Design Pattern - Singleton Pattern

Singleton pattern is one of the simplest design patterns in Java. This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object.

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.

这个模式涉及到一个单独的类，它负责创建一个对象，并且同时确保只有唯一的一个对象能被创建。这个类提供唯一的方法直接访问这个对象而无需实例化对象。

Implementation

We're going to create a *SingleObject* class. *SingleObject* class have its constructor as private and have a static instance of itself.

*SingleObject* class provides a static method to get its static instance to outside world. *SingletonPatternDemo*, our demo class will use *SingleObject* class to get a *SingleObject* object.

我们创建一个SingleObject类。SingeObject类有一个私有的构造器和一个它自身的静态实例对象。

SingleObject类提供一个静态的方法，外部可以是使用它来获得SingleObject的静态实例。SingletonPatternDemo, 我们的demo类将通过SingleObject类来获得SingleObject 对象。



Step 1

Create a Singleton Class.

创建一个Singleton 类。

*SingleObject.java*

public class SingleObject {

//create an object of SingleObject

private static SingleObject instance = new SingleObject();

//make the constructor private so that this class cannot be

//instantiated

private SingleObject(){}

//Get the only object available

public static SingleObject getInstance(){

return instance;

}

public void showMessage(){

System.out.println("Hello World!");

}

}

Step 2

Get the only object from the singleton class.

从单例类获得唯一的对象。

*SingletonPatternDemo.java*

public class SingletonPatternDemo {

public static void main(String[] args) {

//illegal construct

//Compile Time Error: The constructor SingleObject() is not visible

//SingleObject object = new SingleObject();

//Get the only object available

SingleObject object = SingleObject.getInstance();

//show the message

object.showMessage();

}

}

Step 3

Verify the output.

校验输出。

Hello World!