INFO0027

Programming Techniques

Project #2

Academic year 2020-2021

LIÈGE université

PROJECT PRESENTATION

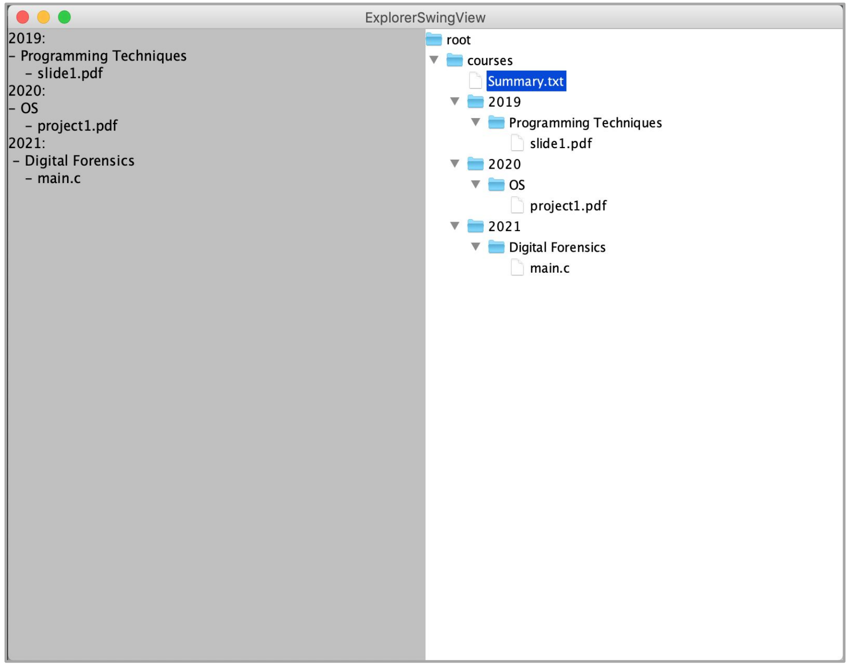
FILE EXPLORER

For this project, you will implement the logic of a **Graphical File Explorer** in *Java*:

- The application you will develop is only a **simulation** of a file explorer:
  - You **don't have** to deal with real file and/or folders.
- The *objective* is to let you apply design patterns (***creational***, ***structural*** and ***behavioural***).

Project statement is available on ecampus.

FILE EXPLORER



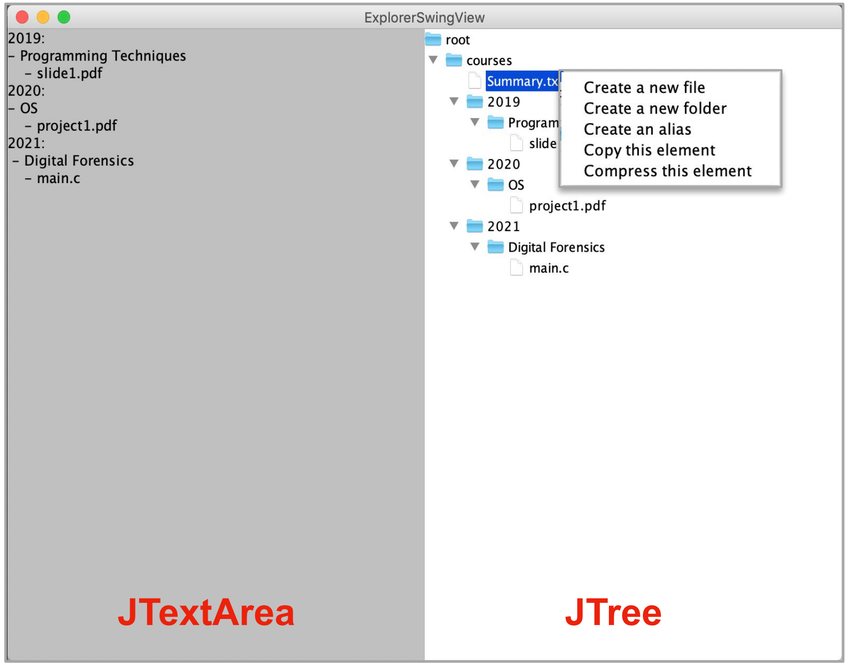
ExplorerSwingView

```
2019:
- Programming Techniques
   - slide1.pdf
2020:
- OS
   - project1.pdf
2021:
 - Digital Forensics
   - main.c
```

```
root
▼ courses
    Summary.txt
  ▼ 2019
    ▼ Programming Techniques
        slide1.pdf
  ▼ 2020
    ▼ OS
        project1.pdf
  ▼ 2021
    ▼ Digital Forensics
        main.c
```

