



今克

Java EE框架 ---Spring

Java EE framework --Spring

王磊

部门：研发部

CONTENTS



引入Spring 本地约束



创建JavaBean



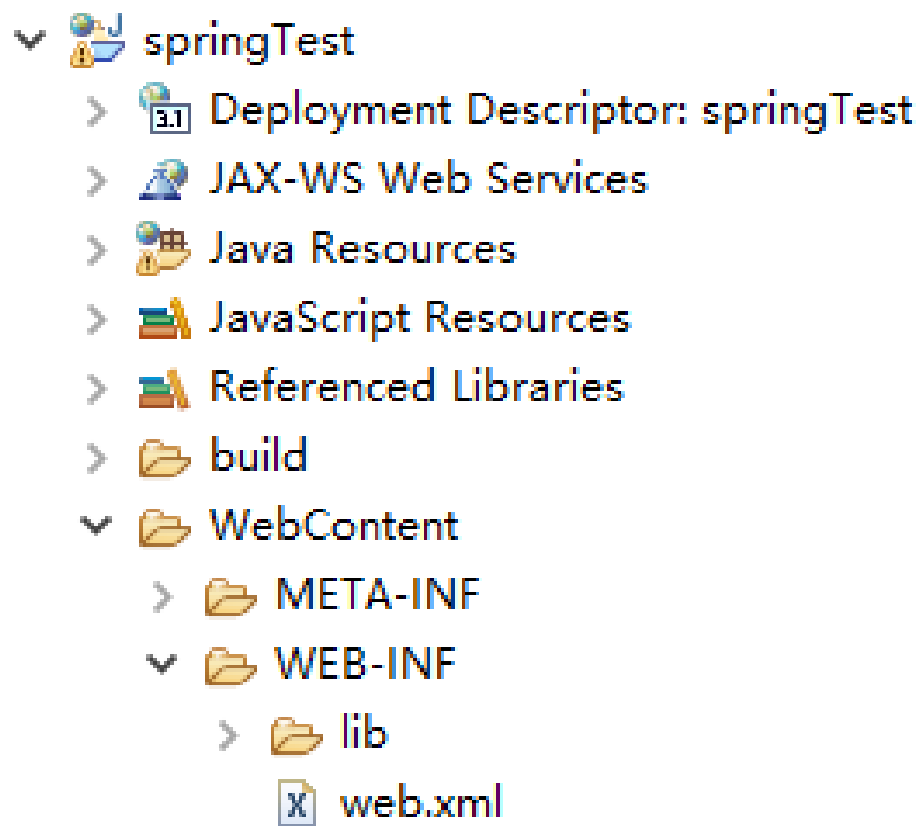
创建配置文件



单元测试

• 一、创建JavaWeb项目












