

观察者模式

模式定义

观察者模式定义了对象之间的一对多的依赖, 这样一来, 当一个对象改变状态时,他的所有的依赖者都会收到通知并自动更新

问题需求描述

模拟一个报纸订阅发送和b站和b站用户去b站get更新的行为

Code

Push模式, 报纸

Newspaper.java

```
package Observer;

import java.util.Iterator;
import java.util.List;

public class NewsPaper {
    List<NewsPaperSubscriber> list;

    public void addSubscriber(NewsPaperSubscriber subscriber) {
        list.add(subscriber);
    }

    public void ereaseSubscriber(NewsPaperSubscriber subscriber) {
        Iterator<NewsPaperSubscriber> iterator = list.iterator();
        while (iterator.hasNext()) {
            NewsPaperSubscriber subscriber1 = iterator.next();
            if (subscriber.equals(subscriber1)) {
                iterator.remove();
                return;
            }
        }
    }

    public void sendNewspaper() {
        for (NewsPaperSubscriber subscriber : list) {
            subscriber.getNewspaper();
        }
    }
}
```

NewsPaperSubscriber.java

```
package Observer;

public class NewsPaperSubscriber {
    private int index;
    private static int next = 0;

    public NewsPaperSubscriber() {
        index = next++;
    }

    @Override
    public boolean equals(Object obj) {
        if (obj == this) return true;
        else if (obj instanceof NewsPaperSubscriber) {
            return this.index == ((NewsPaperSubscriber) obj).index;
        } else return false;
    }

    void getNewspaper() {
        System.out.println("Num: " + index + "get message!!");
    }
}
```

pull 模式

Bilibili.java

```
package Observer;

public class Bilibili {
    private String message;

    public String getMessage() {
        return message;
    }

    public void refreshMessage(String s) {
        message = s;
    }
}
```

BilibiliUser.java

```
package Observer;

public class BilibiliUser {
    private Bilibili bilibili;

    BilibiliUser(Bilibili bilibili) {
        this.bilibili = bilibili;
    }

    void getMessage() {
        System.out.println("I get message: " + bilibili.getMessage());
    }
}
```

运行结果

Main.java``java package Observer;

```
public class Main { public static void main(String[] args) { NewsPaper newsPaper = new NewsPaper(); newsPaper.addSubscriber(new NewsPaperSubscriber());
newsPaper.addSubscriber(new NewsPaperSubscriber()); newsPaper.addSubscriber(new NewsPaperSubscriber()); newsPaper.sendNewspaper(); Bilibili bilibili = new
Bilibili(); BilibiliUser user = new BilibiliUser(bilibili); BilibiliUser user1 = new BilibiliUser(bilibili); BilibiliUser user2 = new BilibiliUser(bilibili);
bilibili.refreshMessage("23333"); user.getMessage(); user1.getMessage(); user2.getMessage(); bilibili.refreshMessage("12222"); user.getMessage(); user1.getMessage();
user2.getMessage(); } }``
```

out

```
Num: 0get message!!
Num: 1get message!!
Num: 2get message!!
I get message: 23333
I get message: 23333
I get message: 23333
I get message: 12222
I get message: 12222
I get message: 12222
```