UNIVERSITÀ
DEGLI STUDI
DE L'AQUILA

disim

Mobile Programming Laboratory

ANDROID
Design Pattern

Teachers

Ing. Tarquini Francesco, Ph.D

Ph.D in Computer Science Engineering

francesco.tarquini@univaq.it

Ing. D'Errico Leonardo

Ph.D Student in Computer Science Engineering

leonardo.derrico@graduate.univaq.it

Teaching Materials

Available on MOODLE platform

http://www.didattica.univaq.it

Google Drive Repository

https://drive.google.com/drive/folders/1ISqZfn0i9Ub3eWNXbvW00rd0hD9ya8OL?usp=sharing

Topics

- Design Pattern
  - Model View Controller (MVC)
  - Singleton

Design Pattern

Design Patterns are reusable solutions to commonly occurring problems (in the context of software design).

Design patterns were started as best practices that were applied again and again to similar problems encountered in different contexts.

They become popular after they were collected, in a formalized form, in the Gang Of Four book in 1994.

Design Pattern - Singleton

Singleton pattern is one of the simplest design patterns in Java.

This pattern involves a single class which is responsible to create an object while making sure that only single object gets created. This class provides a way to access its only object which can be accessed directly without need to instantiate the object of the class.

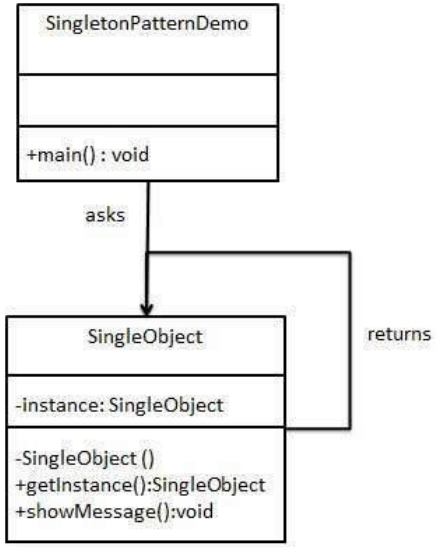
```
public class JavaClass {

   private static JavaClass instance = null;

   private JavaClass() { }

   public static JavaClass get() {
      if(instance == null) instance = new JavaClass();
      return instance;
   }

   public void someMethod(){
      // Do Something
   }
}
```



Design Pattern - MVC

MVC Pattern stands for Model-View-Controller Pattern.

This pattern is used to separate application's concerns.

Model: Model represents an object or JAVA POJO 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.