Lazy & Eager Loading :

For Singleton scope : only one object for bean will be created

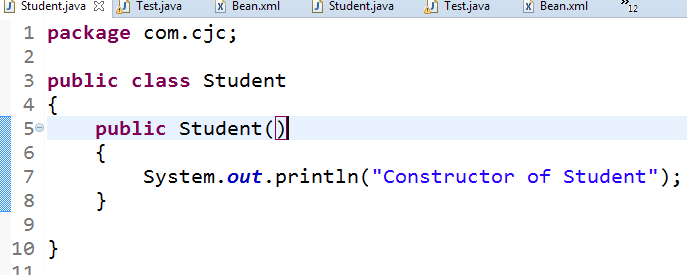
* BeanFactory : lazy-loading (i.e. object is created on request)
* ApplicationContext : eager- loading (i.e. object is created when xml file loads)

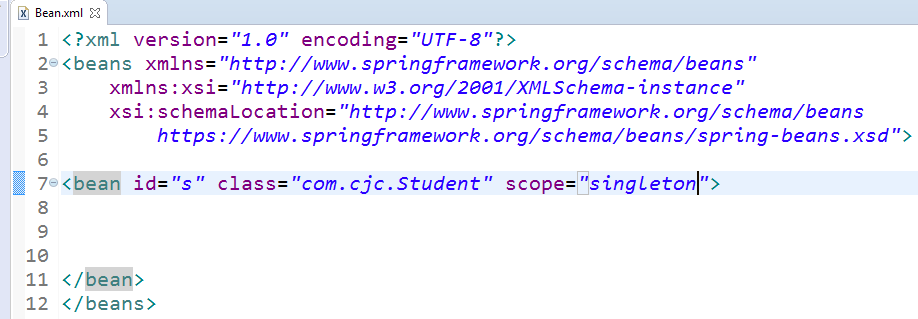
Let us see above concept practically...

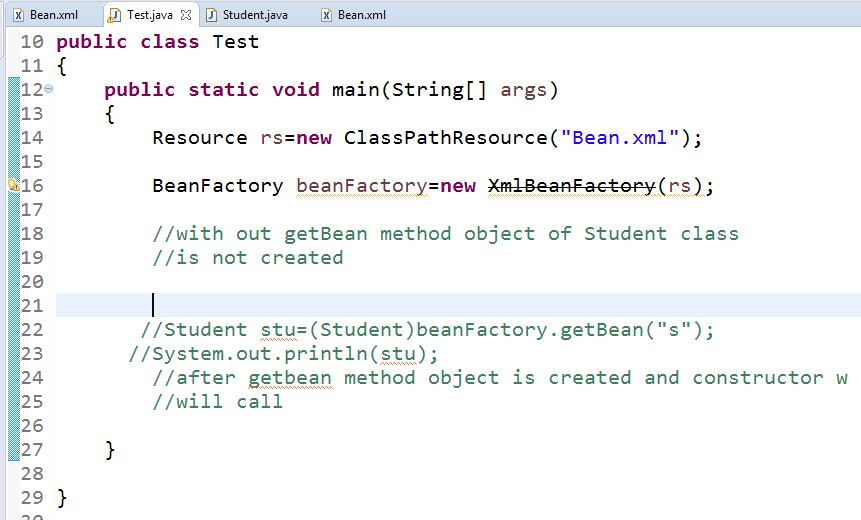
For prototype Scope:

* BeanFactory : lazy-loading (i.e. object is created on request)
* ApplicationContext : Lazy- loading (i.e. object is created on request)

Let us see above concept practically...







