Lab Task: Three

This assignment contains a single task with two parts.

- The task of the first part is to design a class diagram solution to the given problem. To generate the UML class diagram, it is recommended to use the ArgoUML or StarUML design tool, but the choice is yours.

- The task of 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 Abstract Factory** patterns.

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(for example, we can change the color of the elements), high detailed design style(for example, we can change the color and text size of the elements). 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 - UML Diagram (40 points)

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

Task 2 - Java Program (60 points)

All interactions with the program should be through console. However, the final output will be a graphical interface.

Note:

- **All sources codes will be checked for plagiarism. Do not try to cheat!**
- **If you want then you can use any other language also.**

.

Assignment Submission Instructions

1. Class diagram name should follow the specific format: **ClassDiagram33.png**.
2. After that send a docs file which will describe your assumptions.
3. Each of the file should be attach separately. Moreover, you have to append your roll no in each of the attach file name. For example, if one of your file name is **Solution.java** then you have submit the file after modifying the file name to **Solution33.java**
4. Finally attach each of the files separately in the assignment and then submit. **Don't zip the all the files. Attach all the file separately.**

Don't forgot to turn in the assign and finally submit the assignment.