- 1、新建Dynamic Web Project
- 2、引入 最小Spring jar包
- 3、构建路径



二、搭建Spring开发环境

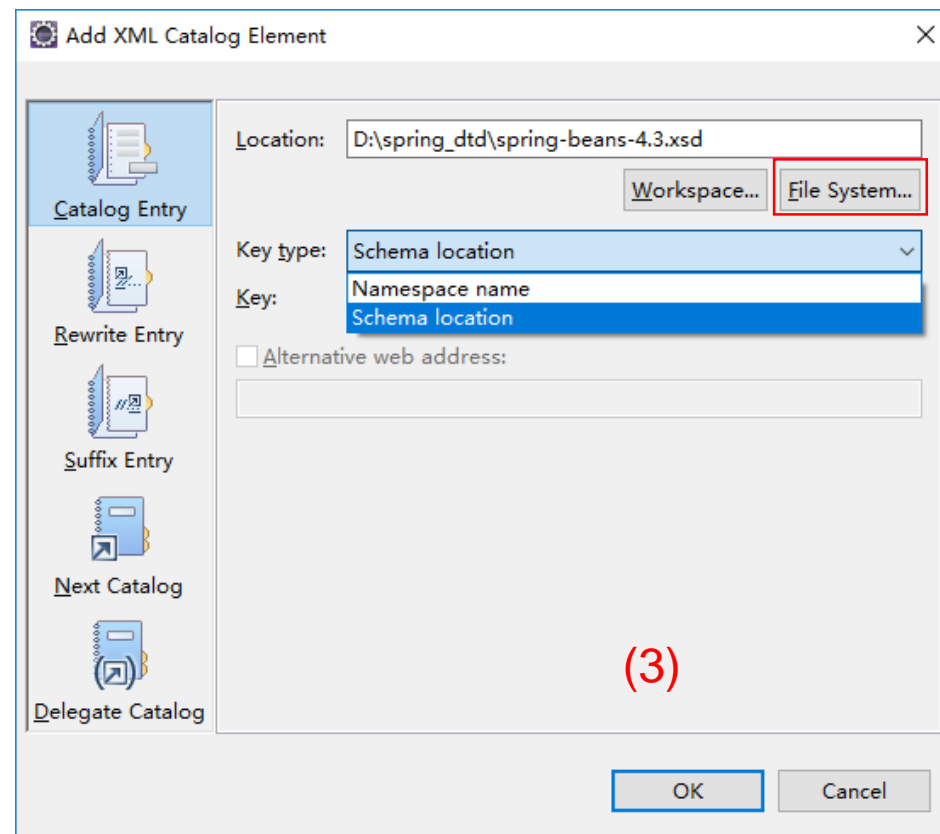
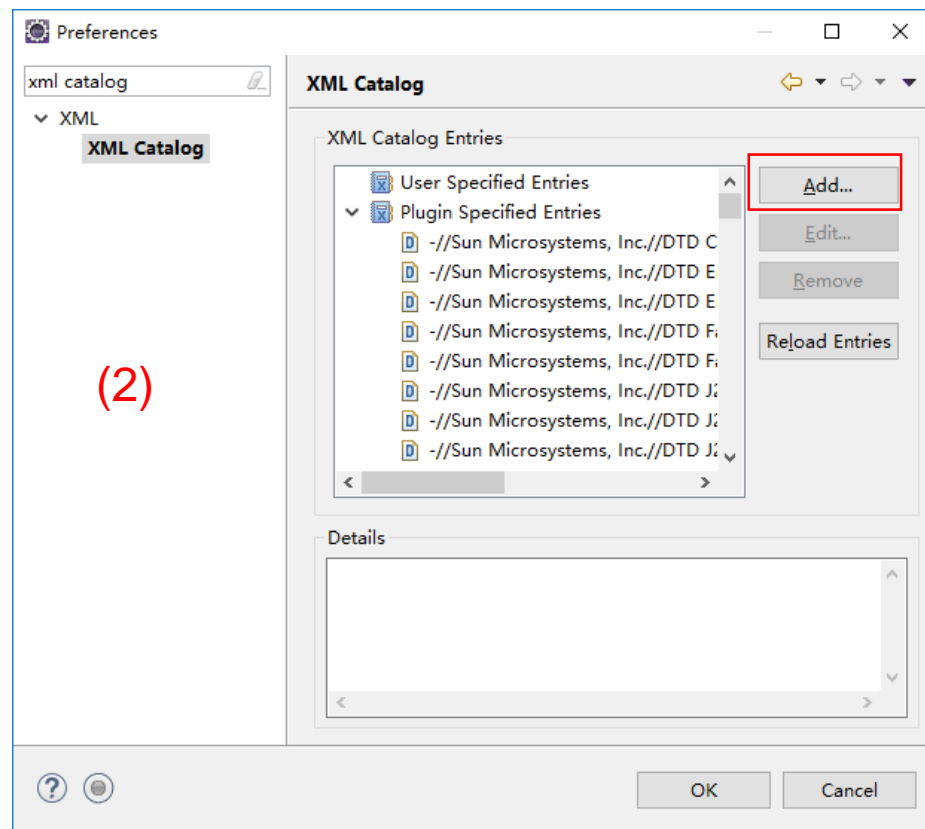
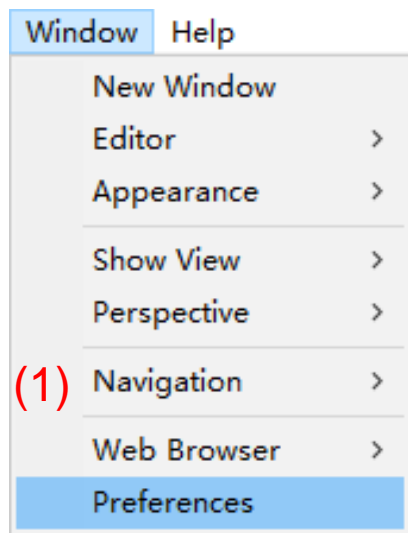
加入搭建Spring 开发环境的必需Jar包

