链滴

Java, Java Bean 和 Spring Bean 的区别

作者：thas

原文链接：https://ld246.com/article/1591502312146

来源网站：链滴

<img src="https://ld246.com/images/img-loading.svg" alt="" data-src="https://b3logfile
com/bing/20181202.jpg?imageView2/1/w/960/h/540/interlace/1/q/100">
Java 是一名面向对象的编程语言, Java Object 是对象在 Java 中的表示形式, Java Class 则是 Jav
语言用来描述对象元信息的特殊对象 (Class 类是描述对象元信息的特殊类).
<h2 id="Bean">Bean</h2>
Bean 是一类特殊的对象, 比如具有属性, 事件等特定的特征, 它是脱离于编程语言的规范.
<h2 id="Java-Bean">Java Bean</h2>
C# 语言的类原生很强大, 本身就支持 Bean 规范.
<pre><code class="language-csharp highlight-chroma"><span
class="highlight-cl">public cl
ss Bean {

    <span class="h
ghlight-k">private String <span class="highlight
m">_value =</span
"";

    <span class="h
ghlight-k">public Bean<span class="highlight-p
>() {

    <span class="h
ghlight-p">}

    <span class="h
ghlight-k">public string <span class="highlight
n">Value {
        <span class=
highlight-k">get {
            <span clas
="highlight-k">return _<span class="highlight-
">value;
        <span class=
highlight-p">}
        <span class=
highlight-k">set {
            <span clas
="highlight-m">_value <span class="highlight-p
>= value;
        <span class=
highlight-p">}
    <span class="h
ghlight-p">}
<span class="high
ight-p">}
</code></pre>
Java 语言在这方面还有欠缺, 因为 Java Class 只有字段 (Field) 和方法 (Method, 构造器也算)
种成员, 为了表示 Bean 的 属性, Java 必须利用字段和方法两个成员组合实现 (俗称 "胶水代码"):
<pre><code class="language-java highlight-chroma"><span c
ass="highlight-cl">public c
ass Bean impleme
ts Serializable {</sp
n>

    private String value = "";

    public Bean() {

    }

    public String getValue() {
        return value;
    }

    public void setValue(String value) {
        this.value = value;
    }
}
</code></pre>
Bean 的规范并不是固定的, 比如早期 Bean 中的事件是为了 GUI 开发而制定的. 一般来说, Bean 有 POJO 和 EJB 两种规范. 由于 EJB (充血模型) 太重了, 所有就有了更加简洁的 POJO (贫血模型). 现在一般所讲的 Java Bean 都是 POJO, 所有回答 Java Bean 是什么, 回答 POJO 的定义就算对.
POJO 定义:

拥有无参构造
支持序列化
属性私有, 通过公开的 setter 或 getter 访问.

下面的一段话是个人理解, 非权威:
制定 Bean 规范的目的就是为了更好的描述对象 (抽象), 一个对象在程序运行时的表示不应当只是贫血模型, 它应当具有更多的特征或行为. Java Bean 可以用来实现可复用的组件, 但并不是说 Java Bean 就是可复用的组件.
<h2 id="Spring-Bean">Spring Bean</h2>
Spring Bean 对 Bean 做了一个对业务代码无侵入的实现方案, 引入了 BeanDefinition .开发者需要再强制依赖 EJB 的接口, 不需要再对无用接口进行空实现.
Spring Bean 并不强制 Bean 必须拥有哪些特征, 你可以通过 BeanDefinition 来定制 Bean 的特, Bean 可以是最简单的 POJO, 也可以具有完整的状态, 行为, 事件, 事务, 生命周期等.

Spring Bean 也并不强制 Spring Bean 就是 业务代码, 你可以通过字节码操作动态的生成一个类, 比如动态代理. Java 类和 Spring Bean 是隔离开的, 你写的 Java 类只是用来一种创建 Bean 的方式, 用这个类创建的对象则是 Spring Bean 在 JVM 中的表现形式. Spring Bean 是编程语言无关的.