- The **GUI** is provided:
  - Your task is to **implement the logic** !
- The library is provided as a ***jar* file** called *graphics.jar* and contains:
  - Several **methods** to manipulate the graphical elements - *e.g., add a new node to the tree, display text in the text area, … -.*
  - Several **callbacks** for signaling that the user has performed some actions: *-e.g., click on the “create file” menu, exit the program, … -.*

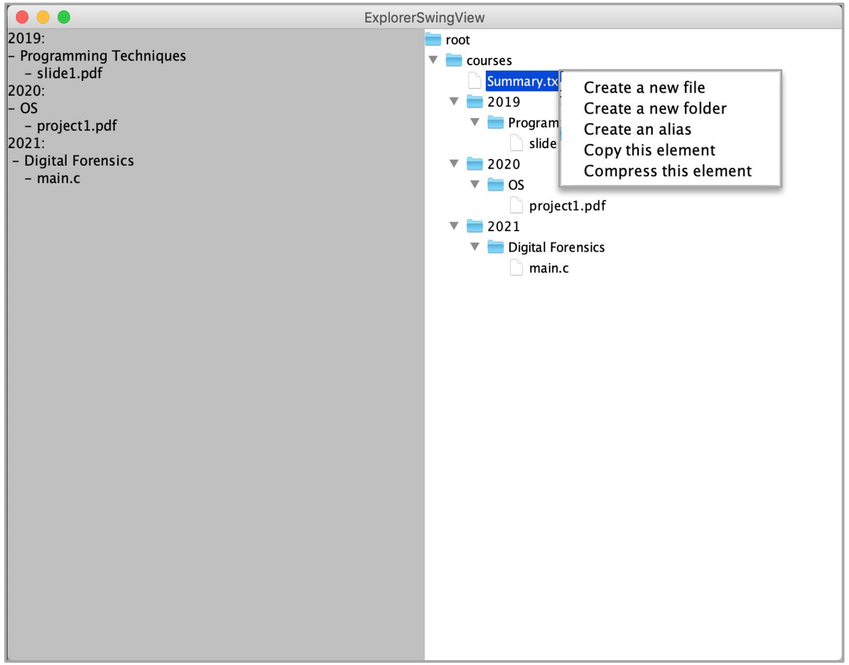
FILE EXPLORER



```
2019:
- Programming Techniques
   - slide1.pdf
2020:
- OS
   - project1.pdf
2021:
 - Digital Forensics
   - main.c
```

- The application is composed of **two main components**:
  - a ***JTree*** which represents the hierarchical file system.
  - a ***JTextArea*** which will be used to display the content of a file/folder.
- A ***contextMenu*** should be displayed when the user performs a right click on the ***JTree*** component.
  - It contains a menu allowing a user to *create a file/folder*, *copy a file/folder*, *make an alias* and *compress a folder* (to ignore)

FILE EXPLORER


ExplorerSwingView

```
2019:
- Programming Techniques
   - slide1.pdf
2020:
- OS
   - project1.pdf
2021:
 - Digital Forensics
   - main.c
```

```
root
▼ courses
    Summary.tx
  ▼ 2019
    ▼ Program
        slide
  ▼ 2020
    ▼ OS
        project1.pdf
  ▼ 2021
    ▼ Digital Forensics
        main.c
```

