

SUPSI

Educational project

Studente/i	Relatore	Correlatore
Giada Galdiolo Ambra Giuse Graziano	Giancarlo Corti	Tommaso Agnola
Corso di laurea	Modulo / Codice Progetto	Anno
Ingegneria Informatica	Software Engineering I	2024/2025
Committente	Data	
	03/2025	

Table of contents

1. Problem
2. Approach
3. Results
4. Conclusion

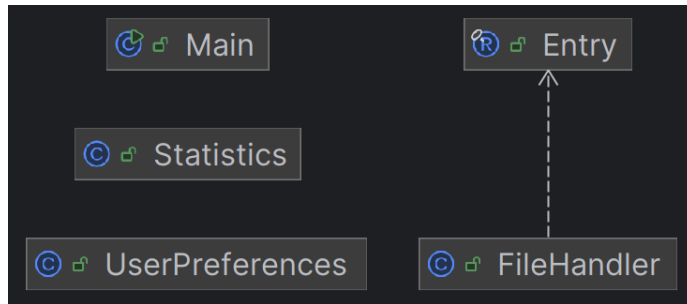
Problem

The goal was to develop a software product that could:

- Read movie data from a CSV file
- Compute useful statistics
- Save the results in an output CSV file
- Allow users to set file paths through a preferences file
- Run as a standalone command-line program
- Run transparently on both MS-Windows, MacOS, and GNU/Linux

Approach

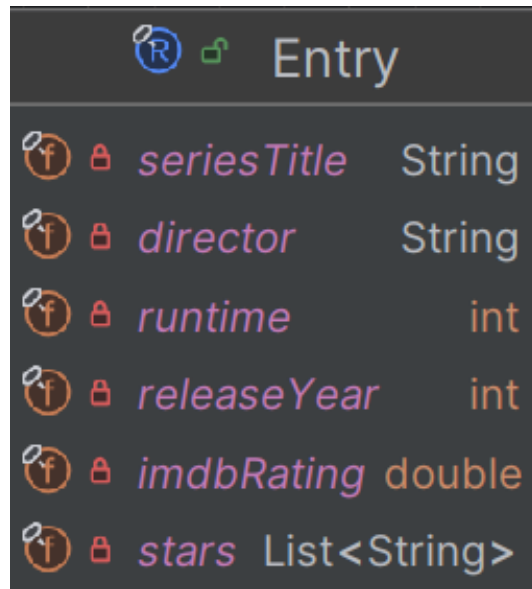
- **Requirements Elicitation and Formalization**
- **Source code and product versioning, software dependency** → Git, OpenCSV, *JSR 353*
- **Software design and development**
 - Main.java
 - Entry.java
 - FileHandler.java
 - Statistics.java
 - UserPreferences.java



Entry.java

Implemented as a record:

- Data immutable
- More concise syntax for storing movie details



FileHandler.java

- It uses the OpenCSV library to read and write CSV files efficiently.
- Handles possible errors in reading/writing

```
© FileHandler  
  
readEntries(Path) List<Entry>  
parseRuntime(String) int  
writeStatistics(Path, int, double, String, String, int) void
```

Statistics.java

- Class to compute key movie statistics
- It uses Java Streams for efficient data processing


```
© Statistics  
  
mostProductiveYear(List<Entry>) int  
averageRunTime(List<Entry>) double  
mostPresentActor(List<Entry>) String  
getDirectorAverageRating(String, List<String>, List<Double>) double  
totalMovies(List<Entry>) int  
bestDirector(List<Entry>) String
```

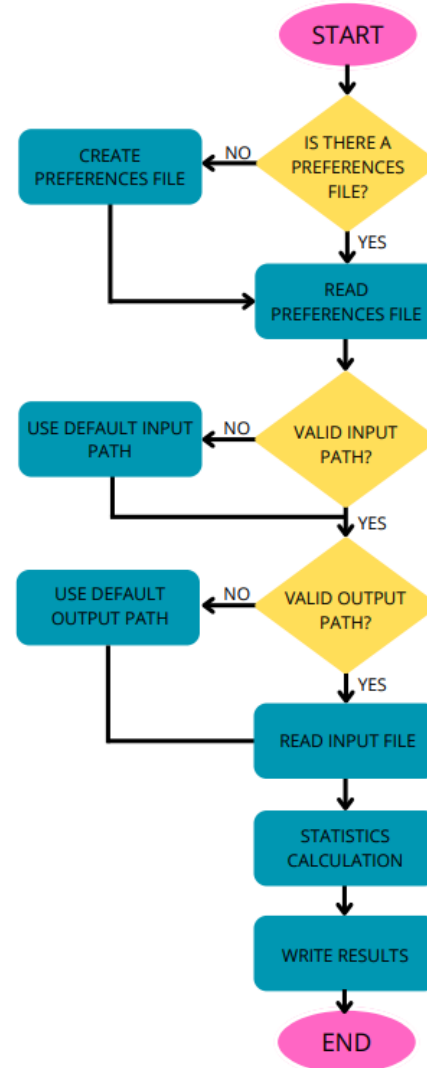
UserPreferences.java

- It manages user preferences through a JSON file
- It uses the *Java EE JSON Processing (JSR 353)* library for reading and writing user preferences in JSON format

© ⓘ UserPreferences		
🔗 🔒	<code>DIRECTORY_PATH</code>	Path
🔗 ⓘ	<code>PREFERENCES_FILE</code>	Path
🔗 🔒	<code>preferences</code>	Map<String, String>
🔗 🔒	<code>loadPreferences()</code>	void
🔗 🔒	<code>savePreferences()</code>	void
🔗 ⓘ	<code>getInputFilePath()</code>	String
🔗 ⓘ	<code>getOutputFilePath()</code>	String
🔗 🔒	<code>createDefaultPreferences()</code>	void
🔗 🔒	<code>parseJson(JsonObject)</code>	Map<String, String>
🔗 🔒	<code>toJson(Map<String, String>)</code>	JsonObject

Results

- Requirements met 
- Demo



Conclusion

Summary of Work

- Developed a tool for automatic movie data analysis
- Processes data, computes statistics, and saves results in CSV
- Configurable via preferences file and cross-platform compatible

Challenges Faced

- Proper handling of CSV files
- Managing user preferences and configuration settings

Thank You for Your Attention!

We are happy to take any questions you may have