

Java Programming II

Lab6

514770-1

Fall 2024

11/6/2024

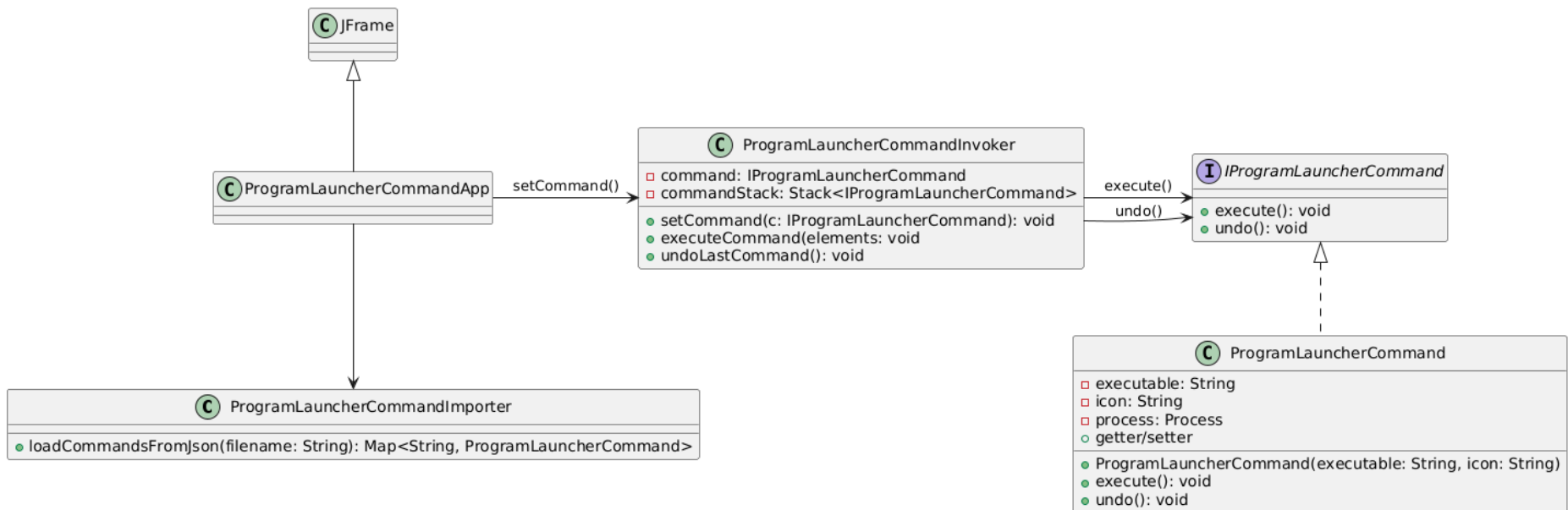
Kyoung Shin Park
Computer Engineering
Dankook University

Lab6

- Practice to write an **application launcher program** using **Command pattern**.
 - `ProgramLancherCommandsImporter` loads command data from a json file and returns a `HashMap<String, ProgramLauncherCommand>`
 - `IProgramLauncherCommand` interface has `execute` and `undo`.
 - `ProgramLauncherCommand` implements `IProgramLauncherCommand`, storing program path(`executable`) and icon path(`icon`).
 - `ProgramLauncherCommandInvoker` is the invoker class.
 - `ProgramLauncherCommandApp` extends `JFrame` to create a Swing application, displaying each program command as a button with an icon.

Lab6

- The `ProgramLauncherCommandApp` client use `ProgramLauncherCommandInvoker` to **set** command and then, **execute** the command.
- You can also call the command **undo**.



Lab6

```
public class ProgramLauncherCommand implements
IProgramLauncherCommand {
    private String executable;
    private String icon;
    private Process process;

    @Override
    public void execute(){
        try {
            ProcessBuilder pb = new ProcessBuilder(executable.split("
")));
            process = pb.start();
        }
        ...
    }
}
```