- **Several features**:
  - *Create a file*
  - *Create a folder*
  - *Make an alias (shortcut)*
  - *Copy a file/folder*
  - *Bonus - Log the user activity*

- **Several Constraints**:
  - Can’t add a file/folder to a file.
  - …

FILE EXPLORER

You are totally free regarding the implementation **BUT**:

- You should provide a **well-designed** architecture:
  - Using **design patterns** seen at the theoretical course (all types) !!
- You should **structure** your code (not a single file with all the code).
- Your program **must** use the two following classes:
  - ***Main.java*** which is the main class of the program which contains the entry point.
  - ***GuiHandler.java*** which handles the graphical events and provide callbacks.

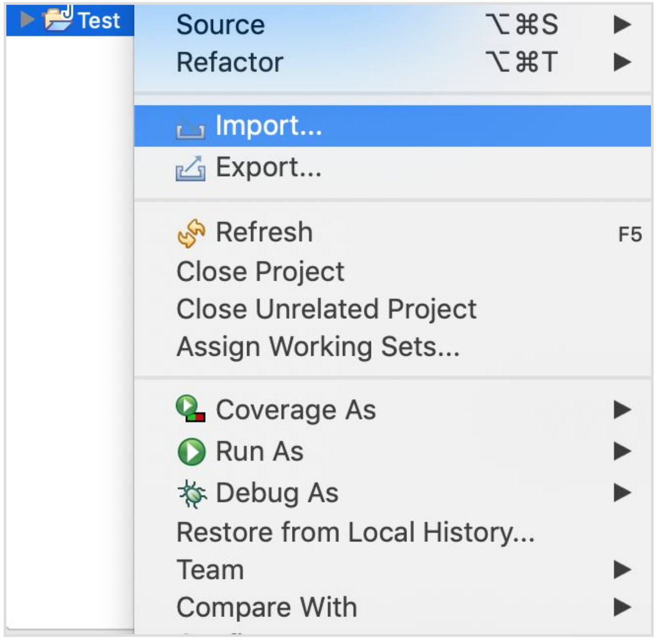
PROJECT SETUP

JAR FILES

A ***.jar file*** can be seen as the “executable” of the Java world:

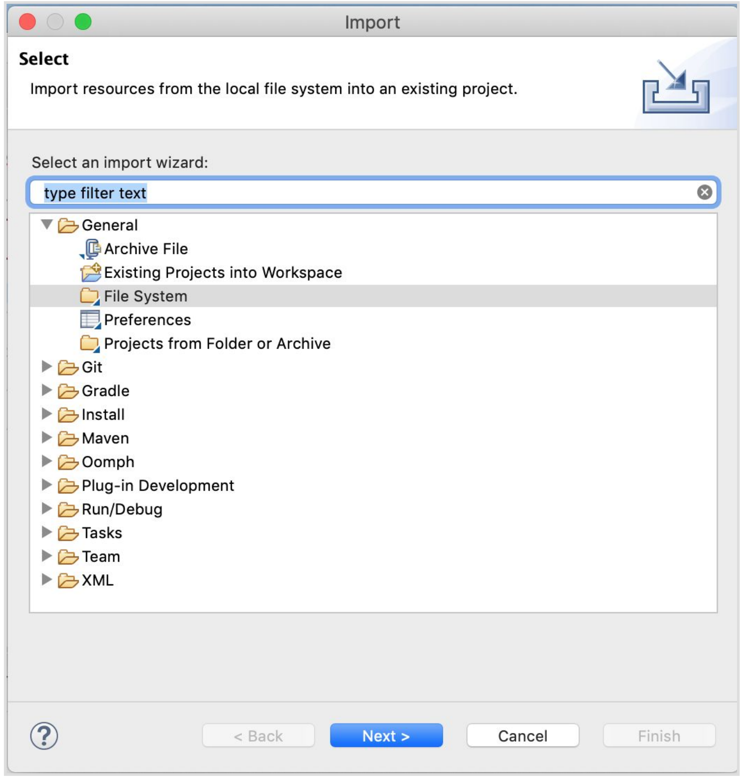
- It allows to distribute non executable libraries.
- It’s a simple ZIP archive with a metadata file.
- Eclipse can easily generate a .jar file (useless for you), import and integrate ***.jar files*** into your project!