▼  lib

-  commons-logging-1.2.jar
-  spring-aop-4.3.8.RELEASE.jar
-  spring-aspects-4.3.8.RELEASE.jar
-  spring-beans-4.3.8.RELEASE.jar
-  spring-context-4.3.8.RELEASE.jar
-  spring-core-4.3.8.RELEASE.jar
-  spring-expression-4.3.8.RELEASE.jar
-  spring-jdbc-4.3.8.RELEASE.jar
-  spring-tx-4.3.8.RELEASE.jar
-  spring-web-4.3.8.RELEASE.jar
-  web.xml

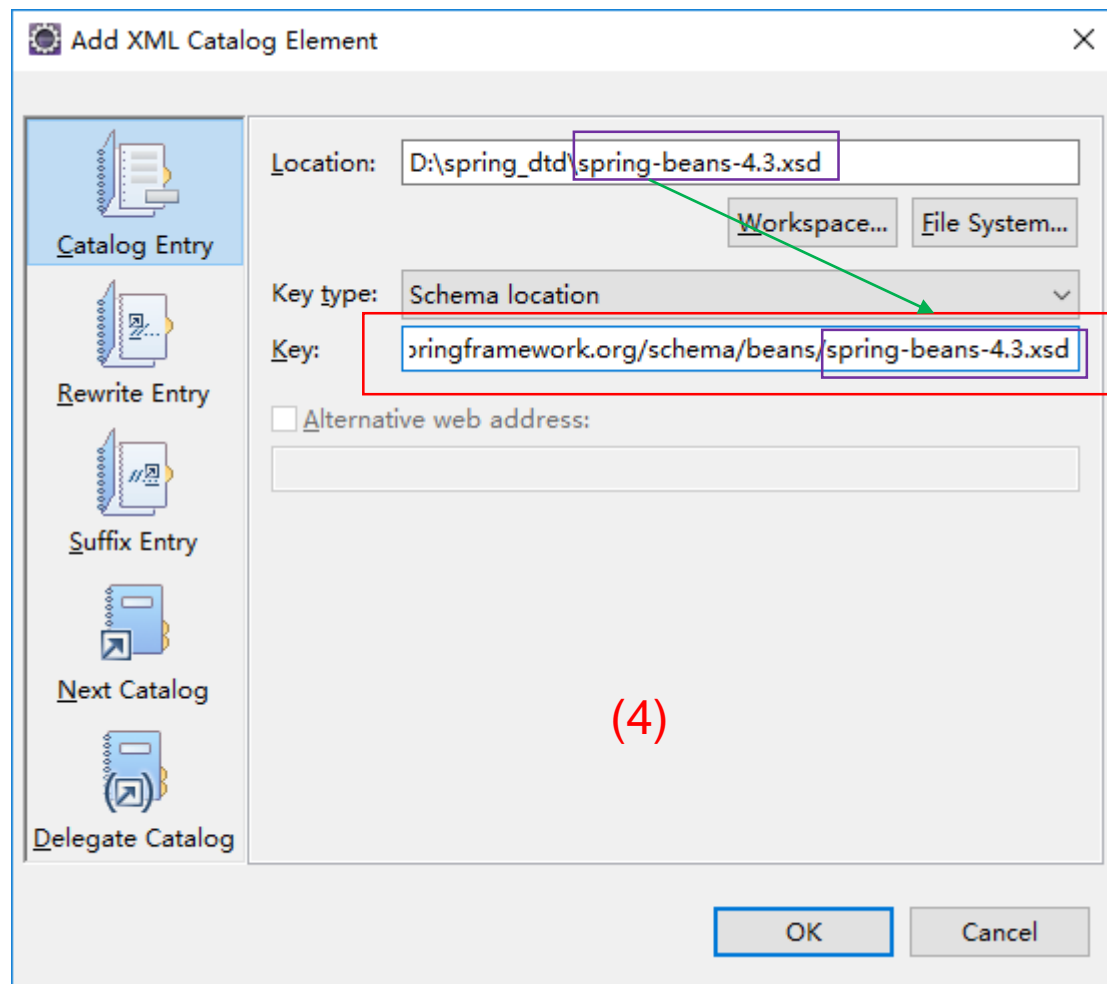
三、引入Spring 本地约束

