroeger en nu
- Programmeren en **computationeel denken**

# DOELSTELLINGEN (3)

Conceptueel **inzicht** in en elementaire **gebruiksvaardigheid** met een aantal min of meer courante benaderingen en omgevingen voor programmeren en/of computationeel denken in het secundair onderwijs.

Zelf, in **samenwerking met collega's**, benaderingen en werktuigen voor het aanleren van computationeel denken en/of programmeren in de eerste en/of de tweede graad van het secundair onderwijs kunnen **opsporen, onderzoeken, uittesten**, in minstens één ervan een project voor beoogde leerlingen **uitwerken**, en dit alles aan collega's **verduidelijken** zowel conceptueel als qua gebruik.

→ via project in groep

# LEERSTOF

- Slides zijn voornaamste bron
- Handboek: de Programmeursleerling
- Bijkomende materialen via Toledo

# EVALUATIE

Project "Programmeren in het sec. onderwijs":

- Telt mee voor 50% van het totaal voor dit vak
- In groep
- Opstartsessie voorzien begin oktober: meer info over doelen, werkwijze, evaluatie
- In principe krijgt ieder teamlid hetzelfde punt, maar hier kan van afgeweken worden op basis van feedback. De docent beslist uiteindelijk.

# EVALUATIE (2)

Examen over de leerstof:

- Telt mee voor 50% van het totaal voor dit vak
- Mondelinge verdediging
- Bespreking en verdediging van het portfolio met programmeeroefeningen dat je doorheen het OPO opbouwt:
  - Inzicht in concepten
  - Procedurele vaardigheden (concepten toepassen)
  - Metacognitieve kennis
  - Attitude: doorzettingsvermogen, samenwerking, initiatief, leergierigheid

Hoe dit voor te bereiden:

- Maak de oefeningen tijdens het jaar en hou ze geordend bij
- Verzin zelf vragen/oefeningen en probeer ze op te lossen
- Ga online op zoek naar nieuwe oefeningen/theorie/concepten

Het examen is open boek. Je brengt je portfolio mee en mag het raadplegen tussendoor. Je mag ook online informatie opzoeken tijdens het examen.



## EVALUATIE (3)

Tijdens de tweede zittijd kan enkel het examen opnieuw gedaan worden. Je punt van het project wordt ongewijzigd overgenomen.

# **VARIA**

## **PRAKTISCHE ZAKEN, TIPS**

# PRAKTISCH

- Lessen op maandagavond en donderdagvoormiddag
  - Let op met sluiting van de poort/fietsenstalling
- Zie lessenrooster voor exacte data
- Meestal op de campus, maar een paar zullen online zijn. Dit wordt op voorhand aangekondigd via Toledo
- De les op donderdag 12/10 wordt vervangen door een online zelfstudie ter voorbereiding van het project

# TIPS OM TE LEREN PROGRAMMEREN

- Programmeren leer je **niet** door alleen maar code te lezen, maar door ze **zelf** te schrijven
- Belangrijkste is om de concepten en het denkproces onder de knie te krijgen:
  - Afkijken van oplossingen heeft weinig zin, als je er niets mee doet
  - Veel oefeningen (her)maken is nuttig

**Zelf (online) kunnen zoeken naar informatie, is cruciaal voor een programmeur.**

Neem initiatief en raadpleeg online/offline bronnen, vraag raad aan medestudenten of de lector, ...