ECLIPSE: IMPORT LIBRARY



1. **Right-click** on your project
2. Choose ***Import***

ECLIPSE: IMPORT LIBRARY



Import

Select

Import resources from the local file system into an existing project.

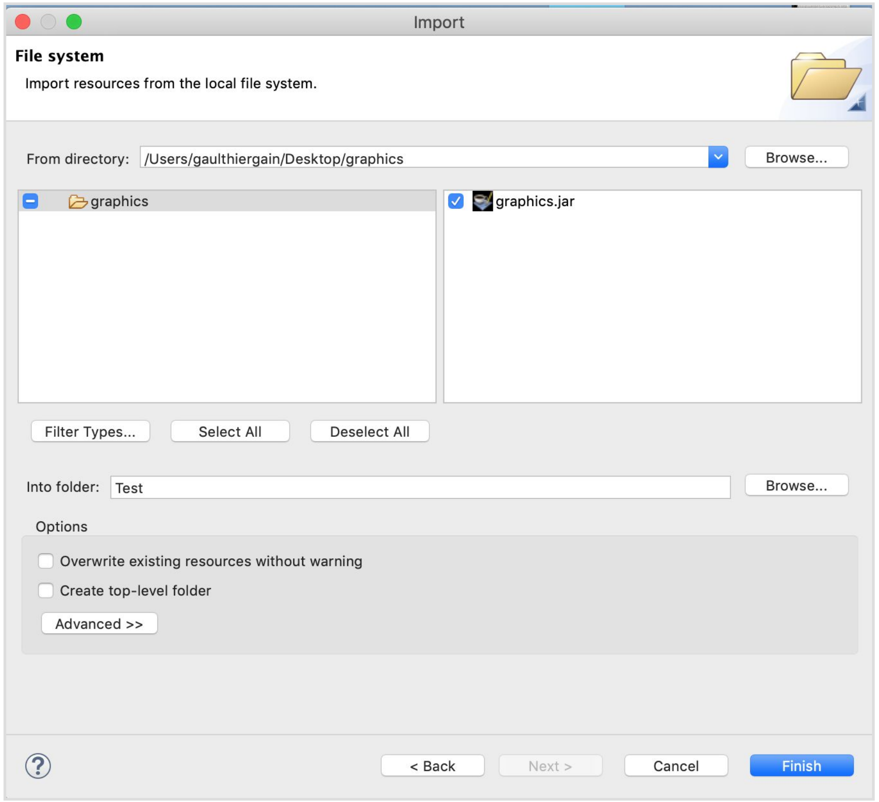
Select an import wizard:

type filter text

- General
  - Archive File
  - Existing Projects into Workspace
  - File System
  - Preferences
  - Projects from Folder or Archive
- Git
- Gradle
- Install
- Maven
- Oomph
- Plug-in Development
- Run/Debug
- Tasks
- Team
- XML

1. Choose:
   a. General
      i. File System
2. Click on ***Next>***

ECLIPSE: IMPORT LIBRARY



Import

File system

Import resources from the local file system.

