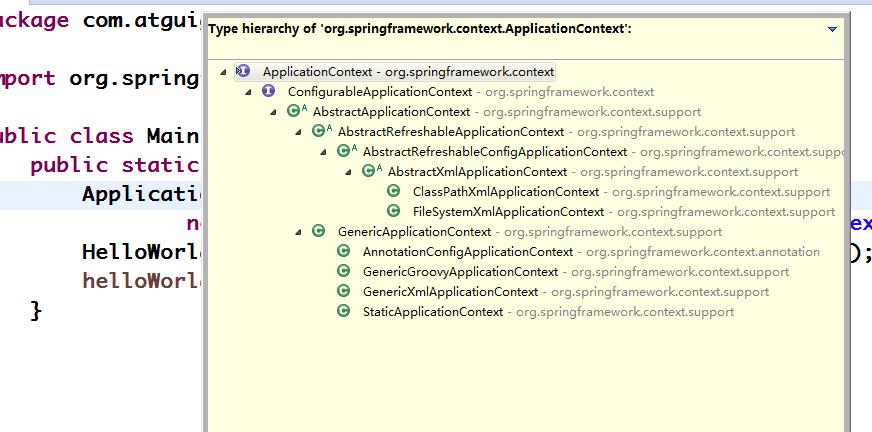
Video1-3：

1. spring的helloworld
2. 装eclipse插件
3. 导Jar包 （logging+4必须） -->build path
4. applicationContext.xml配置bean
5. ApplicationContext接口(它代表IOC容器)继承最底层IOC容器BeanFactory接口

继承图：ctrl+t





1. 属性注入 用set方法/constructor-arg注入

a). <bean id=*"person2"* class=*"com.atguigu.spring.helloworld.Person"*>

<constructor-arg value=*"Jerry"*></constructor-arg>

<constructor-arg value=*"25"*></constructor-arg>

<constructor-arg ref=*"car"*></constructor-arg>

<!-- 为级联属性car.maxSpeed赋值 -->

<property name=*"car.maxSpeed"* value=*"300000"*> </property>

</bean>

必须有ref=”car”不能直接给级联属性赋值，struts可以…(这点很少用)

b).

使用<list> <set>配置集合属性，如一人有多辆车。。

集合标签还可以定义内部bean

一人多车实例

<bean id=*"person3"* class=*"com.atguigu.spring.beans.collection.Person"*>

<property name=*"name"* value=*"Lilly"*></property>

<property name=*"age"* value=*"28"*></property>

<property name=*"cars"*>

<list>

<ref bean=*"car"*/>

<ref bean=*"car2"*/>

<bean class=*"com.atguigu.spring.helloworld.Car"*>

<constructor-arg value=*"Ford"*></constructor-arg>

<constructor-arg value=*"changan"*></constructor-arg>

<constructor-arg value=*"200000.0"* type=*"double"*></constructor-arg>

</bean>

</list>

</property>

</bean>

配置map：用<map>+entry子节点就好了。。。简单

<bean id=*"newperson"* class=*"com.atguigu.spring.beans.collection.NewPerson"*>

<property name=*"name"* value=*"Rose"*></property>

<property name=*"age"* value=*"29"*></property>

<property name=*"cars"*>

<map>

<entry key=*"AAAA"* value-ref=*"car"*></entry>

<entry key=*"BBBB"* value-ref=*"car2"*></entry>

</map>

</property>

</bean>

另外：使用<props>配置java.util.Properties（这东西是map的一个子类）

<bean id=*"dataSource"* class=*"com.atguigu.spring.beans.collection.DataSource"*>

<property name=*"properties"*>

<props>

<prop key=*"user"*>root</prop>

<prop key=*"password"*></prop>

<prop key=*"jdbcUrl"*>jdbc:mysql:///hibernate</prop>

<prop key=*"driverClass"*>com.jdbc.mysql.Driver</prop>

</props>

</property>

</bean>

c）:配置单利的集合bean，以供大家公用。。。。

<!-- 配置单利的集合bean，以供大家使用。。。。需要导入util namespace -->

<util:list id=*"cars"*>

<ref bean=*"car"*/>

<ref bean=*"car2"*/>

</util:list>

<bean id=*"person4"* class=*"com.atguigu.spring.beans.collection.Person"*>

<property name=*"name"* value=*"Jack"*></property>

<property name=*"age"* value=*"88"*></property>

<property name=*"cars"* ref=*"cars"*></property>

</bean>

d). <!-- 使用p:方式给属性赋值， 需要导入p namespace,这样更简单了-->

<bean id=*"person5"* class=*"com.atguigu.spring.beans.collection.Person"* p:age=*"20"* p:name=*"Queen"* p:cars-ref=*"cars"*></bean>

Video5..自动装配

Spring根据byName或者byType自动装配，bean里的

要求：

byName：若有setAddress方法，则bean id必须要是address才能装配上

byType：根据bean的类型和当前bean的属性的类型自动匹配，如果有两个address、address2，则不能装配。。。

实际开发中，很少自动装配。。。整合第三方框架时用的多些。。

Video6.

