

1、Spring中的DI给创建的对象属性赋值

DI：依赖注入，除了创建对象，还给对象的属性赋值，其中注入就是赋值的意思

#####

IOC控制反转在Spring中是用DI的方式实现，而DI这个方式实现有两种语法方式：

- 1)、使用Spring的配置文件，使用其中的标签和属性完成对象的创建，也可以用该方式给对象的属性赋值（注入），被称为基于XML的di实现
- 2)、使用Spring的注解类，完成对象创建，完成属性的赋值，被称为基于注解的di实现

#####

DI实现给对象属性的赋值又分为两种语法方式：

- 1)、set注入（设值注入）：Spring底层调用对象类的setter方法，来完成属性的赋值，80%使用
- 2)、构造方法注入：Spring底层调用对象类的有参数构造方法给属性赋值（很明显，这样需要对象类中有无参数和有参数的构造方法）

#

在idea-maven-spring工程中新建一个maven项目模块b-spring-di-fields-02，骨架与前一个项目模块一样

2、基于XML的di实现

2.1、基于XML的set注入----简单类型

1) 主程序创建一个a1setfields包，创建一个Student1类，其中的set方法输出一句话，测试是否调用了set方法

```
package com.studymyself.a1setfields;

public class Student1 {

    private String name;
    private int age;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        System.out.println("Spring调用setName: "+name);
    }
}
```

```

        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        System.out.println("Spring调用setAge: "+age);
        this.age = age;
    }

    @Override
    public String toString() {
        return "Student1{" +
            "name='" + name + '\'' +
            ", age=" + age +
            '}';
    }
}

```

2) 、在resources目录下创建a1setfields目录，新建Spring配置文件applicationContext.xml，要注意Spring获取该配置文件的路径变为 a1setfields\applicationContext.xml

```

<?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.xsd">

    <!--
        声明Student1对象
        di给属性赋值：
        set注入：
        1、简单类型的属性set注入：
        <bean id="xx" class="yyy">
            <property name="属性名字" value="该属性的值"/>
            <property .../>
        </bean>
        注意：
        1、一个property标签只能给一个属性注入，多个属性要用到多个标签
        2、标签中的name属性的值是对象类中set方法中参数的名字，不是属性的名字，但是
        开发中java类中的所有属性的set方法中的参数名都是跟其对应的属性名是一样的，
        所以name就填属性名
        3、一定要记住的是，Spring只是调用对象类中的set方法，无论是否有属性被赋值
        只要你提供了set方法中的参数名，以及给定的值，还有该对象中一定有对应的set方法
        Spring就不会报错
        4、记住xml中属性的值都是用双引号括起来的，即使参数值的类型是int类型
        也不例外，只不过到时注入的时候，判断是int类型，调用String类中的valueOf
        方法将数字字符串转换成int类型
    -->
    <bean id="student1" class="com.studymyself.a1setfields.Student1">
        <property name="name" value="赵四"/>
        <property name="age" value="40"/>
    </bean>

```

```
</beans>
```

3)、在test中新建一个MyTest01测试类

```
package com.studymyself;

import com.studymyself.a1setfields.Student01;
import org.junit.Test;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MyTest01 {

    /**
     * 测试di实现基于XML的set注入
     */
    @Test
    public void testStudent01(){

        //定义配置文件路径
        String config = "bao01/applicationContext.xml";

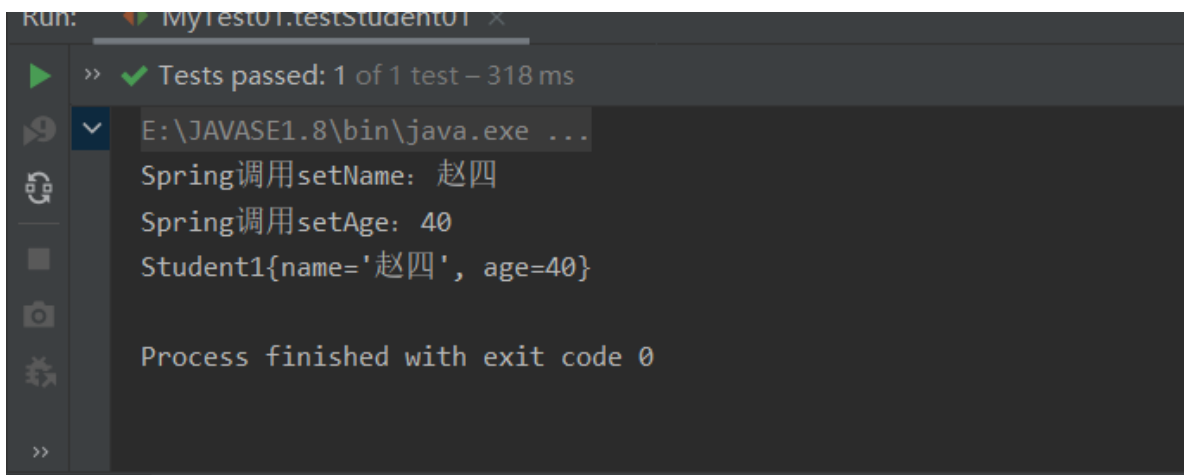
        //获取容器对象
        ApplicationContext ac = new ClassPathXmlApplicationContext(config);

        //获取容器中的对象，强制转换Student01类型
        Student01 student01 = (Student01) ac.getBean("student01");
        //输出对象
        System.out.println(student01);

    }

}
```