From directory: /Users/gaulthiergain/Desktop/graphics

graphics

✓ graphics.jar

Into folder: Test

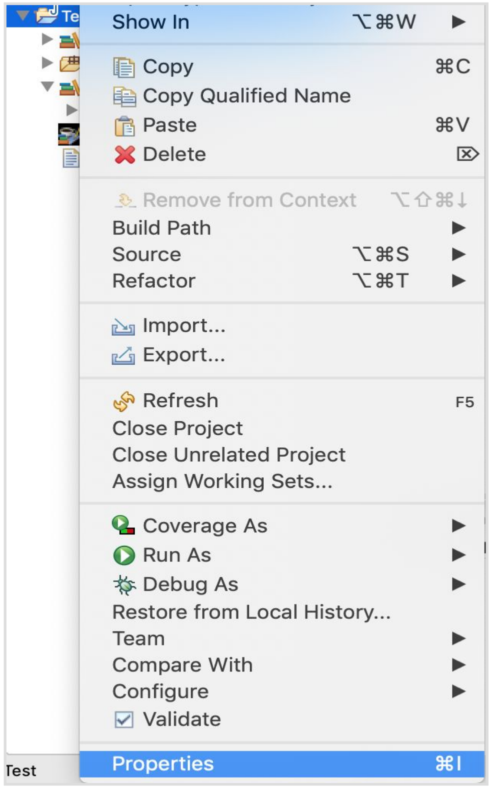
Options

☐ Overwrite existing resources without warning

☐ Create top-level folder

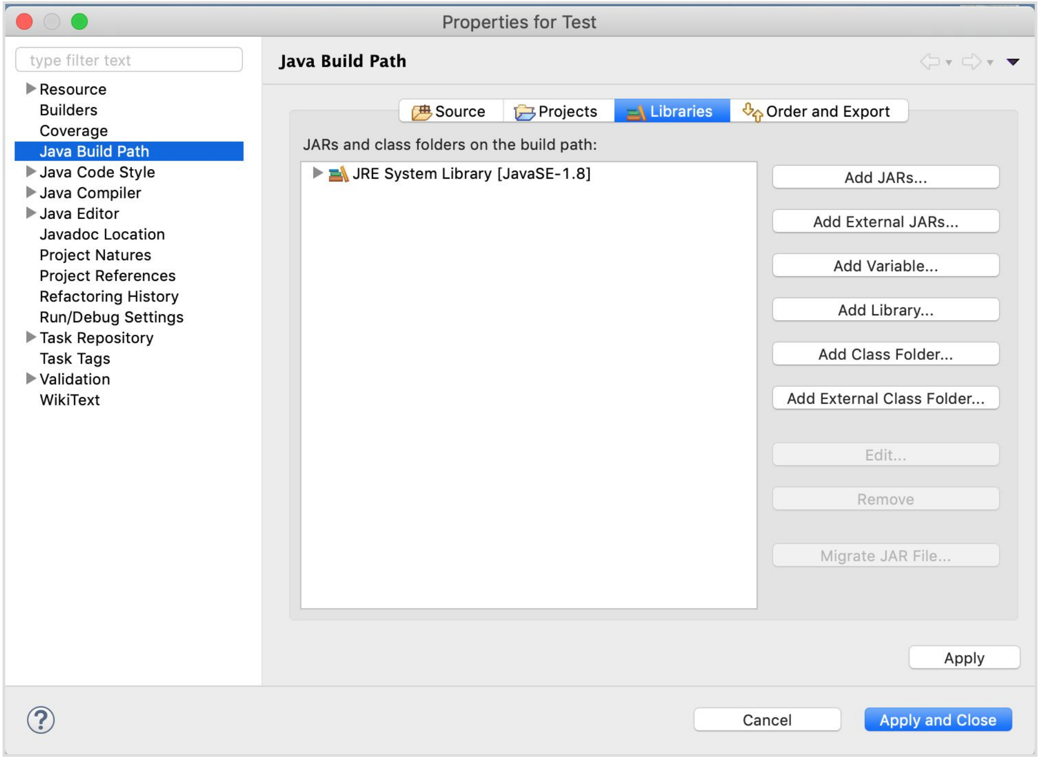
1. Click on **Browse** (1) & choose the folder containing graphics.jar
2. **Check** *graphics.jar* only (2)
3. Finally, click on **Finish**

