SOFTWARE DESIGN PATTERN

Assignment 03

Assumptions of Project

Implementation of Abstract Factory, Singleton and Adapter Pattern

---

Submitted by:
Fabliha Anber
Roll- RH-56
Registration number: 2017-415-019

Submitted to:

Dr. Md. Samiullah
Assistant Professor,
Department of Computer Science and Engineering

University of Dhaka

Assumptions relating to abstract factory pattern implementation

The abstraction factory pattern is used in the implementation of the GUI functions. Interfaces are declared for each of the graphical user interface family- edit box (for editing text), text box and button. The simple and high-detailed variants (simple edit box, high detailed edit box, simple button, high detailed button, simple text box, high detailed text box) of this GUI objects are defined in classes which implements the interfaces. The Abstract Factory - *DesignFactory* is the interface which contains the list of creation methods of all GUI objects that are part of the GUI family(*EditBox*, *TextBox* and *Button*). Separate factory classes (*SimpleFactory* and *HighDetailedFactory*) and these factory classes returns GUI objects of that particular kind. The Client (*GUIApplication*) can work with any of these factory/product variant and the factory type is initialized in *WindowManager56* class.

When *SimpleFactory* type is initialized, the GUI elements changes color every time the application runs. And in *HighDetailed Factory* type, The GUI elements changes color as well as text size with every time the application runs. The size and color are chosen randomly for each element every time.

Assumptions relating to singleton pattern implementation

The *ConfigManager* class deals with database of the files. So, this class follows Singleton design pattern and only one instance of *ConfigManager* class is available in *WindowManager56* class. So, the data files in *ConfigManager* class cannot be potentially overwritten by some other code and returns single instance *config* of class *ConfigManager.* The *config* instance *loadUI(ConfigManager config)* of *WindowManager56* class is a single instance.

Assumptions relating to Adapter factory pattern implementation

The default file system is in text file handles by *ConfigManager* class. But to make incompatible file types with xml extension to collaborate, an adapter class *XmlFileAdapter* is used to convert the xml file to a text file. The default file is defined by *ConfigFile* class and other classes must also be able to collaborate with this class. The Adapter class which is the *XmlFileAdapter* inherits the default *ConfigFile* class and wraps the *XmlFile* object and converts it into a compatible object which is this case is the text file.

Conclusion

When the program is executed, the design style option is taken from the user through console and the design style type is initialized by *WIndowManager56* class at first. The GUI items are loaded from text and xml files and the final output is displayed in the screen.