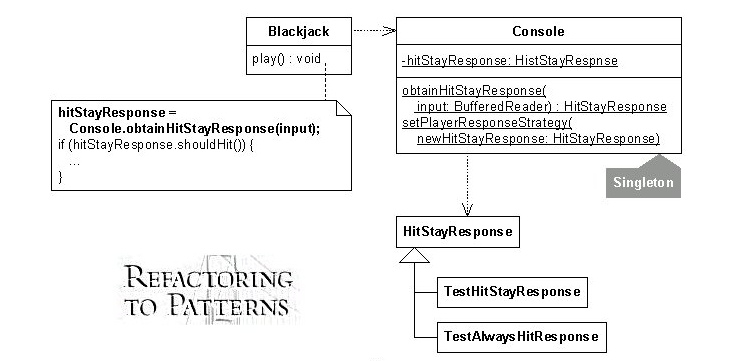
Title: Refactoring: Move the Singleton's features to a class that stores and  
provides access to the object. Delete the Singleton.

Code needs access to an object but doesn’t need a global point of access to it.

Motivation to refactor:

Using a Singleton is not always necessary when some code needs access to an object. Singletons should only be used when a class has only one instance and can provide a global point of access to it. It is usually simpler to pass object resources as a reference to objects that need it. This ensures that classes are easier to read as similar functionality is grouped together, rather than having to reference the singleton every time.



**Sample of code:**

Java program implementing Singleton class

// with getInstance() method

class Singleton

{

    // static variable single\_instance of type Singleton

    private static Singleton single\_instance = null;

    // variable of type String

    public String s;

    // private constructor restricted to this class itself

    private Singleton()

    {

        s = "Hello I am a string part of Singleton class";

    }

    // static method to create instance of Singleton class

    public static Singleton getInstance()

    {

        if (single\_instance == null)

            single\_instance = new Singleton();

        return single\_instance;

    }

}

// Driver Class

class Main

{

    public static void main(String args[])

    {

        // instantiating Singleton class with variable x

        Singleton x = Singleton.getInstance();

        // instantiating Singleton class with variable y

        Singleton y = Singleton.getInstance();

        // instantiating Singleton class with variable z

        Singleton z = Singleton.getInstance();

        // changing variable of instance x

        x.s = (x.s).toUpperCase();

        System.out.println("String from x is " + x.s);

        System.out.println("String from y is " + y.s);

        System.out.println("String from z is " + z.s);

        System.out.println("\n");

        // changing variable of instance z

        z.s = (z.s).toLowerCase();

        System.out.println("String from x is " + x.s);

        System.out.println("String from y is " + y.s);

        System.out.println("String from z is " + z.s);

    }

}