Declarative Programming

Introduction and Organization

Bernhard K. Aichernig, Felix Wallner

Institute of Software Engineering and Artificial Intelligence
Graz University of Technology
Austria

Summer Term 2025

Version: February 25, 2025

Declarative Programming?

- Declarative Programming (DP) is a programming paradigm that expresses the logic of computation without describing its control flow.
- DP is a non-imperative style of programming in which programs describe their desired results without explicitly listing commands or steps that must be performed.
- Subparadigms include:
  - Functional Programming
  - Logic Programming

Goals

- Getting familiar with declarative programming styles
- Introduction to a
  - Functional Programming Language (Scala)
  - Logic Programming Language (Prolog)
- Knowing the similarities and differences between the two
- Improving the abstraction skills
- Better ability to design solutions for recursive problems
- Understanding the pros and cons of declarative programming
- Complementing the VU Software Paradigms

Organisation

- Registration in TUGRAZonline until 6 March 2025 !!!
- **Lectures**
  - March: Monday, 3 & 10 March 10:15-11:00, HS i13 & Wednesday, 19 & 26 March, 12:15-13:00, HS i12
  - April – June, Wednesday, 12:15 – 13:00, HS i12
  - 3 written exams:
    - Location: HS i13
    - Dates: Monday, 7 April; Monday, 19 May; Monday, 16 June
  - Retake exam: on Monday, 14 July *(see details on Slide 6)*
- **Practicals**
  - March: Monday, 3 & 10 March 11:00–11:45, HS i13 & Wednesday, 19 & 26 March, 13:00–13:45, HS i12
  - April – June, Wednesday, 13:00–13:45, HS i12
  - Publication of exercises in TeachCenter
- Lecture with continuous assessment (immanentem Prüfungscharakter)

Marking

- Maximum points: 100
- Exams: 33, 33 & 34 points, 45 Min.
- Weekly exercises for exam preparation:
  - not graded & not obligatory
  - covering exam topics
- Additional *voluntary* programming assignment
  - up to 10 bonus points
  - individual submission via TeachCenter + assignment interview
- Grading key:
  50,01% - 62,50%: genügend
  62,51% - 75,00%: befriedigend
  75,01% - 87,50%: gut
  87,51% - 100,0%: sehr gut
- To be positive, you need
  - $>$ 50 points not counting bonus points
- After participation in an exam you will be marked.

Organisation - Retake Exam

- The retake exam takes place on Monday, 14 July in HS i13 (16:00)
- The retake exam *replaces* one of the three partial exams
  - meaning the points from your lowest-points partial exam will be replaced with the points from the retake exam
- The content of the retake is *the entire course content*
  - not just the content of the partial exam it replaces
- You can participate if one of the following conditions is met:
  - You were *excused*[1] for one of the previous partial exams
  - You are negative (≤ 50 points)

[1] *Excused* means you sent us written confirmation for the reason of your absence, e.g., sick notification, at `dp.sai@tugraz.at` *and* we have officially excused you.

Exercises – Support

- First practical lecture: today, March 3
- Three study assistents
  - Verena Schaffer (Functional Programming 1, 2 & 3)
  - Florian Zanotti (Functional Programming 4 & Actors)
  - Christopher Liebmann (Logic Programming)
- **Questions?**
  Discord Server: SAI - Group Aichernig
- Only (!) for private organisational questions, e.g. sick notification:
  `dp.sai@tugraz.at`

Discord

- Discord Server: SAI - Group Aichernig , Category: Declarative Programming, Channels:
  - dp-announcements - important information on the course (read-only)
  - dp-lecture - questions about the topics discussed in the lecture
  - dp-organization - organisational questions about the exercises, exams, or lecture, e.g., exam registration & marking
  - dp-exercises - questions about exercises
- Rules
  1. Read the channels regularly
  2. Post to the correct channel
  3. Create threads for questions
  4. Search before you post
  5. Be precise
  6. Quote when you reply
  7. Stay on topic
  8. Be friendly
  9. Use your real name

Schedule

Date	Lecture	Exercise
3.3.	Intro and Organisation, Intro FP	Tools
10.3.	Intro FP, Forms of Recursion	FP1
19.3.	Higher-Order Functions, Lists	FP2
26.3.	Curried Functions, Higher-Order List Functions	FP3
2.4.	Immutable Classes, Actors	FP4
7.4.	Exam 1 (i13, 14:00-16:00)	
9.4.	Actors (cont.)	FP-Actors Assignment
	Easter break	
30.4.	Intro LP	FP-Actors Q & A
7.5.	Recursion	LP1 + FP-Actors Deadline
14.5.	Lists	LP2
19.5.	Exam 2 (i13, 14:00-16:00)	
21.5.	Tail Recursion	LP3
28.5.	Cuts	LP4
4.6.	Interpreter & DCGs	LP5
11.6.	CLP	LP6
16.6.	Exam 3 (i13, 12:00-14:00)	
18.6.	Symbolic AI	

Details on Exercises

- Study assistents will present exercises in the practical sessions
- Presented and additional exercises will be uploaded to the TeachCenter
- Exercises
  - not graded & not obligatory
  - are declarative programming tasks
  - cover exam topics
- The actor assignment will be graded for potential bonus points
  - not obligatory
  - individual assignment
  - will require an assignment interview

Further Resources

- Scala: `https://www.scala-lang.org`
- Scala book: Martin Odersky, Lex Spoon, Bill Venners, Programming in Scala: A Comprehensive Step-by-Step Guide, 3rd Edition, Artima Press, 2016.
- SWI Prolog: `https://www.swi-prolog.org`
- Prolog book: Leon Sterling, Ehud Shapiro, The Art of PROLOG, 2nd Edition: Advanced Programming Techniques (Logic Programming), MIT Press,1994.