

Assignment 2 – (CSCI4731) Software Design and Patterns – Spring 2020

This assignment contains a single task with two parts.

- The first part is to design a class diagram solution to the given problem. To generate the UML class diagram, you are allowed to use any tool convenient to you.
- The second part is to implement the given design structure in Java language using Object- Oriented Programming techniques. Make sure, that your code is neat, clean and thoroughly commented. Try to use meaningful variable names. This assignment requires you to use *the Adapter, the Singleton and the Factory* patterns.

All submissions should be uploaded to Blackboard as a single zip file.

Deadline: March 25th, 23:30

User Interface

Description

The problem is to design a modern User Interface (UI) that supports multiple design styles (motif or "look and feel"). The core of the UI is the single Window Manager (WM), which is responsible for the management of the UI items, such as buttons, text boxes and edit boxes. Different design styles are supported by the system: simplistic design style, high detailed design style and futuristic design style. WM should be initialized only by specifying the design style. Each UI item looks differently when the design style of the system is different. Also, each item has a value, which is displayed on the item itself (i.e. value of a button is the text displayed on the button).

The structure of a UI is described in a special config file. This file contains the structure of a UI as a list of UI items, their values and their coordinates. Example of a config file:

Button, Click on me, X: 250, Y: 300

EditBox, Some text to edit, X: 250, Y: 350

...

A Config Manager class is responsible for loading config files. It has methods *nextItem()* – returns the next item in the list and *hasMoreItems()* - returns true if iterating through the list is not over yet. Window Manager has a method *loadUI(ConfigManager config)*, which goes through the *config* step by step and *displays* all UI items.

To extend the functionality of the system, it must also be possible to load the configuration from an XML file. Your application should adapt one of the native java XML parsing methods (DOM Parser/Builder, SAX Parser, Java XML I/O) and make it compatible with the **ConfigManager** interface. Example of an XML file:

```
<Button value=\Click on me" X=\250" Y=\300" />
<EditBox value=\Some text to edit" X="250" Y=\350" />
...
```

Finally, to test the system, load UI elements from a config file, then from an XML file and then create several items during the runtime programmatically.

Task 1 - 40 points

Generate UML class diagram for this problem. Apply the Singleton, the Factory and the Adapter patterns. Upload only final image of the UML diagram. Accepted formats: BMP, JPG or PNG.

Task 2 - 60 points

Implement the solution in Java language using the IDE of your preference (both Eclipse and IntelliJ are welcome). All interactions with the program should be through console. Before uploading, export your project (File –> Export ...) into a zip file and don't just copy the source files from your solution. Along with the solution, in a separate folder test, include the config files (regular and XML). **Note: all sources codes will be checked for plagiarism. Do not try to cheat!**