1、加入搭建Spring 开发环境的必需Jar包



三、引入Spring 本地约束

1、加入搭建Spring 开发环境的必需Jar包



```
applicationContext.xml ⌵
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans>
3
4 </beans>
```

(5)

applicationContext.xml ⌵	
Node	Content
?? xml	version="1.0" encoding="UTF-8"
e beans	

(6)

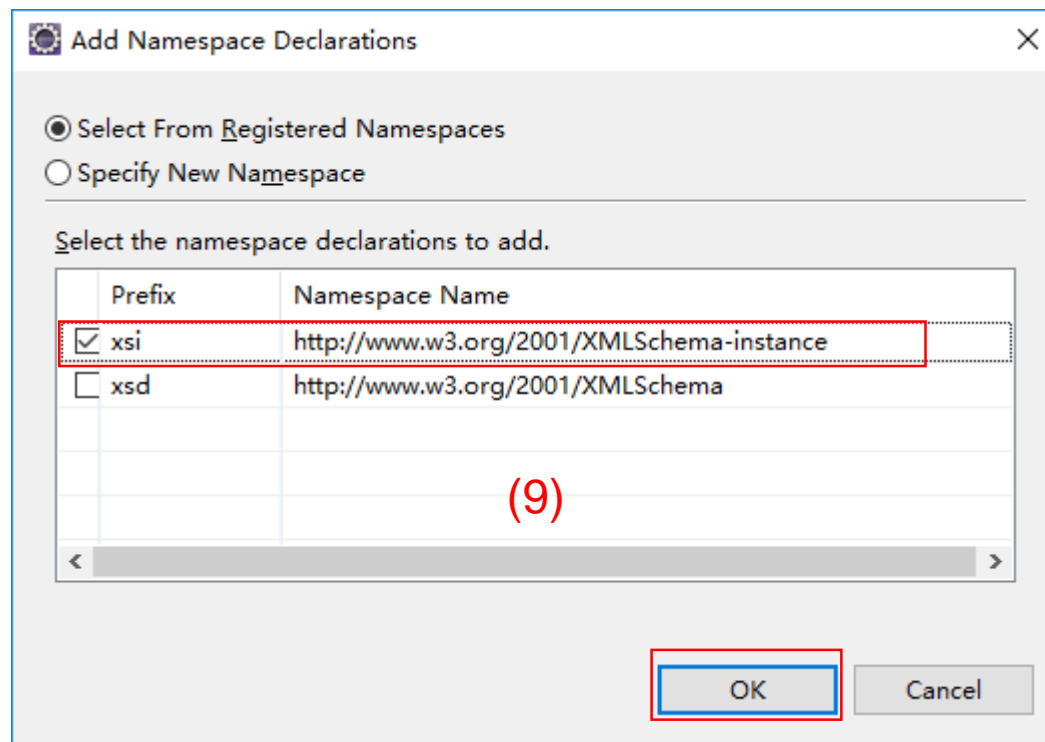
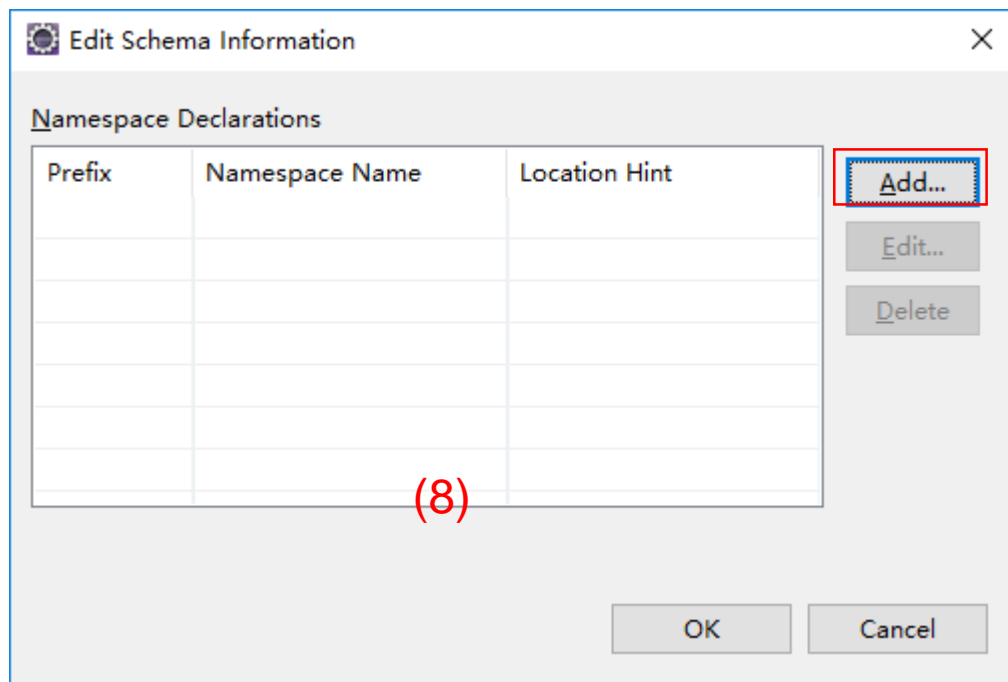
Node	Content
?? xml	version="1.0" encoding="UTF-8"
e beans	

