# INFO0027

# Project2 Presentation

# Project2: File explorer

- For this project, implement the **logic** of a graphical file explorer in `Java`.
- This application is only a simulation of a file explorer: you must not deal with real files and/or folders.
- The objective is to let you apply design patterns (creational, structural and behavioural).

Project is available on my website: http://www.montefiore.ulg.ac.be/~gain/courses/info0027.php#projects

# Project2: File explorer
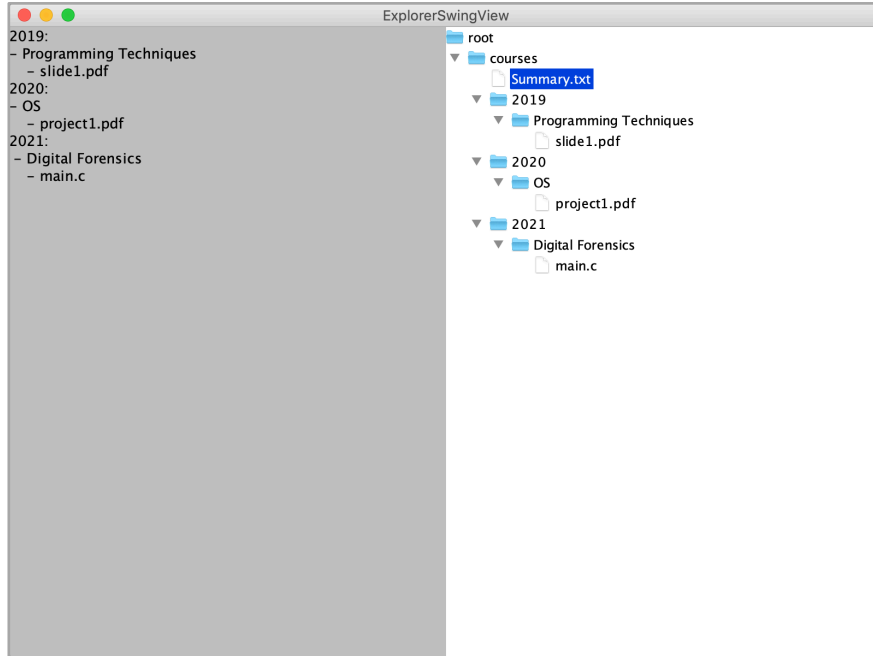


The GUI is to be handled by a library that will be made available to you: <u>not have to program it by yourself</u>.

⚠ You must only implement the logic.

The library is provided as a jar file called `graphics.jar.` It contains:

- Several <u>methods</u> to manipulate the graphical elements: add a new node to the tree, display text in the text area, …

- Several <u>callbacks</u> for signaling that the user has performed some actions: click on the "create file" menu, exit the program, …

# Project2: File explorer
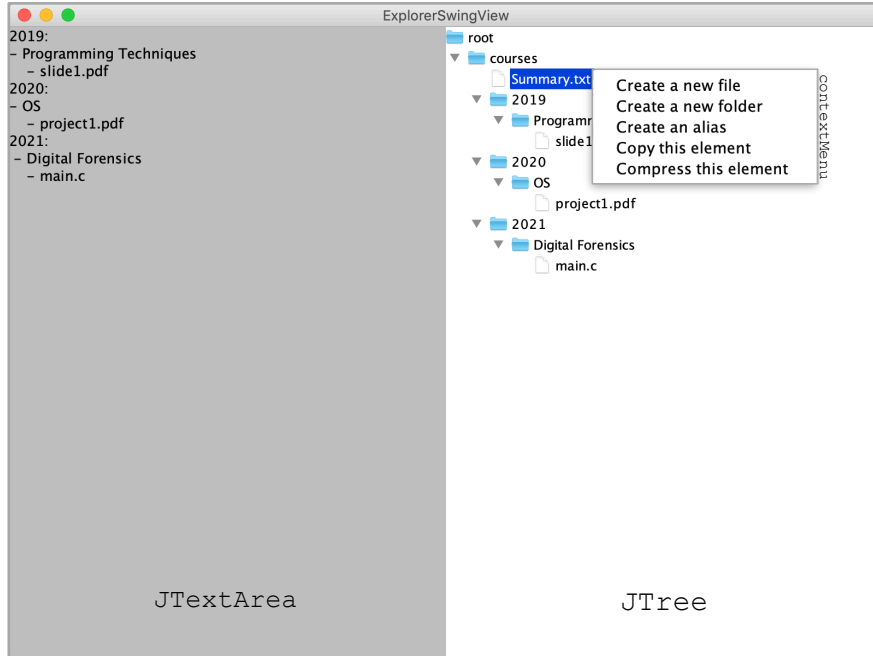


The application is composed of two main components:

1. a `JTree` represents the hierarchical file system: interact with it to add new files/folders.