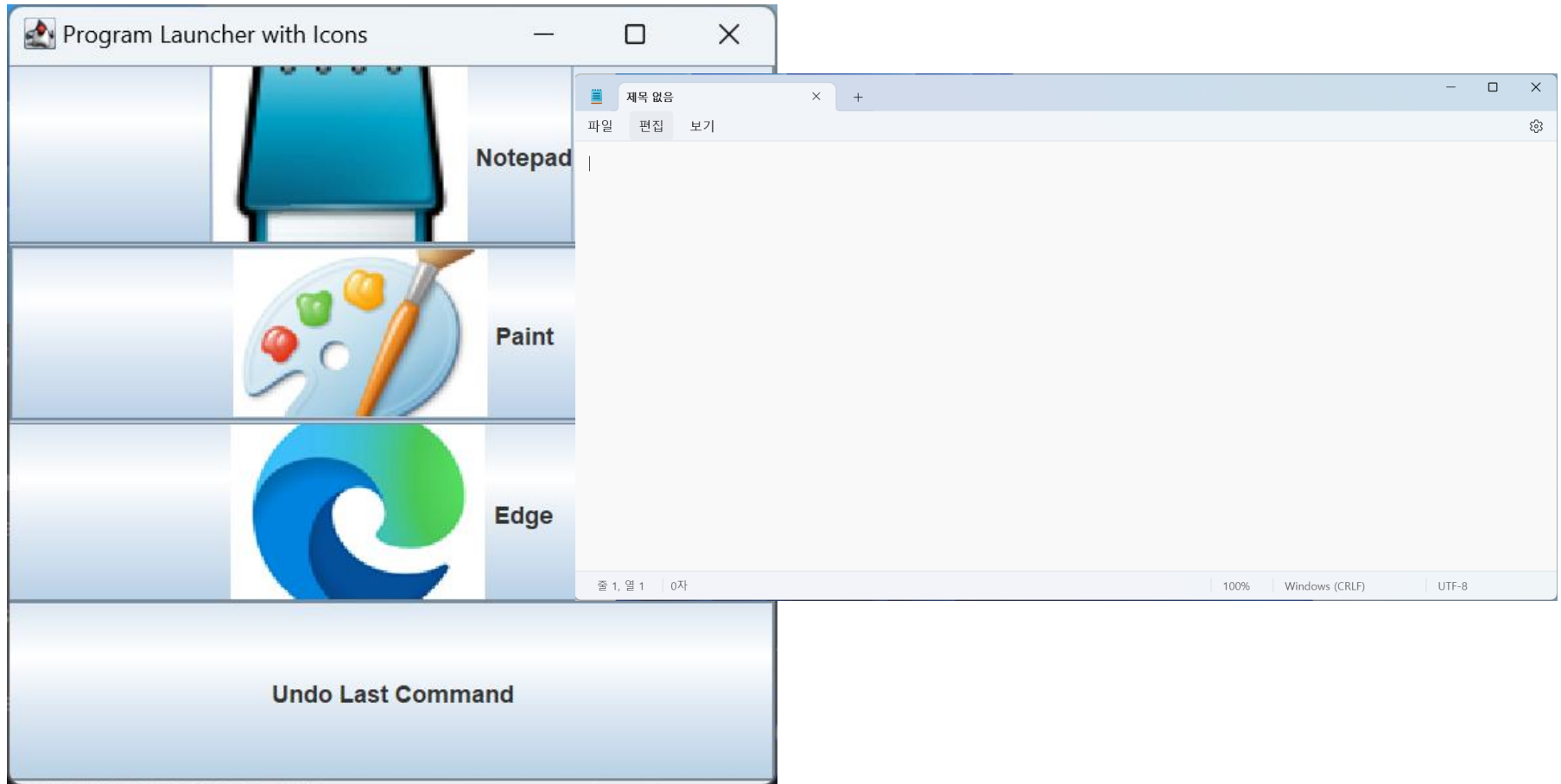
Lab6

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer:** The project structure for `HW6-LAUNCHERCOMMAND` is visible. The `lib` folder is highlighted with a yellow box, containing the file `json-simple-1.1.1.jar`.
- Editor:** The `commands.json` file is open, showing a JSON array of command definitions:

```
1 {  
2   "commands": [  
3     {  
4       "name": "Notepad",  
5       "executable": "notepad",  
6       "icon": "images/Notepad-Icon.png"  
7     },  
8     {  
9       "name": "Edge",  
10      "executable": "cmd /c start msedge",  
11      "icon": "images/MS-Edge-Icon.png"  
12     },  
13     {  
14      "name": "Paint",  
15      "executable": "mspaint",  
16      "icon": "images/MS-Paint-Icon.png"  
17     }  
18   ]  
19 }
```
- Callout:** A yellow callout box with the text `commands.json` points to the file name in the Explorer.
- Status Bar:** The bottom status bar shows `Ln 19, Col 4`, `Spaces: 2`, `UTF-8`, `CRLF`, and `{ } JSON`.

Lab6



Lab6

```
public class ProgramLauncherCommandApp extends JFrame {
    private ProgramLauncherCommandInvoker launcher = new
ProgramLauncherCommandInvoker();
    private Map<String, ProgramLauncherCommand> commandsMap;

    public ProgramLauncherCommandApp() {
        setTitle("Program Launcher with Icons");
        setSize(400, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(0, 1));

        // Load commands from JSON
        commandsMap =
ProgramLauncherCommandImporter.loadCommandsFromJson("commands.json");
    }
}
```

Lab6

```
// Create buttons dynamically based on commandsMap
for (Map.Entry<String, ProgramLauncherCommand> entry :
commandsMap.entrySet()) {
    JButton button = createButtonWithIcon(entry.getKey(),
entry.getValue()); // Trigger execute in invoker
    // Add button to the GUI
    add(button);
}
// Add Undo button at the bottom
JButton undoButton = new JButton("Undo Last Command");
undoButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        launcher.undoLastCommand(); // Trigger undo in invoker
    }
});
add(undoButton);
}
```


Submit to e-learning

- Add your code (e.g., additional method, class, routine, etc) in the Lab6 assignment.
- Submit the Lab6 assignment (JAVA24-2-Lab6-ID-name.zip including the report) to e-learning (due by 11/12).