- Remove
- Add DTD Information...
- Edit Namespaces...
- Add Attribute
- Add Child
- Add Before
- Add After

(7)

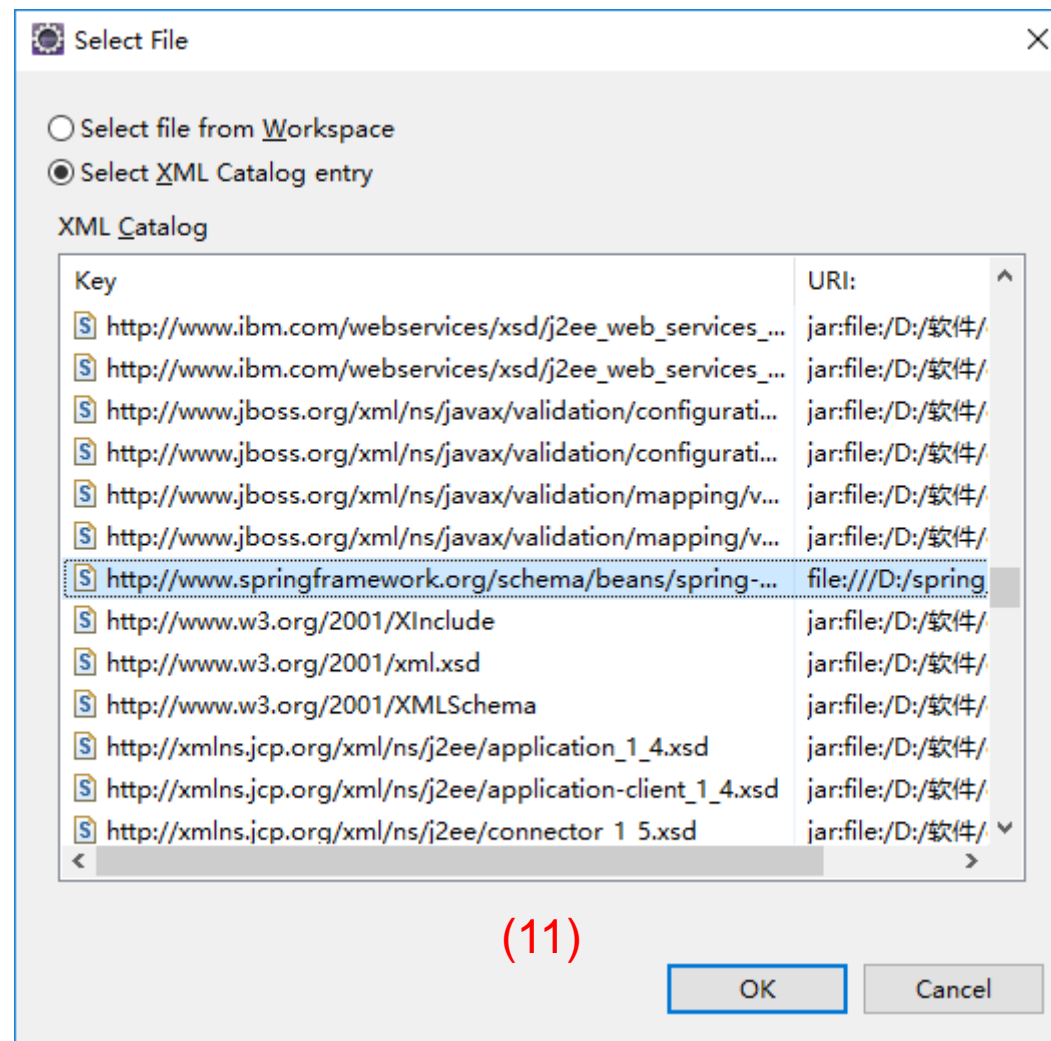
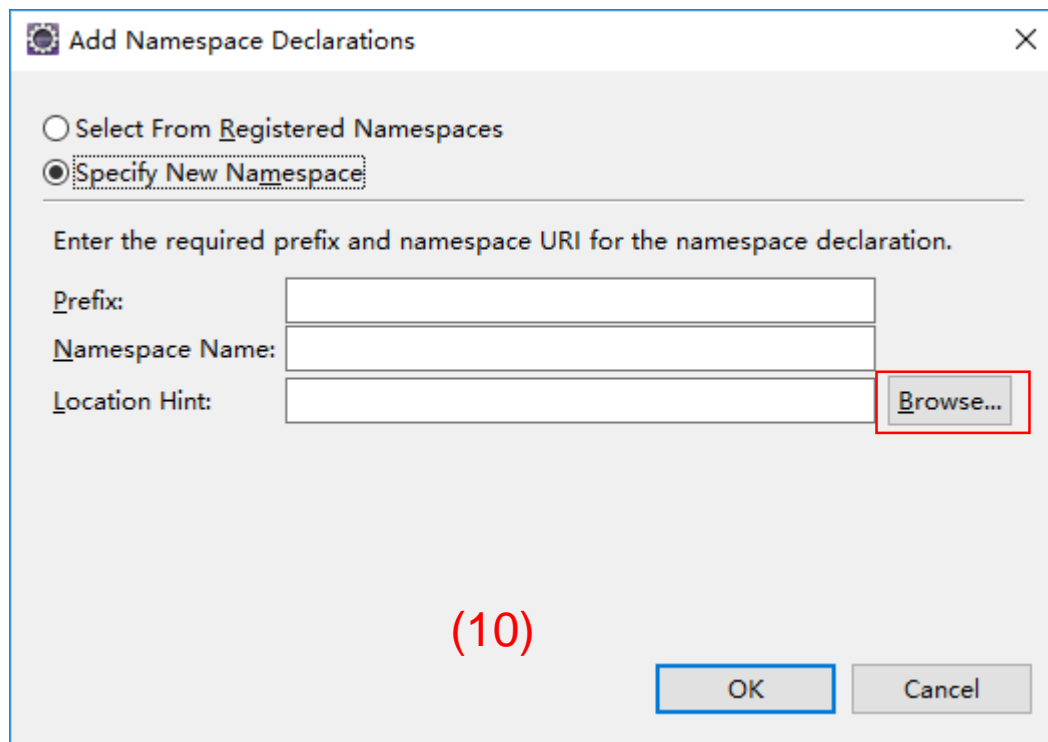
三、引入Spring 本地约束

