Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TEAMS

Create Team

Null Object Design Pattern in LinkedList

Ask Question

I am trying to Implement a doubly linked with null objects at the beginning and end of the list using null object design pattern. So an empty list will contain two null objects. So I wrote this code Does this follow null object design pattern? If not how can I achieve that. ANy suggestions will be appreciated.

Updated Code-

```
// Creating a doubly linked list.
        doubleLinkedList = new DoubleLinkedList();

    class DoubleLinkedList {

        private NewLink firstNode;
        private NewLink lastNode;
        private NewLink rootNode;



        public DoubleLinkedList() {

//So this satisfies my question that I have asked meaning null objects at
beginning and last node or something else I have to do.
            firstNode = NewLink.NULL_NODE;
            lastNode  = NewLink.NULL_NODE;

        }

    }



    class NewLink {

        public String  data;
        public NewLink nextPointer;
        public NewLink previousPointer;


public static final NewLink NULL_NODE = new NewLink();
```

```
        public NewLink(String id) {

            data = id;

        }

public NewLink() {

    }
        // Overriding toString method to return the actual data of the node
        public String toString() {

            return "{" + data + "} ";

        }
    }
```

java design-patterns null-object-pattern

edited Jul 17 '15 at 15:51
Brad Larson ♦
160k 40 359 534

asked Feb 19 '12 at 10:46
AKIWEB
5,917 45 139 226

Is this your complete code ? – Sajan Chandran Feb 19 '12 at 10:54

I have updated the code. – AKIWEB Feb 19 '12 at 11:05

3 Answers

```
public static final NewLink NULL_NODE = new NewLink();
```

must be in `NewLink` class

so

```
firstNode = NewLink.NULL_NODE;
secondNode = NewLink.NULL_NODE;
```

also you can make all methods from `NewLink` - abstract
and make two nested classes: for NULL objects and for not NULL object.
It's can be very helpful in difficult situations

edited Feb 19 '12 at 11:18

answered Feb 19 '12 at 11:07
Ilya
22.2k 17 85 138

i updated the code as per your suggestions. let me know if it is right now? – AKIWEB Feb 19 '12 at 11:10

It is fine now:) – Petar Minchev Feb 19 '12 at 11:13

No, your code doesn't implement `Null Object Design Pattern` . Its essence is not to use `null` but to create an object which will represent the `null` .

For example:

```
public static final NewLink NULL_NODE = new NewLink();
```

And then:

```
firstNode = NULL_NODE;
lastNode  = NULL_NODE;
```

answered Feb 19 '12 at 10:52
Petar Minchev
38.4k 10 81 110

I have updated the code. Let me know if it is right? – AKIWEB Feb 19 '12 at 11:04

Yep, that's the basic idea. – Petar Minchev Feb 19 '12 at 11:05

@Nevzz03 - See `user1143825` answer for moving the definition to the other class. It is more logical. – Petar Minchev
Feb 19 '12 at 11:09

We use Null object design pattern if we want to assign some default behaviors (or prevent some behaviors to happen as a default behavior) For example using Null object pattern we can replace this code :

```
if(myObj!=null)
myObj.DoSomething();
else
DoSomethingElse();
```

wtih this one:

```
myObj.DoSomething() //assuming that myObj can be a Null object (not a null
reference) that has implemented DoSomethingElse when you ask it to DoSomething()
```

Thus a Null Object design pattern actually uses a default object reference (not a null reference)

answered Feb 19 '12 at 11:10

Beatles1692
3,670 22 51