

Java Career

永久域名 <http://gaojiewyh.javaeye.com>

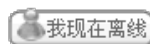


gaojiewyh

浏览: 46132 次

性别:

来自: 济南



[详细资料](#) [留言簿](#)

搜索本博客

最近访客 [>>更多访客](#)



[azraelzxc](#)



[tajear](#)



[taijh999](#)



[luoshg](#)

博客分类

- [全部博客 \(165\)](#)
- [个人思考 \(1\)](#)
- [数据库 \(4\)](#)
- [java基础 \(25\)](#)
- [软件工程 \(2\)](#)
- [flex \(0\)](#)
- [zk开源框架 \(14\)](#)
- [设计模式 \(25\)](#)
- [javascript \(4\)](#)
- [css \(5\)](#)
- [UML \(2\)](#)
- [CMMI软件需求 \(3\)](#)
- [CMMI软件设计 \(2\)](#)
- [CMMI软件实现 \(1\)](#)

2009-06-26

[问题汇总](#) | [java中的clone机制及序列化](#)

0顶

0踩

原型模式

文章分类:[Java编程](#) 关键字: clone, 序列化

前面java基础中介绍了clone和序列化, 结合clone和序列化的知识, 今天学习下原型模式。

定义: 用原型实例指定创建对象的种类, 并且通过拷贝这些原型创建新的对象.Prototype模式允许一个对象再创建另外一个可定制的对象, 根本无需知道任何如何创建的细节,

工作原理: 通过将一个原型对象传给那个要发动创建的对象, 这个要发动创建的对象通过请求原型对象拷贝它们自己来实施创建。

适用性: 当要实例化的类是在运行时刻指定时, 例如, 通过动态装载;

或者为了避免创建一个与产品类层次平行的工厂类层次时;

或者当一个类的实例只能有几个不同状态组合中的一种时。

建立相应数目的原型并克隆它们可能比每次用合适的状态手工实例化该类更方便一些。

因为我们编程要面向接口编程, 所以这里我们为原型模式的具体实现类抽取出一个接口。

1、采用序列化的方式实现

Java代码

```
1.  /**
2.   *
3.   * @功能 : 工作经历
4.   * @创建人 gao_jie
5.   * @创建日期 May 11, 2009
6.   * @版本 1.0
7.   *
8.   */
9.  public class WorkExperience implements Serializable{
10.
11.      private String workdate; // 工作时间
12.      private String workcompany; // 工作的公司
13.      /**
14.       * @return
15.       */
16.      public String getWorkdate() {
17.          return workdate;
18.      }
19.
20.      /**
21.       * @param workdate
22.       */
23.      public void setWorkdate(String workdate) {
24.          this.workdate = workdate;
25.      }
26.
27.      /**
28.       * @return
29.       */
30.      public String getWorkcompany() {
31.          return workcompany;
32.      }
33.
34.      /**
35.       * @param workcompany
36.       */
37.      public void setWorkcompany(String workcompany) {
```

- [CMMI软件测试 \(1\)](#)
- [CMMI软件交付 \(0\)](#)
- [正则表达式 \(4\)](#)
- [系统日志配置 \(4\)](#)
- [应用服务器 \(1\)](#)
- [spring \(5\)](#)
- [XML \(5\)](#)
- [java web \(8\)](#)
- [Ajax \(3\)](#)
- [RichFaces \(14\)](#)
- [问题 \(1\)](#)
- [评论文章 \(2\)](#)
- [fushioncharts \(2\)](#)
- [jquery \(21\)](#)
- [Ext js学习 \(2\)](#)
- [学习心得 \(1\)](#)

我的留言簿 >>>[更多留言](#)

- 看了你的文章Richface layout ([http://gaojiewyh.ja ...](http://gaojiewyh.ja...))
-- by [hellboy](#)
- 流口水 流口水 流口水
-- by [sieben](#)
- 看你发表的文章真多!! 一定是个高手!!
-- by [lib](#)

其他分类

- [我的收藏 \(104\)](#)
- [我的论坛主题贴 \(3\)](#)
- [我的所有论坛贴 \(14\)](#)
- [我的精华良好贴 \(1\)](#)