1、加入搭建Spring 开发环境的必需Jar包



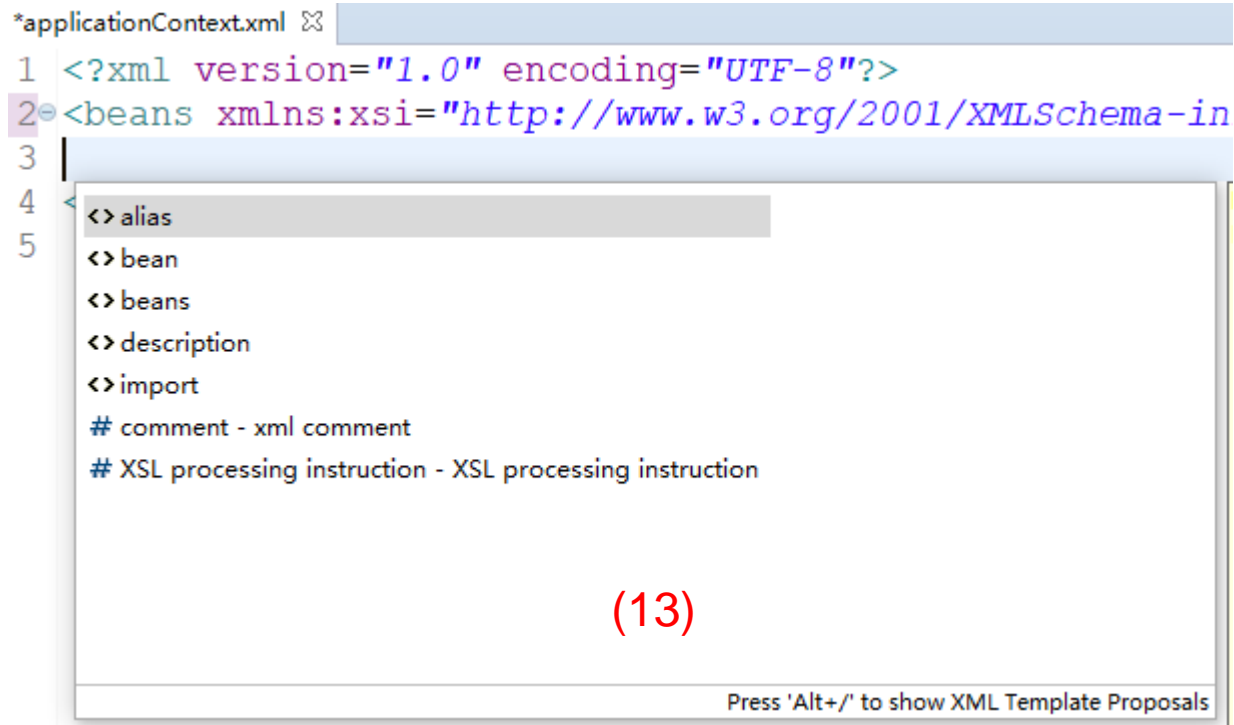
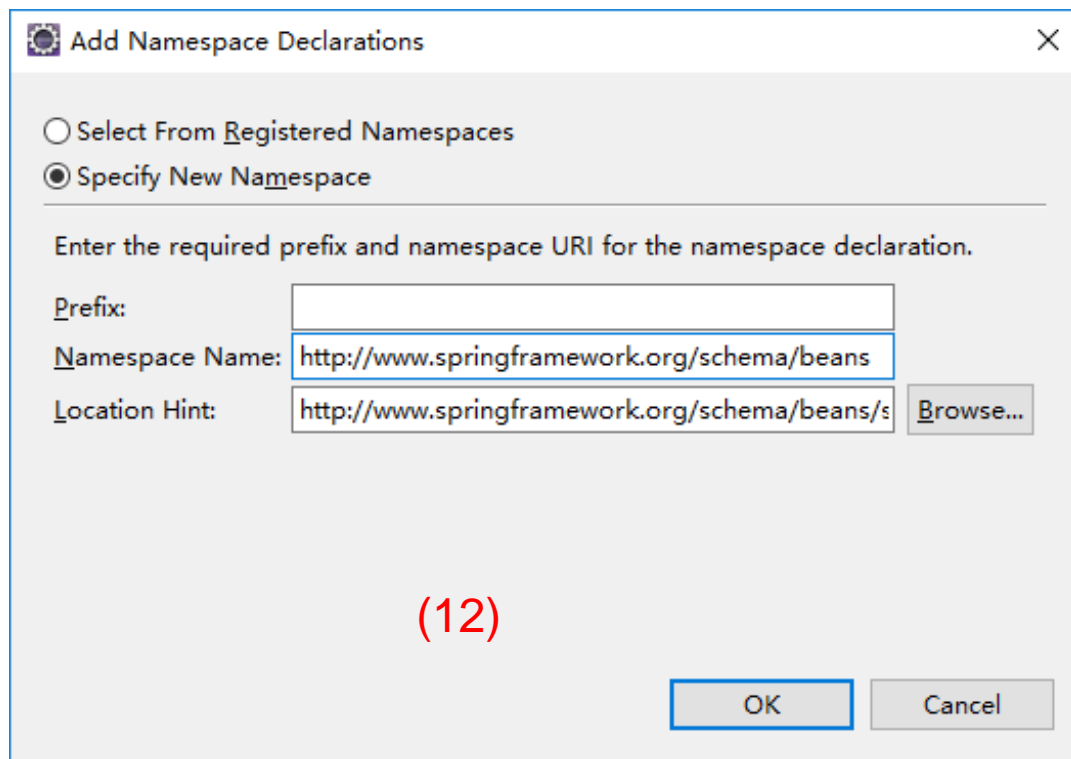
三、引入Spring 本地约束

