工厂方法

**静态工厂**

1. **写一个工厂类**

**public** **class** StaticCarFactory {

**private** **static** HashMap<String, Car> *cars* = **new** HashMap<String, Car>();

**static** {

*cars*.put("aodi", **new** Car("aodi", 300000d));

*cars*.put("baoma", **new** Car("baoma", 200000d));

}

// 静态工厂方法

**public** **static** Car getCar(String name) {

**return** *cars*.get(name);

}

}

1. **配置：**

<!-- 配置静态工厂，获取bean的时候直接获取的是配置好的bean，factory-method指定静态工厂方法名，class指定工厂的全类名 -->

<bean id=*"car1"* class=*"factory.StaticCarFactory"*

factory-method=*"getCar"*>

<!-- 传如工厂方法所需要的参数参数 -->

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

</bean>

**实例工厂方法**

1. **写一个工厂类**

/\*\*

\* 实例工厂方法,必须先创建工厂本身，在调用方法获取bean实例

\*/

**public** **class** InstanceCarFactry {

**private** Map<String, Car> cars = **null**;

**public** InstanceCarFactry() {

cars=**new** HashMap<>();

cars.put("aodi", **new** Car("aodi", 300000d));

cars.put("baoma", **new** Car("baoma", 200000d));

}

**public** Car getCar(String name) {

**return** cars.get(name);

}

}

1. **配置**

<!-- 配置工厂实例 -->

<bean id=*"instanceCarFactory"* class=*"factory.InstanceCarFactry"*></bean>

<!-- 配置bean实例 factory-bean指向实例工厂方法的bean,factory-method为工厂的获取bean方法名 -->

<bean id=*"car2"* factory-bean=*"instanceCarFactory"* factory-method=*"getCar"*>

<!-- 传参 -->

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

</bean>