**Bean Instantion**

• **Spring** creates the instances of the declared beans using reflection.

• Depending on the injection of the dependencies, default, non-argument constructor or a suitable argument-constructor must be provided.

• **Spring** either invokes the constructor of the bean or it calls a static factory method on the same class if specified to create the instance.



Spring reflection kullanarak bean’lerin instancelarını olusturur.

Spring ya default constructor çağırır sonra setlerle inject eder ya da akıllı constructoru çağırıp onunla inject eder.

• Spring can call a static factory method to create the instance.

• For the purpose the name of the static factory method must be provided for the **factory-method attribute** of **<bean/>** element.

• Spring still injects all declared dependencies to the bean.

Eğer istersek o bean’in üzerinde constructor çağrısı yapmasını önleyip static **factory-methodlar** sağlanıp çağırabiliriz.

\***Buradaki factory-method design patternd bloch’un çözümü olan static factory-method çözümüyle yapmak istersek.**

Bunu bean’in üzerindeki attiribute olarak söyleyebiliriz.

• **Spring** can call an **instance** **factory method** on another bean to create the instance.

• For the purpose the name of the class that has **the non-static** **factory** method along with the bean name must be provided for **the factory-bean and factory-method attributes of <bean/> element.**

• class attribute can be omitted.

• **Spring** still injects all **declared** **dependencies** to the object.

**Eğer abstract factory-method patterni kullanacaksak, kendimiz apayrı abstract factory olusturup bunu springe configure ederiz. Bu sekilde de bean olusturulmasını sağlarız.**