Research Project- Project Report

Nicholas Jumpp- 1501298

Kimone V. Roper Ist -14

University of Technology Jamaica

S.O.L.I.D. Principles

“Object-Oriented type of programming brought a new design to software development. This enables developers to combine data with the same purpose/functionality in one class to deal with the sole purpose there, regardless of the entire application.” (Nnamdi, 2018).

The five principles are as follows:

* S – Single Responsibility Principle (SRP)
* O – Open Closed Principle (OCP)
* L – Liskov Substitution Principle (LSP)
* I – Interface Segregation Principle (ISP)
* D – Dependency Inversion Principle (DIP)

The Single Responsibility Principle (SRP) states that “A class should have one, and only one, reason to change.”(Barber, 2015)

The Open Closed Principle (OCP) states that “You should be able to extend a classes behavior, without modifying.”(Barber, 2015)

The Liskov Substitution Principle (LSP) states that “Derived classes must be substitutable for their base classes.”( Barber, 2015)

The Interface Segregation Principle (ISP) states “Make fine grained interfaces that are client specific.”( Barber, 2015)

The Dependency Inversion Principle (DIP) states “Depend on abstraction, not on concretions.”( Barber, 2015)

MVC Pattern

MVC Pattern stands for Model-View-Controller Pattern. This pattern is used to separate application's concerns. (Tutorialspoint.com)

* Model - Model represents an object carrying data. It can also have logic to update controller if its data changes.
* View - View represents the visualization of the data that model contains.
* Controller - Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate.

Repository Pattern

The Repository Pattern restricts us to work directly with the data in the application and creates new layers for database operations, business logic, and the application’s UI.

Singleton Pattern

“The singleton pattern is a design pattern that restricts the instantiation of a class to one object” (Singleton, 2018).

Factory Pattern

“A Factory Pattern says that just define an interface or abstract class for creating an object but let the subclasses decide which class to instantiate” (JavaTPoint, n.d.).

References

Barber, A. P. (2015, August 06). SOLID Principles - Five Principles of Object-Oriented Programming and Design. Retrieved from <https://alanbarber.com/post/solid-principles-five-principles-of-objectoriented-pro> gramming-and-design/

JavaTPoint. (n.d.). Factory Method Design Pattern - Javatpoint. Retrieved from https://www.javatpoint.com/factory-method-design-pattern

Nnamdi, C. (2018, October 09). SOLID Principles every Developer Should Know. Retrieved from <https://blog.bitsrc.io/solid-principles-every-developer-should-know-b3bfa96bb68> 8

Singleton Design Pattern | Implementation. (2018, September 10). Retrieved from https://www.geeksforgeeks.org/singleton-design-pattern/

Tutorialspoint.com. (n.d.). Design Patterns MVC Pattern. Retrieved from https://www.tutorialspoint.com/design\_pattern/mvc\_pattern.htm