

[首页](#) [资讯](#) [精华](#) [论坛](#) [问答](#) [博客](#) [专栏](#) [群组](#) [更多 ▼](#)

[您还未登录！](#) [登录](#) [注册](#)

ericliu1986

- [博客](#)
- [微博](#)
- [相册](#)
- [收藏](#)
- [留言](#)
- [关于我](#)

java事件机制

博客分类：

- [J2SE](#)

[Java](#)

java事件机制包括三个部分：事件、事件监听器、事件源。

1、事件。一般继承自java.util.EventObject类，封装了事件源对象及跟事件相关的信息。

com.javaedu.event.CusEvent类

Java代码 

```
1. package com.javaedu.event;
2.
3. import java.util.EventObject;
4.
5. /**
6.  * 事件类,用于封装事件源及一些与事件相关的参数.
7.  * @author Eric
8.  */
9. public class CusEvent extends EventObject {
10.     private static final long serialVersionUID = 1L;
11.     private Object source;//事件源
12.
13.     public CusEvent(Object source){
14.         super(source);
15.         this.source = source;
16.     }
17.
18.     public Object getSource() {
19.         return source;
20.     }
```

```
21.  
22. public void setSource(Object source) {  
23.     this.source = source;  
24. }  
25. }
```

2、事件监听器。实现java.util.EventListener接口,注册在事件源上,当事件源的属性或状态改变时,取得相应的监听器调用其内部的回调方法。

com.javaedu.event.CusEventListener类

Java代码 ☆

```
1. package com.javaedu.event;  
2.  
3. import java.util.EventListener;  
4.  
5. /**  
6.  * 事件监听器，实现java.util.EventListener接口。定义回调方法，将你想要做的事  
7.  * 放到这个方法下,因为事件源发生相应的事件时会调用这个方法。  
8.  * @author Eric  
9.  */  
10. public class CusEventListener implements EventListener {  
11.  
12.     //事件发生后的回调方法  
13.     public void fireCusEvent(CusEvent e){  
14.         EventSourceObjecteObject = (EventSourceObject)e.getSource();  
15.         System.out.println("My name has been changed!");  
16.         System.out.println("I got a new name,named \""+eObject.getName()+"\"");    }  
17. }
```

3、事件源。事件发生的地方，由于事件源的某项属性或状态发生了改变(比如BUTTON被单击、TEXTBOX的值发生改变等等)导致某项事件发生。换句话说就是生成了相应的事件对象。因为事件监听器要注册在事件源上,所以事件源类中应该要有盛装监听器的容器(List,Set等等)。

com.javaedu.event.EventSourceObject类

Java代码 ☆

```
1. package com.javaedu.event;  
2.  
3. import java.util.HashSet;  
4. import java.util.Iterator;  
5. import java.util.Set;  
6.  
7. /**  
8.  * 事件源。  
9.  * @author Eric  
10.  */
```

```
11. public class EventSourceObject {
12.     private String name;
13.     //监听器容器
14.     private Set<CusEventListener> listener;
15.     public EventSourceObject(){
16.         this.listener = new HashSet<CusEventListener>();
17.         this.name = "defaultname";
18.     }
19.     //给事件源注册监听器
20.     public void addCusListener(CusEventListener cel){
21.         this.listener.add(cel);
22.     }
23.     //当事件发生时,通知注册在该事件源上的所有监听器做出相应的反应（调用回调方法）
24.     protected void notifies(){
25.         CusEventListener cel = null;
26.         Iterator<CusEventListener> iterator = this.listener.iterator();
27.         while(iterator.hasNext()){
28.             cel = iterator.next();
29.             cel.fireCusEvent(new CusEvent(this));
30.         }
31.     }
32.     public String getName() {
33.         return name;
34.     }
35.     //模拟事件触发器，当成员变量name的值发生变化时，触发事件。
36.     public void setName(String name) {
37.         if(!this.name.equals(name)){
38.             this.name = name;
39.             notifies();
40.         }
41.     }
42. }
```

下面是主方法类

com.javaedu.event.MainTest类、

Java代码 

```
1. package com.javaedu.event;
2.
3. public class MainTest {
4.
5.     /**
6.      * @param args
7.      */
8.     public static void main(String[] args) {
9.         EventSourceObject object = new EventSourceObject();
10.        //注册监听器
11.        object.addCusListener(new CusEventListener(){
12.            @Override
13.            public void fireCusEvent(CusEvent e) {
```

```
14.         super.fireCusEvent(e);
15.     }
16. });
17.     //触发事件
18.     object.setName("eric");
19. }
20. }
```

分享到： 

[由Tomcat下的JNDI数据源引出的.....](#)

- 2010-03-31 23:17
- 浏览 27243
- [评论\(2\)](#)
- 分类: [编程语言](#)
- [相关推荐](#)

参考知识库



[Java SE知识库](#) 27003 关注 / 578 收录



[Java 知识库](#) 28822 关注 / 3746 收录



[Java EE知识库](#) 19534 关注 / 1408 收录



[JavaScript知识库](#) 15554 关注 / 1517 收录



[jQuery知识库](#) 9530 关注 / 948 收录



[AngularJS知识库](#) 5077 关注 / 586 收录

评论

2 楼 [killeraction](#) 2015-11-19
观察者模式
1 楼 [x_h_j123](#) 2012-08-31
👍

发表评论



[您还没有登录,请您登录后再发表评论](#)



ericliu1986

- 浏览: 88010 次
- 性别: ♂
- 来自: 天津
- 我现在离线

最近访客

[更多访客>>](#)



[raymond.chen](#)



[df20070608](#)



[快乐的虫子](#)



[声动天下](#)

文章分类

- [全部博客 \(11\)](#)
- [Android \(5\)](#)
- [J2SE \(2\)](#)
- [J2EE \(3\)](#)
- [C \(1\)](#)

社区版块

- [我的资讯](#) (0)

- [我的论坛](#) (1)
- [我的问答](#) (0)

存档分类

- [2010-12](#) (5)
- [2010-10](#) (2)
- [2010-06](#) (3)
- [更多存档...](#)

最新评论

- [killeraction](#) : 观察者模式
[java事件机制](#)
- [banxi1988](#) : 此博客对我有用 !
[Application did not close the cursor or database object that was opened here](#)
- [x_h_j123](#) : ...
[java事件机制](#)
- [txin0814](#) : yangxue00 写道弱弱的问一下，红色字体部分是你从log ...
[Application did not close the cursor or database object that was opened here](#)
- [yangxue00](#) : 弱弱的问一下，红色字体部分是你从logcat中复制过来的吗？ ...
[Application did not close the cursor or database object that was opened here](#)

声明：ITeye文章版权属于作者，受法律保护。没有作者书面许可不得转载。若作者同意转载，必须以超链接形式标明文章原始出处和作者。

© 2003-2017 ITeye.com. All rights reserved. [京ICP证110151号 京公网安备110105010620]