

**Course Unit Software Language Engineering**  
**Masters in Informatics and Computing Engineering (M.EIC)**  
**Department of Informatics Engineering**  
**University of Porto/FEUP**  
**1<sup>st</sup> Semester - 2024/2025**

**A DSL for Data Extraction from Arbitrary Sources**

**Assignment #3**

*Cal uses several tools to obtain results and would like to aggregate dispersed data in a single table in a systematic way. In this case, they are using three tools, and each one outputs reports in a different format (i.e., XML, JSON and YAML).*

*Cal wants to extract just a small, localized portion of the data in the report of each tool, which will be translated to columns of the table that they want to build. A single execution of the three tools will generate data for a single line of the table.*

*Cal was able to configure the execution of the tools so that their output is organized in the following structure:*

```
| -01
|   |- analysis.yaml
|   |- analysis.xml
|   |- profiling.json
| -02
|   |- analysis.yaml
|   |- analysis.xml
|   |- profiling.json
...
```

*From the YAML report, Cal wants to extract the table 'dynamic' that is under the element 'total' and rename the columns by adding the suffix " (Dynamic)"; from the XML report, the table 'static' that is under the element 'total' and rename the columns by adding the suffix " (Static)". From the JSON report, the names and percentages of the three top functions, with column names "name #1", "% #1", "name #2", etc.*

*The final table should have a line per execution (i.e. folder) and columns from the three tools, plus a column with the name of the folder. Additionally, Cal wants to add two lines at the end, one with the sum and another one with the average for the columns that represent numbers.*

By the end of this assignment, it is expected that you have:

- An external DSL that configures and executes the semantic model.
- An expanded version of the semantic model which includes support for simple operations and types.

Tips:

- If you continue to support your configuration file, you can translate the external DSL to a configuration file. If not, you can have code that interprets the Abstract Syntax Tree (AST).
- The technology Xtext, used to implement DSLs, will be presented during class, but for this assignment you can use any technology to implement the parser for the external DSL (e.g., ANTLR, hand-made parser). However, keep in mind that the exam will have questions related to Xtext.
- Some Xtext-related resources:
  - o [15 Minutes Tutorial](#)
  - o [Explanation about Xtext cross-references](#)
  - o [Using Xtext parser in stand-alone mode](#)
  - o [Code example of using Xtext-generated parser in project](#)