ECLIPSE: IMPORT LIBRARY



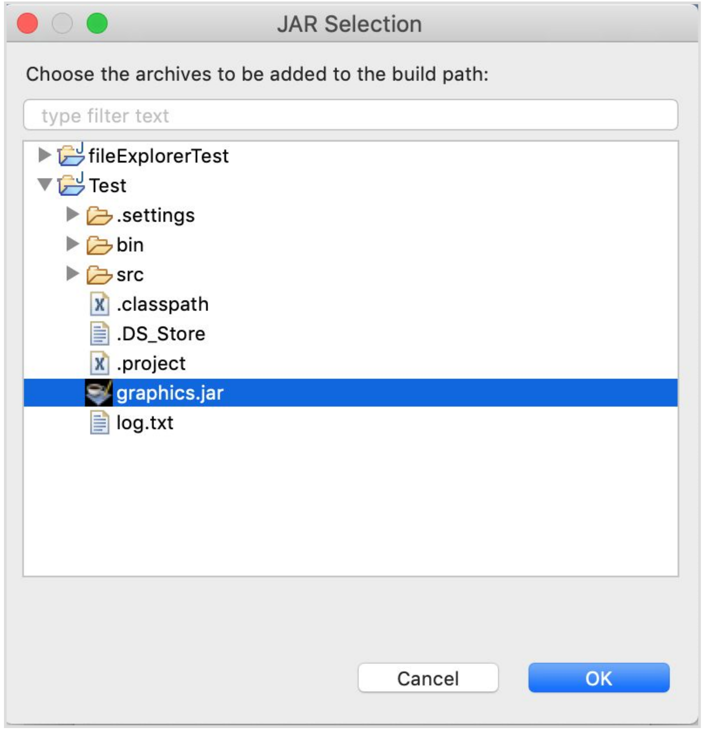
1. **Right-click** on your project
2. Choose ***Properties***

ECLIPSE: IMPORT LIBRARY



1. **Click** on *Java Build Path*.
2. Choose ***Libraries***
3. Click on ***Add JARs***

ECLIPSE: IMPORT LIBRARY


JAR Selection

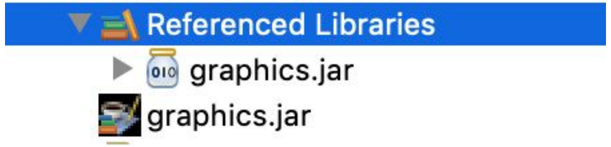
Choose the archives to be added to the build path:

type filter text

```
▶ fileExplorerTest
▼ Test
    ▶ .settings
    ▶ bin
    ▶ src
      .classpath
      .DS_Store
      .project
      graphics.jar
      log.txt
```

1. Click on ***graphics.jar***
2. Then, ***OK***
3. Finally, click on ***OK*** in the *properties window.*

ECLIPSE: IMPORT LIBRARY



1. ***Reference Libraries*** should be there.
2. ***graphics.jar*** should be inside **AND** at the root of your project.

IF YOU DON’T USE ECLIPSE

1. Source code in the ***src*** folder & bytecode in the ***bin*** folder.
2. ***graphics.jar*** file at the root.

Compilation

```
$ mkdir -p bin

$ javac -d bin -cp bin:graphics.jar $(find src -name '*.java' -print)
```

Execution

```
$ cd bin

$ java -cp .:../graphics.jar Main
```

HOW TO USE *ExplorerEvent* CLASS

The ***ExplorerEvent*** class declares methods for handling *mouse events*.

Part of your job is to **implement** those methods.

To do so,

1. Declare your ***handler*** class, using the `implements` keyword.
2. Rewrite *every* method signature with a body.

You can first try by printing simple message on a console for each method and see what happens when you interact with the GUI.

HOW TO USE *ExplorerEvent* CLASS

```
import montefiore.ulg.ac.be.graphics.ExplorerEventsHandler;

public class GuiHandler implements ExplorerEventsHandler {
    @Override
    public void doubleClickEvent(Object selectedNode) {}

    @Override
    public void createFileEvent(Object selectedNode) {}

    @Override
    public void createFolderEvent(Object selectedNode) {}


    //…
```

FILE EXPLORER

Additional Information:

- **20%** of the final mark.
- **Group of two**.
- Submit a **zip** on the submission platform.
- Questions via the ***Discussions*** section in **ecampus**.
- **Don’t forget**: We want **clean code** (OOP), **without error** and, of course, we want **design patterns**.
    - **No design pattern, no mark**.