装配bean 基于 XML

实例方式：

1.默认构造

<bean name="自定义名" class="类路径"></bean>

当读取xml配置文件时会调用该类的默认构造，若不存在则抛出异常

自定义名：建议见名知意

2.静态工厂（开发中工厂由其他人提供，没人提供，所以我自己写工厂，工厂主要用于生产，这里我们生产对象）

1.创建需要的对象类

```
class UserServiceImpt implements Service{
    public void show(){
        System.out.println("你好！");
    }
}
```

2.静态工厂(静态工厂中全是静态方法)

==========静态工厂范围start==========

```
class staticFactory{
    public static Service creation(){
        return new UserServiceImpt();
    }
}
```

3.添加配置信息(xml)

```
<bean id="staticFactory" class="com.znsd.service.UserServiceImpt" factory-method="creation"></bean>
//factory-method:调用工厂指定方法
```

==========静态工厂范围end==========

4.测试类创建对象

```
class TestStaticFactory{
    public static void main(String[] args){
        String xmlUrl = "配置文件地址";
        ApplicationContext cation = new ClassPathXmlApplicationContext(xmlUrl);
        UserServiceImpt userBean = (UserServiceImpt)cation.getBean("staticFactory");
        userBean.show();
```

```
    }
}
```

3.实例工厂

==========实例工厂范围start==========

```
class Factory{
    public Service creation(){
        return new UserServiceImpt();
    }
}
```

添加配置信息(xml)

```
<bean id="Factory" class="com.znsd.service.Factory"></bean>
//读取这一行之后创建了Factory实例
<bean id="ServiceImpt" factory-Bean="Factory" factory-method="creation">
</bean>
//通过调用Factory实例中的creation方法获取ServiceImpt实例
```

==========实例工厂范围end==========

注意：

若 spring 不兼容 jdk 则有些标签是不能使用的

1.8 配 spring 4.0

若配spring 3.2 则有些标签不能使用

也可能会产生一些错误，如原本单例不再单例

Setter属性注入

引用 ref 值 value

举例

引用举例:

第一种方式:

```
<bean name="UserName" class="com.znsd.UserName">
    <poperty name="属性名" ref="需要注入的bean名称"></poperty>
</bean>
```

第二种方式：(若不为主标签都可以当作子标签)

```
<bean name="UserName" class="com.znsd.UserName">
    <poperty name="属性名">
        <ref="需要注入的bean名称"/>
    </poperty>
</bean>
```

值举例:

```
<bean name="UserName" class="com.znsd.UserName">
    <poperty name="属性名" value="需要注入的值"></poperty>
</bean>
```

第二种方式：(若不为主标签都可以当作子标签)

```
<bean name="UserName" class="com.znsd.UserName">
    <poperty name="属性名">
        <value>"需要注入的值"></value>
    </poperty>
</bean>
```

底层通过反射：//dao -> Dao ->setDao set演化为 set方法

P命名空间

使用p命名空间的前提，必须添加命名空间

添加步骤1：

```
<?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/spr
```

复制这段代码

粘贴到下方，在xmlns后加:p 将最后位置的beans改为p 这样p空间就添加好了

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:p="http://www.springframework.org/schema/p"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           http://www.springframework.org/schema/beans/spr
```

使用格式：

值：p:属性名="值";

引用：p:属性名-ref="bean名称";

一个简单的配置

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

<!-- bean definitions here -->
```

```
    <bean name="factorys" class="com.znsd.service.Factorys"
                    p:name="张三" p:age="12" p:testClass-ref="testP">
    </bean>
    <bean name="testP" class="com.znsd.service.TestClass"></bean>
</beans>
```

SpEL

也是与p空间差不多 用于 简化setter 属性注入的

上一个引用 需要加 p:属性名-ref="bean名称"; 值需要p:属性名="值";

俩种格式不是很同一，而SpEL同一了格式，不管是引用还是值统统为 #{}

常用格式：

字符串： #{'abc'}

数值: #{123}

引用:#{bean名称} 此外el 还支持调用引用的方法如：#{String.toString()} ,#{String.toString()?.toString}

?. 代表如果对象不为null则调用，否则直接返回//判断

```
<bean name="" class="">
  <poperty name="属性名" value="#{}"></poperty>
</bean>
```

集合注入

集合的注入都是给<property>添加子标签

数组：<array>

List：<list>

Set：<set>

Map：<map> ，

map存放k/v 键值对，

使用<entry>描述 Properties：

<props> <prop key=""></prop> 【】

普通数据：<value>

引用数据：<ref>

```
<bean id="collDataId" class="com.itheima.f_xml.e_coll.CollData" >
<property name="arrayData">
   <array>
      <value>DS</value>
      <value>DZD</value>
      <value>屌丝</value>
      <value>屌中屌</value>
   </array>
</property>
<property name="listData">
   <list>
      <value>于嵩楠</value>
      <value>曾卫</value>
      <value>杨煜</value>
      <value>曾小贤</value>
   </list>
</property>
<property name="setData">
   <set>
      <value>停封</value>
      <value>薄纸</value>
      <value>关系</value>
   </set>
 </property>
 <property name="mapData">
 <map>
    <entry key="jack" value="杰克">
    </entry>
    <entry>
       <key>
          <value>rose</value>
       </key>
       <value>肉丝</value>
    </entry>
  </map>
  </property>
  <property name="propsData">
     <props>
      <prop key="高富帅">嫩</prop>
       <prop key="白富美">嫩</prop>
        <prop key="男屌丝">拣</prop>
     </props>
  </property>
</bean>
```