最近加入圈子

- [JSF](#)

存档

- [2010-02 \(1\)](#)
- [2010-01 \(9\)](#)
- [2009-12 \(9\)](#)
- [更多存档...](#)

最新评论

- [DAO设计模式 \(希望大家指...](#)
基于接口的模板回调, 代码比较精典! 我以前就用这个方法

```
38.         this.workcompany = workcompany;
39.     }
40. }
41.
42. /**
43.  *
44.  * @功能 原型模式抽象接口
45.  * @创建人 gao_jie
46.  * @创建日期 Jun 27, 2009
47.  * @版本 1.0
48.  *
49.  */
50. public interface Iprototype extends Cloneable, Serializable{
51.
52.     /**
53.      * 设置用户个人信息
54.      *
55.      * @param name
56.      * @param sex
57.      */
58.     public abstract void setPersonInfo(String sex, String age);
59.
60.     /**
61.      * 设置工作经历
62.      *
63.      * @param name
64.      * @param sex
65.      */
66.     public abstract void setWorkExperience(String workdate, String company);
67.
68.     /**
69.      * 显示结果
70.      */
71.     public abstract void display();
72.
73.     /**
74.      * @return
75.      */
76.     public Object clone();
77. }
78. /**
79.  *
80.  * @功能 原型模式具体实现
81.  * @创建人 gao_jie
82.  * @创建日期 May 11, 2009
83.  * @版本 1.0
84.  *
85.  */
86. public class ConcretePrototype implements Iprototype {
87.
88.     private String name;// 姓名
89.     private String sex;// 性别
90.     private String age;// 年龄
91.     private WorkExperience Work;// 工作经历
92.
93.     /**
94.      * 构造函数
95.      *
96.      * @param name
97.      */
98.     public ConcretePrototype(String name) {
99.         this.name = name;
100.        this.Work = new WorkExperience();
101.    }
102. }
```

给项目搭过框架的！，但 ...

-- by [xiao-qiang163](#)

■ [在richfaces中搭建layout ...](#)

楼主，你的附件是加密的？密码呢？

-- by [tonyshen277](#)

■ [jQuery插件开发全解析](#)

yvonxiao 写道

用jQuery.extend的话就是扩

展，直接jQuery. ...

-- by [qesc](#)

■ [jQuery插件开发全解析](#)

用jQuery.extend的话就是扩

展，直接jQuery.foo=functio ...

-- by [yvonxiao](#)

■ [jQuery插件开发全解析](#)

受教，感谢！ 交流下看法。1.类

级别的插件 忘记了在那里看

过，增加全局函数是，jq ...

-- by [qesc](#)

评论排行榜

■ [DAO设计模式（希望大家指正）](#)

■ [jQuery插件开发全解析](#)

■ [richfaces中表格树的展现](#)

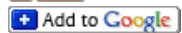
■ [模拟spring中](#)

[的ClassPathXmlApplicationCon](#)

...

■ [zk如何调用本地javascript代码](#)

[向服务器端发 ...](#)