1、加入搭建Spring 开发环境的必需Jar包



三、引入Spring 本地约束

1、加入搭建Spring 开发环境的必需Jar包



四、创建JavaBean

1、创建computer对象

```
package com.xk.model;
public class Computer {
    private String logo;
    private Float price;
    public String getLogo() {
        return logo;
    }
    public void setLogo(String logo) {
        this.logo = logo;
    }
    public Float getPrice() {
        return price;
    }
    public void setPrice(Float price) {
        this.price = price;
    }
}
```

四、创建JavaBean

2、创建user对象

```
package com.xk.model;
public class User {
    private String name;
    private Integer age;
    private Computer computer;
    public Computer getComputer() {
        return computer;
    }
    public void setComputer(Computer computer) {
        this.computer = computer;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public Integer getAge() {
        return age;
    }
    public void setAge(Integer age) {
        this.age = age;
    }
}
```

五、创建配置文件

1、创建配置文件applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.springframework.org/schema/beans"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.3.xsd ">

<!-- 把user对象注册到Spring容器中 -->
    <bean name="user" class="com.xk.model.User">
        <property name="name" value="mary"></property>
        <property name="age" value="22"></property>
        <property name="computer" ref="computer"></property>
    </bean>
    <bean name="computer" class="com.xk.model.Computer">
        <property name="logo" value="DELL"></property>
        <property name="price" value="5000"></property>
    </bean>
</beans>
```

六、创建junit测试类

1、创建junit测试类

```
package com.xk.junit;
import org.junit.Test;

public class Demo {
    @Test
    public void testDemo() {
        //1创建容器
        ApplicationContext ac = new ClassPathXmlApplicationContext("applicationContext.xml");
        //2从容器中取出user对象
        User user = (User) ac.getBean("user");
        //3打印user对象
        System.out.println(user.getName()+"今年"+user.getAge()+"岁, "
            +"有一台"+user.getComputer().getLogo()+"笔记本");
    }
}
```

七、运行并观察结果

1、运行并观察结果

```
十一月17, 2018 9:13:55 上午 org.springframework.context.support
信息: Refreshing org.springframework.context.support.ClassPat
十一月17, 2018 9:13:55 上午 org.springframework.beans.factory.xml
信息: Loading XML bean definitions from class path resource [
mary今年22岁, 有一台DELL笔记本
```

• 总结 •

- 1、引入Spring 本地约束
- 2、创建JavaBean
- 3、创建配置文件applicationContext.xml
- 4、运行并观察结果

