

Curs ID-Introducere

2018-2019

Programare Logică

Cuprins

1 Organizare

2 Privire de ansamblu

Organizare

Informații

Curs:

- **Ioana Leuștean**

Seminar/Laborator

- **Carmen Chiriță**
- **Ioana Leuștean**

Suport curs/seminar/laborator

- <https://cs.unibuc.ro/~ileustean/PL.html>
- <http://moodle.fmi.unibuc.ro/course/view.php?id=186>

Materialele au la bază cursul realizate în colaborare cu Denisa Diaconescu.

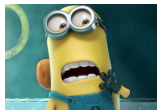
Notare

- **Laborator: 30 puncte**
- **Examen: 60 puncte**
- Se acordă 10 puncte din oficiu!

Notare

- ❑ **Laborator: 30 puncte**
- ❑ **Examen: 60 puncte**
- ❑ Se acordă 10 puncte din oficiu!

- ❑ Condiție minimă pentru promovare:
laborator: minim 15 puncte și
examen: minim 25 puncte.

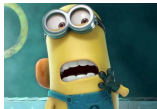


Notare

- **Laborator: 30 puncte**
- **Examen: 60 puncte**
- Se acordă 10 puncte din oficiu!

- Condiție minimă pentru promovare:
laborator: minim 15 puncte și
examen: minim 25 puncte.

- Ambele probe se dau în scris.
- Materiale ajutătoare: cursurile și laboratoarele (tipărite).



□ Curs

□ Programare logică

- Logica de ordinul I
- Algoritmul de unificare
- Semantica de punct fix a programelor în Prolog
- Formă prenex, formă Skolem, univers Herbrand
- Rezoluție în logica de ordinul I
- Rezoluție SLD

□ Laborator

□ limbajul Prolog

- M. Ben-Ari, **Mathematical Logic for Computer Science**, Springer, 2012.
- P. Blackburn, J. Bos, K. Striegnitz, **Learn Prolog now**, College Publications, 2006.
- M. Huth, M. Ryan, **Logic in Computer Science: Modelling and Reasoning about Systems**, Cambridge University Press New York, 2004.
- J.W. Lloyd, **Foundations of Logic Programming**, Springer, 1987.

Privire de ansamblu

Principalele paradigme de programare

- Imperativă (cum calculăm)

 - Procedurală

 - Orientată pe obiecte

- Declarativă (ce calculăm)

 - Logică

 - Funcțională

Principalele paradigme de programare

- Imperativă (cum calculăm)

 - Procedurală

 - Orientată pe obiecte

- Declarativă (ce calculăm)

 - Logică

 - Funcțională

La acest curs, veți învăța
programare logică.

Programare declarativă

- Programatorul spune **ce** vrea să calculeze, dar nu specifică concret **cum** calculează.
- Este treaba interpretorului (compiler/implementare) să identifice cum să efectueze calculul respectiv.
- **Programarea logică** este un tip de programare declarativă!
- Tipuri de programare declarativă:
 - Programare logică (e.g., Prolog)
 - Programare funcțională (e.g., Haskell)

Programare logică

- Programarea logică este o paradigmă de programare bazată pe logică formală.

Programare logică

- Programarea logică este o paradigmă de programare bazată pe logică formală.
- Unul din sloganurile programării logice:

Program = Logică + Control (*R. Kowalski*)

Programare logică

- Programarea logică este o paradigmă de programare bazată pe logică formală.
- Unul din sloganurile programării logice:
Program = Logică + Control (*R. Kowalski*)
- Programarea logică poate fi privită ca o deducție controlată.

Programare logică

- Programarea logică este o paradigmă de programare bazată pe logică formală.
- Unul din sloganurile programării logice:
Program = Logică + Control (R. Kowalski)
- Programarea logică poate fi privită ca o deducție controlată.
- Un program scris într-un limbaj de programare logică este
o listă de formule într-o logică
ce exprimă fapte și reguli despre o problemă.

Programare logică

- Programarea logică este o paradigmă de programare bazată pe logică formală.
- Unul din sloganurile programării logice:
Program = Logică + Control (R. Kowalski)
- Programarea logică poate fi privită ca o deducție controlată.
- Un program scris într-un limbaj de programare logică este
o listă de formule într-o logică
ce exprimă fapte și reguli despre o problemă.
- Exemple de limbaje de programare logică:
 - Prolog
 - Answer set programming (ASP)
 - Datalog

Ce veți vedea la laborator

Prolog

- bazat pe logica clauzelor Horn
- semantica operațională este bazată pe rezoluție
- este Turing complet
- vom folosi implementarea [SWI-Prolog](#)

Ce veți vedea la laborator

Prolog

- ☐ bazat pe logica clauzelor Horn
- ☐ semantica operațională este bazată pe rezoluție
- ☐ este Turing complet
- ☐ vom folosi implementarea [SWI-Prolog](#)

Limbajul Prolog este folosit pentru programarea sistemului IBM Watson!



Puteți citi mai multe detalii [aici](#).