

概念
代码
原始类
备忘录类
测试类
总结

概念

在用文本编辑器的时候，如果不小心删掉了某一句话，可以通过撤销来进行恢复到没有删除之前的状态。有些文本编辑器甚至可以撤销很多模式就是意图在不破坏封装的情况下，记录下来对象的状态，并且可以进行恢复

代码

原始类

```
1 public class Note {
2
3     private String txt;
4
5     public String getTxt() {
6         return txt;
7     }
8
9     public void setTxt(String txt) {
10         this.txt = txt;
11     }
12 }
```

备忘录类

```
1 public class NoteHistory {
2     private Stack<Note> stack = new Stack<>();
3
4     public Note getPrev(){
5         return stack.pop();
6     }
7
8     public void push(Note note){
9         Note note1 = new Note();
10         note1.setTxt(note.getTxt());
11         stack.push(note1);
12     }
13
14     public int size(){
15         return stack.size();
16     }
17 }
```

测试类

```
1 public class Main {
2
3
4     public static void main(String[] args) {
5         Note note = new Note();
6
7         NoteHistory history = new NoteHistory();
8
9         note.setTxt("1");
10        history.push(note);
11
12        note.setTxt("2");
13        history.push(note);
14
15        note.setTxt("3");
16        history.push(note);
17
18        note.setTxt("4");
19        history.push(note);
20
21        note.setTxt("5");
22        history.push(note);
23
24        System.out.println(history.getPrev().getTxt());
25        System.out.println(history.getPrev().getTxt());
26        System.out.println(history.getPrev().getTxt());
27        System.out.println(history.getPrev().getTxt());
28        System.out.println(history.getPrev().getTxt());
29    }
30 }
```

## 总结

### 优点：

- 1、给用户提供了一种可以恢复状态的机制，可以使用户能够比较方便地回到某个历史的状态。
- 2、实现了信息的封装，使得用户不需要关心状态的保存细节。

### 缺点：

消耗资源。如果类的成员变量过多，或者记录的长度过大，势必会占用比较大的资源，而且每一次保存都会消耗一定的内存（可以使用享元模式+