**Singleton**

Motivation

Sometimes it's important to have only one instance for a class. The singleton pattern is one of the simplest design patterns: it involves only one class which is responsible to instantiate itself, to make sure it creates not more than one instance. In the same time it provides a global point of access to that instance. In this case the same instance can be used from everywhere, being impossible to invoke directly the constructor each time.

Intent

-Ensure that only one instance of a class is created.

-Provide a global point of access to the object.

Implementation

The implementation involves a static member in the "Singleton" class, a private constructor and a static public method that returns a reference to the static member. The Singleton Pattern defines a GetInstance operation which exposes the unique instance which is accessed by the clients. GetInstance() is responsible for creating its class unique instance in case it is not created yet and to return that instance.