[\[什么是RSS?\]](#)

```
103.         /* (non-Javadoc)
104.          * @see eight.prototypedeepcopy2.Prototype#setPersonInfo(java.lang.String, java.l
ang.String)
105.          */
106.         public void setPersonInfo(String sex, String age) {
107.             this.sex = sex;
108.             this.age = age;
109.         }
110.
111.         /* (non-Javadoc)
112.          * @see eight.prototypedeepcopy2.Prototype#setWorkExperience(java.lang.String, ja
va.lang.String)
113.          */
114.         public void setWorkExperience(String workdate, String company) {
115.             this.Work.setWorkdate(workdate);
116.             this.Work.setWorkcompany(company);
117.         }
118.
119.         /* (non-Javadoc)
120.          * @see eight.prototypedeepcopy2.Prototype#display()
121.          */
122.         public void display() {
123.             System.out.println("名字: " + name);
124.             System.out.println("性别: " + sex);
125.             System.out.println("年龄: " + age);
126.             System.out
127.                 .println("工作经历: " + Work.getWorkdate() + Work.getWorkco
mpany());
128.         }
129.
130.         /*
131.          * (non-Javadoc)
132.          *
133.          * @see java.lang.Object#clone()
134.          */
135.         public Object clone() {
136.             try {
137.                 ByteArrayOutputStream bos = new ByteArrayOutputStream();
138.                 ObjectOutputStream oos = new ObjectOutputStream(bos);
139.                 oos.writeObject(this);
140.                 // 将流序列化
141.                 ByteArrayInputStream bis = new ByteArrayInputStream(bos
142.                     .toByteArray());
143.                 ObjectInputStream ois = new ObjectInputStream(bis);
144.                 return ois.readObject();
145.             } catch (IOException e) {
146.                 e.printStackTrace();
147.             } catch (ClassNotFoundException e) {
148.                 e.printStackTrace();
149.             }
150.             return null;
151.         }
152.     }
153. /**
154.  *
155.  * @功能：客户端测试类
156.  * @创建人 gao_jie
157.  * @创建日期 May 11, 2009
158.  * @版本 1.0
159.  *
160.  */
161. public class Client {
162.
163.     /**
164.      * @param args
```

```

165.         */
166.         public static void main(String[] args) {
167.             Iprototype a = new ConcretePrototype("大鸟");
168.             a.setPersonInfo("男", "29");
169.             a.setWorkExperience("2009-12-19", "中创软件");
170.             Iprototype b = (ConcretePrototype)a.clone();
171.             b.setPersonInfo("女", "25");
172.             b.setWorkExperience("2008-12-13", "中间件");
173.
174.             a.display();
175.             b.display();
176.         }
177.     }
178. 客户端显示结果:
179. 名字: 大鸟
180. 性别: 男
181. 年龄: 29
182. 工作经历: 2009-12-19中创软件
183. 名字: 大鸟
184. 性别: 女
185. 年龄: 25
186. 工作经历: 2008-12-13中间件

```

2、采用深度复制方式实现

Java代码

```

1.  /**
2.   *
3.   * @功能 :工作经历
4.   * @创建人 gao_jie
5.   * @创建日期 May 11, 2009
6.   * @版本 1.0
7.   *
8.   */
9.  public class WorkExperience implements Cloneable{
10.
11.      private String workdate;// 工作时间
12.      private String workcompany;// 工作的公司
13.      /**
14.       * @return
15.       */
16.      public String getWorkdate() {
17.          return workdate;
18.      }
19.
20.      /**
21.       * @param workdate
22.       */
23.      public void setWorkdate(String workdate) {
24.          this.workdate = workdate;
25.      }
26.
27.      /**
28.       * @return
29.       */
30.      public String getWorkcompany() {
31.          return workcompany;
32.      }
33.
34.      /**
35.       * @param workcompany
36.       */
37.      public void setWorkcompany(String workcompany) {
38.          this.workcompany = workcompany;
39.      }
40.      public Object clone() throws CloneNotSupportedException {

```

```

41.         return super.clone();
42.     }
43. }
44. /**
45.  *
46.  * @功能 原型模式抽象接口
47.  * @创建人 gao_jie
48.  * @创建日期 Jun 27, 2009
49.  * @版本 1.0
50.  *
51.  */
52. public interface Iprototype extends Cloneable, Serializable {
53.
54.     /**
55.      * 设置用户个人信息
56.      *
57.      * @param name
58.      * @param sex
59.      */
60.     public abstract void setPersonInfo(String sex, String age);
61.
62.     /**
63.      * 设置工作经历
64.      *
65.      * @param name
66.      * @param sex
67.      */
68.     public abstract void setWorkExperience(String workdate, String company);
69.
70.     /**
71.      * 显示结果
72.      */
73.     public abstract void display();
74.
75.     /**
76.      * @return
77.      */
78.     public Object clone() throws CloneNotSupportedException;
79. }
80.
81. /**
82.  *
83.  * @功能 原型模式具体实现
84.  * @创建人 gao_jie
85.  * @创建日期 May 11, 2009
86.  * @版本 1.0
87.  *
88.  */
89. public class ConcretePrototype implements Iprototype {
90.
91.     private String name; // 姓名
92.     private String sex; // 性别
93.     private String age; // 年龄
94.     private WorkExperience Work; // 工作经历
95.
96.     /**
97.      * 构造函数
98.      *
99.      * @param name
100.     */
101.     public ConcretePrototype(String name) {
102.         this.name = name;
103.         this.Work = new WorkExperience();
104.     }
105.

