CS4015 Term Project

Michael Spencer, 3532671

Design Patterns Used

The following design patterns are implemented in this project:

1. Factory: Uses the FactoryContact.java class to create Contact objects for other classes to use.
2. Singleton: Used with the ContactFactory.java class to ensure that only a single instance of FactoryContact can be created.
3. Facade: Used to hide the inner workings of the system from the user and the UserInterface.java class. Uses the SystemFacade.java class to accomplish this.
4. Decorator: The decorator design pattern is used to add optional “decorations” to the printed list of Contact objects. If the user selects to add the decorator, the contact list will be printed out with optional decorative features.
5. Strategy: Strategy.java class defines an interface that is implemented by OperationAdd.java. OperationAdd contains the code to add a Contact into the ContactList. Context.java is used to create a new instance of OperationAdd.
6. Memento: Used to support undo operations in the user interface layer
7. Command: Also used to support undo operations in the user interface layer.
8. Iterator: Used to print out the contact list.
9. Adapter: The Contact.java class implements List and uses the GoF list interface.
10. Visitor: Used to help search the contactList for searching for a contact
11. Flyweight: Used in the FactoryContact.java class to keep the memory usage of Contact objects as small as possible.
12. Builder: Used to build the contact list by adding objects to it piece by piece
13. Prototype: Uses the PrototypeDisplay.java class to create Command objects for other classes to use.

Software Architectural Patterns Used

The following software architecture patterns are implemented in this project:

1. Layered architectural Pattern: The design contains three layers: UI layer, System layer and the Data Management layer.