测试截图



问题：在测试程序的时候，配置文件的路径中每个目录的名字只允许最前面和最后面有0，不然找不到配置文件。aa0cc/xxx.xml 这是无法找到文件的。

2.2、基于XML的set注入-----引用类型

1)、新建一个a2setfields包，里面新建一个Student02和一个School类

```
package com.studymyself.a2setfields;

public class Student02 {

    private String name;
    private int age;

    //引用类型的字段
    private School school;

    public void setName(String name) {
        System.out.println("Spring调用setName: "+name);
        this.name = name;
    }

    public void setAge(int age) {
        System.out.println("Spring调用setAge: "+age);
        this.age = age;
    }

    public void setSchool(School school) {
        System.out.println("Spring调用setSchool: "+school);
        this.school = school;
    }

    @Override
    public String toString() {
        return "Student02{" +
            "name='" + name + '\'' +
            ", age=" + age +
            ", school=" + school +
            '}';
    }
}
```

```
package com.studymyself.a2setfields;

public class School {

    private String name;
    private String addr;

    public void setName(String name) {
        System.out.println("Spring调用setName: "+name);
        this.name = name;
    }

    public void setAddr(String addr) {
        System.out.println("Spring调用setAddr: "+addr);
    }
}
```

```

        this.addr = addr;
    }

    @Override
    public String toString() {
        return "School{" +
            "name='" + name + '\'' +
            ", addr='" + addr + '\'' +
            '}';
    }
}

```

2) 、在resources目录下创建一个bao02目录，新建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.xsd">

    <!--
        引用类型的set注入：
        引用类型与简单类型的区别就是value位置是ref属性，值是一个对象或者说bean标签的id
        这个引用对象也被在另一个bean标签中声明，引用对象的属性也在其中被set注入
        层层深入
    -->
    <bean id="student02" class="com.studymyself.a2setfields.Student02">
        <property name="name" value="智控"/>
        <property name="age" value="25"/>
        <property name="school" ref="school"/>
    </bean>

    <!--声明School类的对象-->
    <bean id="school" class="com.studymyself.a2setfields.School">
        <property name="name" value="北京大学"/>
        <property name="addr" value="北京"/>
    </bean>
</beans>

```

然后进行测试就行了

2.3、基于XML的构造方法注入

该方式了解就行，就不写实现例子了

很明显该方式得需要对象类中有无参数和有参数的构造方法，使用的是标签，类中构造方法的参数有多少个，就使用多少个该标签，其中的相关属性意义如下：

name：表示构造方法中传入参数的形参名

index: 表示构造方法中该形参的位置下标，从左往右下标依次是0,1,2,3...

value:构造方法的形参类型是简单类型，使用value属性保存传入的值

ref: 形参是引用类型，用ref属性

```
<bean id="student03" class="com.studymyself.a2setfields.Student02">
    <constructor-arg name="name" value="zz"/>
    <constructor-arg name="age" value="25"/>
    <constructor-arg name="school" ref="school"/>
</bean>
```

或者通过索引的方式

```
<bean id="student03" class="com.studymyself.a2setfields.Student02">
    <constructor-arg index="0" value="zz"/>
    <constructor-arg index="1" value="25"/>
    <constructor-arg index="3" ref="school"/>
</bean>
```

<!--声明School类的对象-->

```
<bean id="school" class="com.studymyself.a2setfields.School">
    <property name="name" value="北京大学"/>
    <property name="addr" value="北京"/>
</bean>
```