```

```

106.      /*
107.       * (non-Javadoc)
108.       *
109.       * @see eight.prototypedeepcopy2.Prototype#setPersonInfo(java.lang.String,
110.       *      java.lang.String)
111.       */
112.      public void setPersonInfo(String sex, String age) {
113.          this.sex = sex;
114.          this.age = age;
115.      }
116.
117.      /*
118.       * (non-Javadoc)
119.       *
120.       * @see eight.prototypedeepcopy2.Prototype#setWorkExperience(java.lang.String,
121.       *      java.lang.String)
122.       */
123.      public void setWorkExperience(String workdate, String company) {
124.          this.Work.setWorkdate(workdate);
125.          this.Work.setWorkcompany(company);
126.      }
127.
128.      /**
129.       * @return
130.       */
131.      public WorkExperience getWork() {
132.          return Work;
133.      }
134.
135.      /*
136.       * (non-Javadoc)
137.       *
138.       * @see eight.prototypedeepcopy2.Prototype#display()
139.       */
140.      public void display() {
141.          System.out.println("名字: " + name);
142.          System.out.println("性别: " + sex);
143.          System.out.println("年龄: " + age);
144.          System.out
145.              .println("工作经历: " + Work.getWorkdate() + Work.getWorkco
mpany());
146.      }
147.
148.      /*
149.       * (non-Javadoc)
150.       *
151.       * @see java.lang.Object#clone()
152.       */
153.      public Object clone() throws CloneNotSupportedException {
154.          ConcretePrototype cprototype=(ConcretePrototype)super.clone();
155.          Work=(WorkExperience) cprototype.getWork().clone();
156.          return cprototype;
157.      }
158.  }
159.  测试类
160.  /**
161.   *
162.   * @功能: 客户端测试类
163.   * @创建人 gao_jie
164.   * @创建日期 May 11, 2009
165.   * @版本 1.0
166.   *
167.   */
168.  public class Client {
169.

```

```
170.         /**
171.          * @param args
172.          */
173.         public static void main(String[] args) {
174.             Iprototype a = new ConcretePrototype("大鸟");
175.             a.setPersonInfo("男", "29");
176.             a.setWorkExperience("2009-12-19", "中创软件");
177.             Iprototype b = null;
178.             try {
179.                 b = (ConcretePrototype)a.clone();
180.             } catch (CloneNotSupportedException e) {
181.                 // TODO Auto-generated catch block
182.                 e.printStackTrace();
183.             }
184.             b.setPersonInfo("女", "25");
185.             b.setWorkExperience("2008-12-13", "中间件");
186.
187.             a.display();
188.             b.display();
189.         }
190.     }
191.     测试结果:
192.     名字: 大鸟
193.     性别: 男
194.     年龄: 29
195.     工作经历: 2009-12-19中创软件
196.     名字: 大鸟
197.     性别: 女
198.     年龄: 25
199.     工作经历: 2008-12-13中间件
```

两种方式实现设计模式中的原型模式。

[DJFocus新兴Web开发组件](#)

开发效率提高10倍，代码量减少80% 立即下载免费版！

www.djfocus.cn

Google 提供的广告

0

顶

0

踩

[问题汇总](#) | [java中的clone机制及序列化](#)

22:28 | [浏览 \(111\)](#) | [评论 \(0\)](#) | 分类: [设计模式](#) | [相关推荐](#)

评论

发表评论

文章已被作者锁定，不允许评论。

声明：JavaEye 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。

© 2003-2010 JavaEye.com. All rights reserved. 上海炯耐计算机软件有限公司 [沪ICP备05023328号]