1.继承 beans-relation.xml

就是parent、abstract属性的配置

若某个bean没有class属性，它直接就是个模板bean

<?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 id=*"address"* p:city=*"Beijing~~~"* p:street=*"Huolushan"* abstract=*"true"*></bean>

<bean id=*"address2"* class=*"com.atguigu.spring.beans.aotuwire.Address"*

parent=*"address"* p:city=*"wuhan"* p:street=*"Wudaok"*></bean>

</beans>

1. depends-on属性：依赖。。。不会赋值。。。。

<bean id=*"address"*

p:city=*"Beijing~~~"* p:street=*"Huolushan"* abstract=*"true"*></bean>

<bean id=*"address2"* class=*"com.atguigu.spring.beans.aotuwire.Address"*

parent=*"address"* p:city=*"wuhan"* p:street=*"Wudaokou"* ></bean>

<bean id=*"car"* class=*"com.atguigu.spring.beans.aotuwire.Car"*

p:brand=*"audi"* p:price=*"120000"*></bean>

<bean id=*"person"* class=*"com.atguigu.spring.beans.aotuwire.Person"*

p:name=*"Tom"* p:address-ref=*"address2"* depends-on=*"car"*/>

Video7。。bean的作用域

<bean id=*"car"*

class=*"com.atguigu.spring.beans.aotuwire.Car"*

scope=*"prototype"*

p:brand=*"audi"* p:price=*"10000"*>

</bean>

Video8。。加载外部属性文件（如mysql数据库连接配置信息）

PPT有。。。简单。

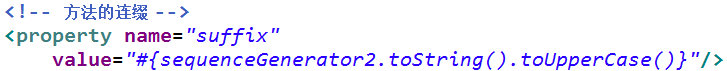
Video9。。SpEL

几个特别的：

引用其他对象的属性：



调用其他方法，还可以链式操作。。



等等 正则。。。，。。。。见PPT

Spel例子：

<bean id=*"address"* class=*"com.atguigu.spring.beans.spel.Address"*>

<property name=*"city"* value=*"#{'BeiJing'}"*></property>

<property name=*"street"* value=*"五道口"*></property>

</bean>

<bean id=*"car"* class=*"com.atguigu.spring.beans.spel.Car"*>

<property name=*"brand"* value=*"Audi"*></property>

<property name=*"price"* value=*"500000"*></property>

<!-- 用spel引用静态变量 -->

<property name=*"tyrePerimeter"* value=*"#{T(java.lang.Math).PI \* 80}"*></property>

</bean>

<bean id=*"person"* class=*"com.atguigu.spring.beans.spel.Person"*>

<property name=*"name"* value=*"Tom"*></property>

<property name=*"car"* value=*"#{car}"*></property>

<property name=*"city"* value=*"#{address.city}"*></property>

<property name=*"info"* value=*"#{car.price > 30 ?'金领':'金领'}"*></property>

</bean>

Video10：配置bean的生命周期：

<bean id=*"car"* class=*"com.atguigu.spring.beans.cycle.Car"*

init-method=*"init"* destroy-method=*"destroy"*>

<property name=*"brand"* value=*"Audi.."*></property>

</bean>

<bean class=*"com.atguigu.spring.beans.cycle.MyBeanPostProcessor"*></bean>

**public** **class** Main {

**public** **static** **void** main(String[] args) {

ClassPathXmlApplicationContext ctx = **new** ClassPathXmlApplicationContext("beans-cycle.xml");

Car car = (Car) ctx.getBean("car");

System.***out***.println(car);

ctx.close();

}

}

Car`s constructor..

set Brand......

postProcessBeforeInitialization

Car`s constructor..

set Brand......

init...

postProcessAfterInitialization.....

Car [brand=QQ..]

三月 05, 2015 6:56:09 下午 org.springframework.context.support.AbstractApplicationContext doClose

信息: Closing org.springframework.context.support.ClassPathXmlApplicationContext@4ebaecb8: startup date [Thu Mar 05 18:56:08 CST 2015]; root of context hierarchy

destroy...