If BeanFactory scope is prototype then also lazy loading will Occure

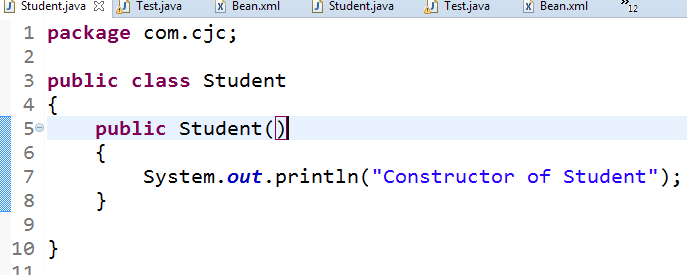
All program is same as previous program

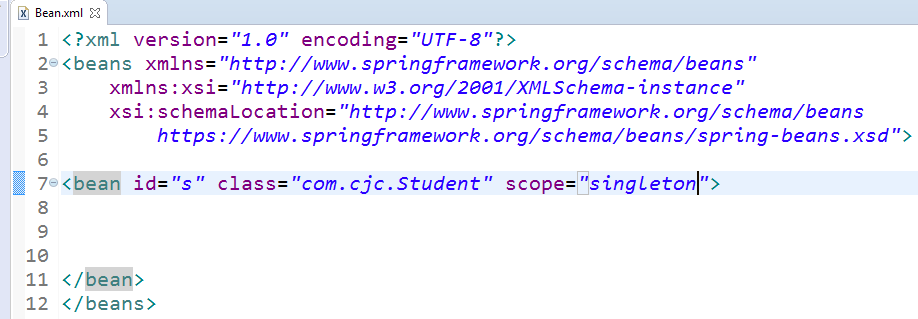
Only change scope=prototype

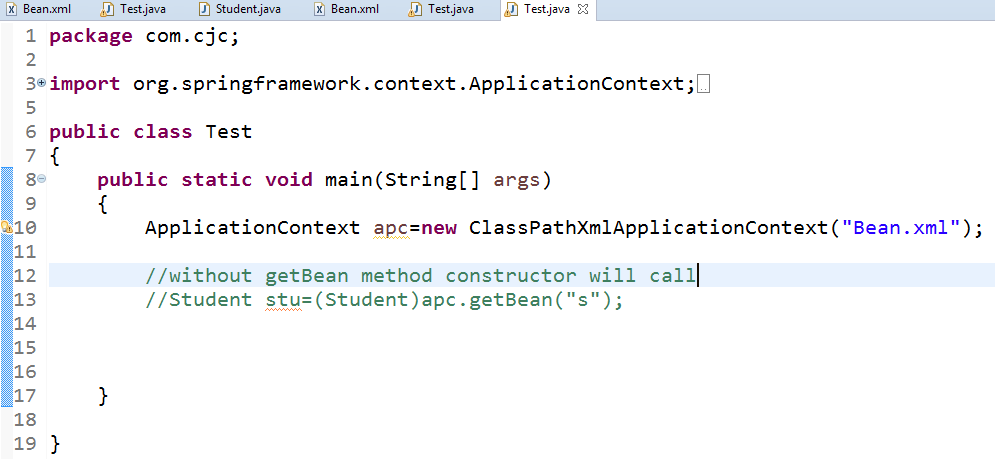
* ApplicationContext : eager- loading (i.e. object is created when xml file loads)

Let us see above concept practically...

For this scenario eager loading will occure







LazyLoading And EagerLoading both container:

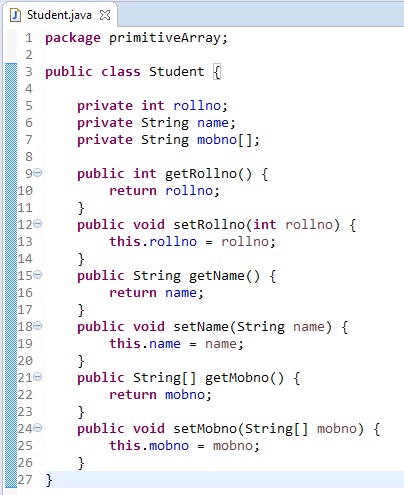
|  |  |  |
| --- | --- | --- |
| Containers | Singleton | Prototype |
| BeanFactory | LazyLoading | LazyLoading |
| ApplicationContext | EagerLoading | LazyLoading |

Setter Injection –primitiveArray:

