

实验九.txt
实验九：Bean与容器的关系

```
BeanFactoryDemo
|
|---src
|   |
|   |---tarena
|       |
|       |-----spring-config.xml
```

加入Spring2的相关的JAR包

步骤一，创建FirstBean 类

```
public class FirstBean {

}
```

步骤二，创建SecondBean 类

```
public class SecondBean {

}
```

步骤三，创建BeanFactoryBean 类

```
public class BeanFactoryBean implements BeanFactoryAware {

    public BeanFactoryBean() {
        System.out.println("实例化");
    }

    public void setBeanFactory(
        BeanFactory beanFactory)
        throws BeansException {

        System.out.println("bean获得IoC容器的信息");
    }
}
```

实验九.txt

```
        boolean isHave =  
beanFactory.containsBean("FirstBean");  
  
        System.out.println(isHave?"有":"无");  
  
        boolean isSingleton =  
beanFactory.isSingleton("FirstBean");  
  
        System.out.println(isSingleton?"单例":"原型");  
  
    }  
  
}
```

步骤四，创建spring配置文件

```
<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  
xsi:schemaLocation="http://www.springframework.org/schema/beans  
http://www.springframework.org/schema/beans/spring-beans-2.0.x  
sd">
```

```
    <bean id="FirstBean" class="tarena.FirstBean" />
```

```
    <bean id="SecondBean" class="tarena.SecondBean" />
```

```
    <bean id="BeanFactoryBean"  
class="tarena.BeanFactoryBean"/>  
</beans>
```

步骤五，创建测试类

```
public class Test {  
  
    public static void main(String[] args) {
```

实验九.txt

```
Resource resource =  
    new  
ClassPathResource("tarena/spring-config.xml");  
  
BeanFactory beanFactory =  
    new XmlBeanFactory(resource);  
  
BeanFactoryBean bfb =  
(BeanFactoryBean)beanFactory.getBean("BeanFactoryBean");  
  
    }  
}
```

步骤六，结果

实例化

bean获得IoC容器的信息

有

单例