2. a `JTextArea` to display the content of a file/folder.

In addition, a `contextMenu` is displayed when the user performs a right click on the `JTree` component.

It contains the following menu options: create a file, create a folder, copy a file/folder, make an alias and compress a folder.

# Project2: File explorer
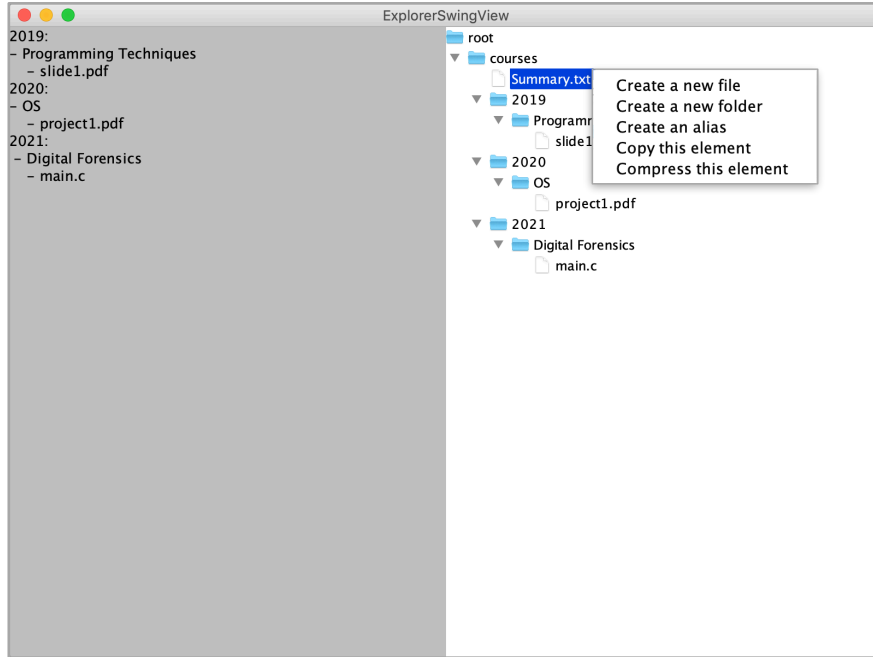


**Several features**:

1. Create a file
2. Create a folder
3. Make an alias (shortcut)
4. Copy a file/folder
5. Compress a folder
6. Log the user activity: keep track of the user actions by record them in the console/in a file.

According to the feature, design patterns are <u>not always</u> necessary.

There exist some restrictions: we can't add a file/folder to a file, …

# Project2: File explorer

Free concerning the implementation **but**:

- You must provide a well-designed structure (forbidden to have only a single file containing all the code).

- Your program must have the two following classes:

  - `Main.java`: main class of the program which contains the entry point.

  - `GuiHandler.java`: handle the graphical events and will provide callbacks.

Provided (view website)

# Project2: File explorer

How many design patterns?

**Clue 1**: Between 1 and 100.

**Clue 2**: Between 1 and 10.

**Clue 3**: All types of design patterns.

**Clue 4**: Design patterns that have been seen at the theoretical course.

# Project2: File explorer

The following files will be provided on my website:

- Statements + Appendix (additional information about the library)
- A zip package: `Main.java`, `GuiHandler.java`, `TestA.java` and `TestB.java`.
- The library `graphics.jar` with JavaDoc.
- Images that show you some examples of the filesystem hierarchy.
- Additional resources:
  - `setupProject.pdf`: How to install and use the library.
  - `terminal.pdf`: How to use the terminal (`Java`).
  - `eclipse.pdf`: How to use eclipse.
  - `network8.pdf`: How to use the ms800 machines.

Quick demonstration

+ Setup the project (tips)

# Project2: File explorer

Bonus part:

- During a copy, the keyword `(copy)` is added after the name of the file/folder.

- If there are several copies: `file (copy) (copy) (copy) (copy)`

- You can avoid it by implementing a (bonus) feature:
    - `file`
    - `file (copy_1)`
    - `file (copy_2)`
    - `file (copy_3)`
    - `file (copy_4)`

# Project2: File explorer

Others:
- 20% of the final mark
- Group of two
- Submit a zip on the submission platform
- More info in the statements (view PDF file)

Don't forget: we want clean code (OOP), without error and, of course, we want design patterns.
- No design pattern, no mark.

# **Project2: File explorer**

**Deadline: 17th May 2019**

Happy coding!