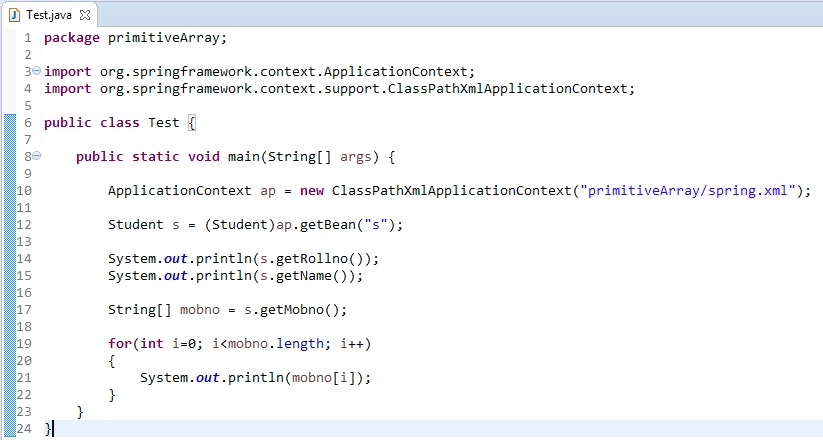
Files required : 1)Student.java

2) Test.java

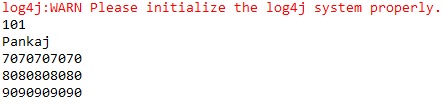
3)spring.xml +SpringJars







Output :



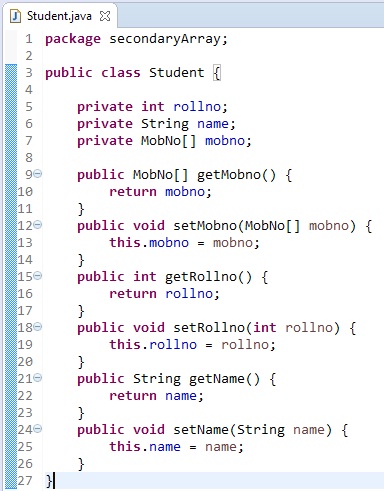
Setter Injection –secondaryArray:

