04 Lab
Advanced FP in Scala

Mirko Viroli, Gianluca Aguzzi
{mirko.viroli,gianluca.aguzzi}@unibo.it

C.D.L. Magistrale in Ingegneria e Scienze Informatiche
Alma Mater Studiorum—Università di Bologna, Cesena

a.a. 2024/2025

Lab 04: Outline

Outline

- Exercises on advanced mechanisms of functional programming in Scala
  - ▶ Experiments with advanced ADTs modeling
  - ▶ Experiments with type classes and monads
  - ▶ (Optional) try to extend the MVC example

Getting started

- Fork/clone repository `https://github.com/unibo-pps/pps-lab04`
- Then, follow the instructions in the following slides

Exercise 1: ADTs

- Look at the package `tasks.adts` in the `u04lab.code` package
- **Task 1**: complete the `Ex1ComplexADT` module so that it passes the tests (`ComplexADTTest`).
  - ▶ Thinking at the ADT, what is the best way to represent a complex number?
- **Task 2**: implement the `Ex2SchoolModel` module along with the `SchoolModelTest` test
  - ▶ Follow the suggestions in the comments of the `SchoolModel` module
- **Task 3**: implement the `Ex3StackADT` along with the `StackADTTest` test
  - ▶ Follow the suggestions in the comments of the `StackADT` module

Exercise 2: Type classes

- Look at the package `tasks.typeclasses` in the `u04lab.code` package
- **Task 4**: follow the instructions in the comments of the `Ex4Summables` module
  - ▶ this is a one-kinded type class, composed of methods `zero` and `sum`
- **Task 5**: follow the instructions in the comments of the `Ex5Traversable` module
  - ▶ this is an higher-kinded type class, so it takes a type constructor as a parameter

Exercise 3: Monads

- Look at the package `tasks.monads` in the `u04lab.code` package
- **Task 6**: implement the `Try` monad as described in the comments of the `Ex6TryMonad` module
    - ▶ Try is similar to the Optional monad, but it can carry an exception
    - ▶ Understand the implementation of the `Try` monad
    - ▶ Complete the tasks and verify that the main works

Optional exercises

- Consider the MVC example shown in the lecture
- **Task 7**: extend the counter with a GUI by adding a necessary text field and a "set" button, so that pressing the "set" button sets the counter to the value in the text field (if valid).
- **Task 8**: using a GUI with a label, a button, and a text field, and by implementing the model with state, try to realize the MVC (Model-View-Controller) for "drawmynumber".