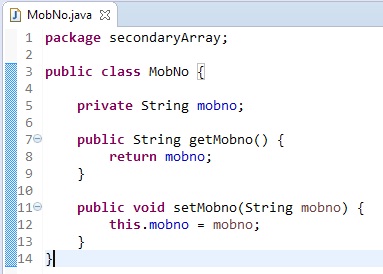
Files required : 1)Student.java

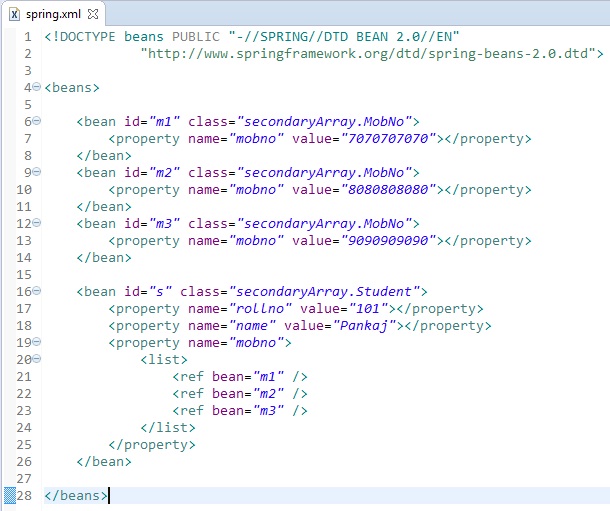
2)MobNo.java

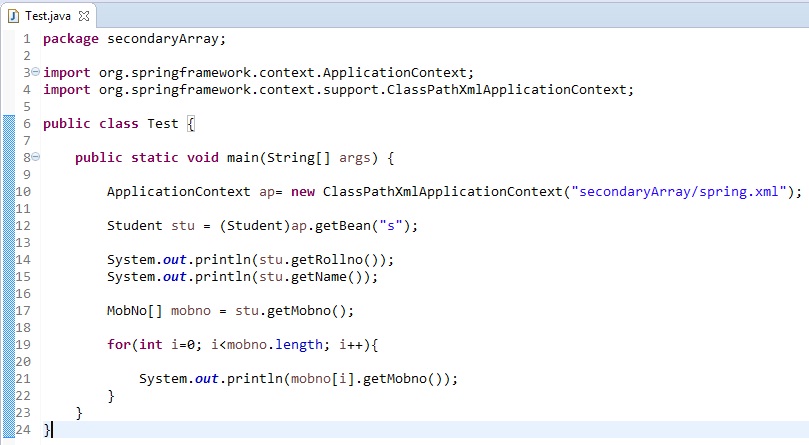
3) Test.java

4)spring.xml +SpringJars

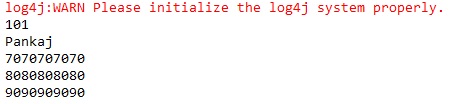








Output :



Setter Injection –byInstance:

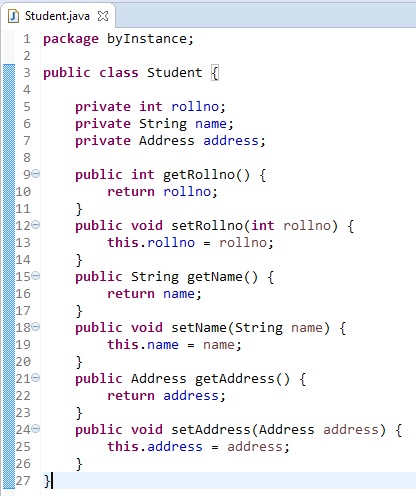
ByInstance – creating bean inside bean

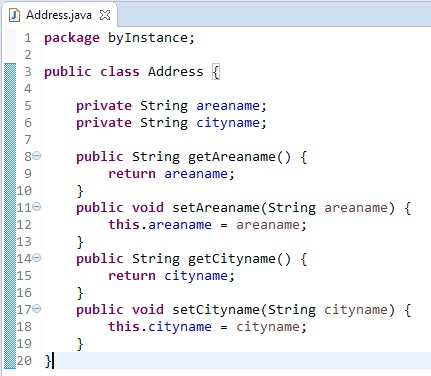
Files required : 1)Student.java

2) Address.java

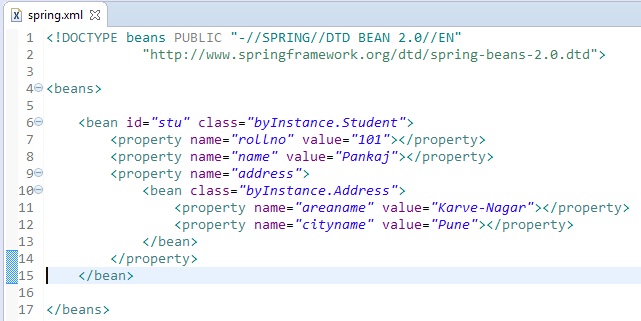
3) Test.java

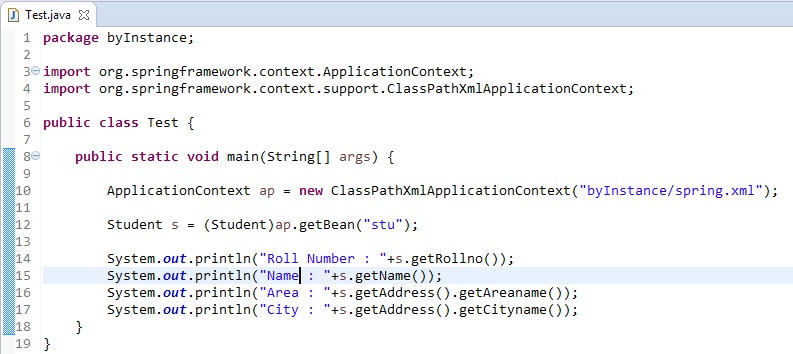
4)spring.xml +SpringJars





Here we are creating bean for address class inside bean of student class.





Output :

