Bộ môn: Kỹ Thuật Phần Mềm

Lập trình WWW *Java*

.

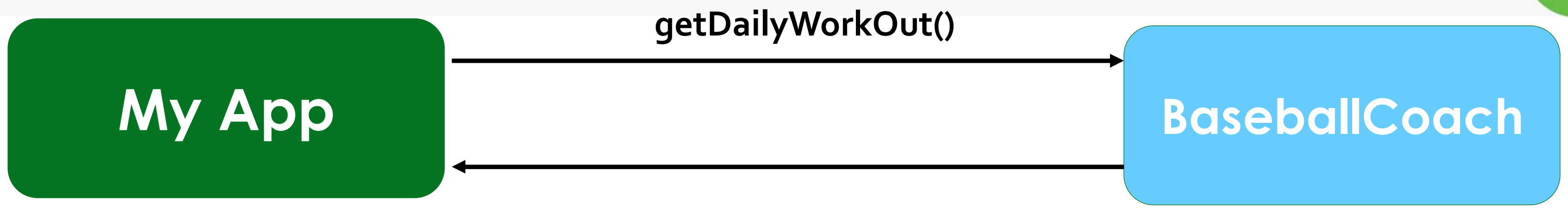
Spring IoC (Inversion of Control)

Inversion of Control (IoC)

Outsource to an object factory

The approach of outsourcing the construction and management of Objects

IoC – Coding Scenario

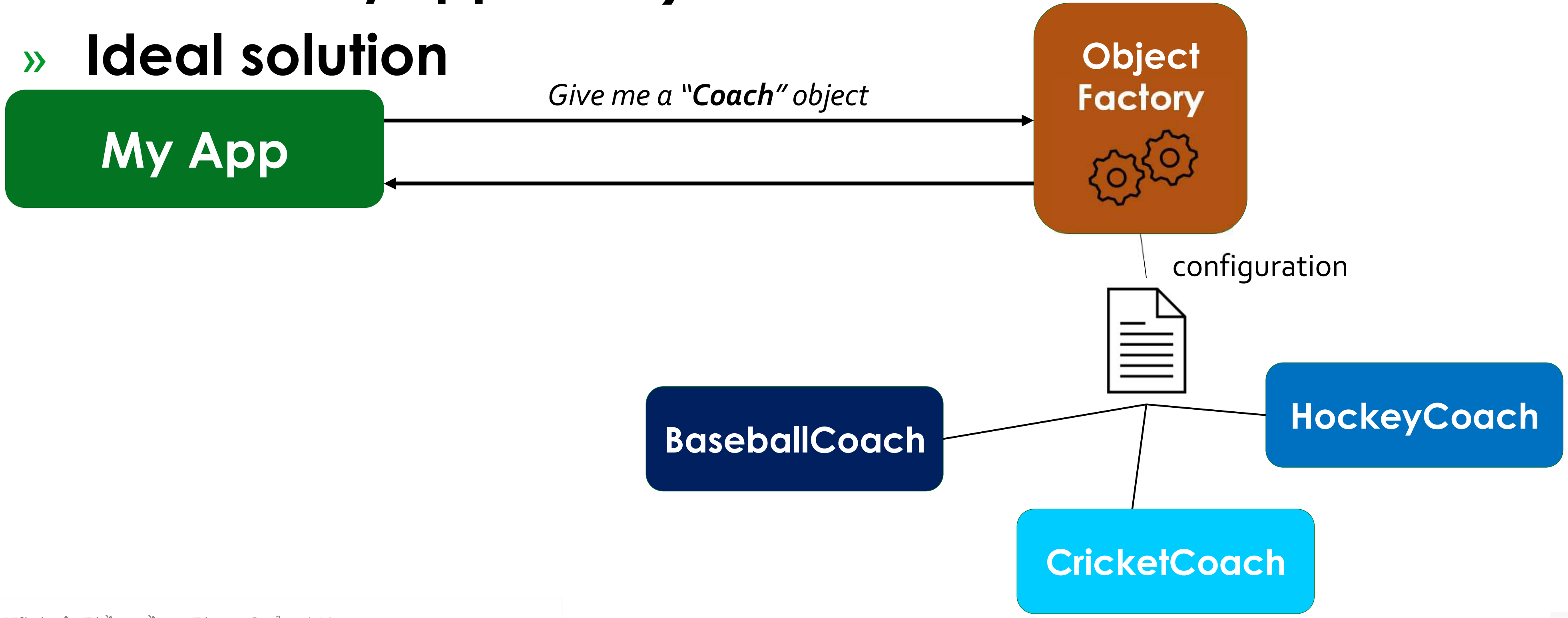


» **Easily change the coach** for other sport **Hockey, Cricket, Tennis etc.**

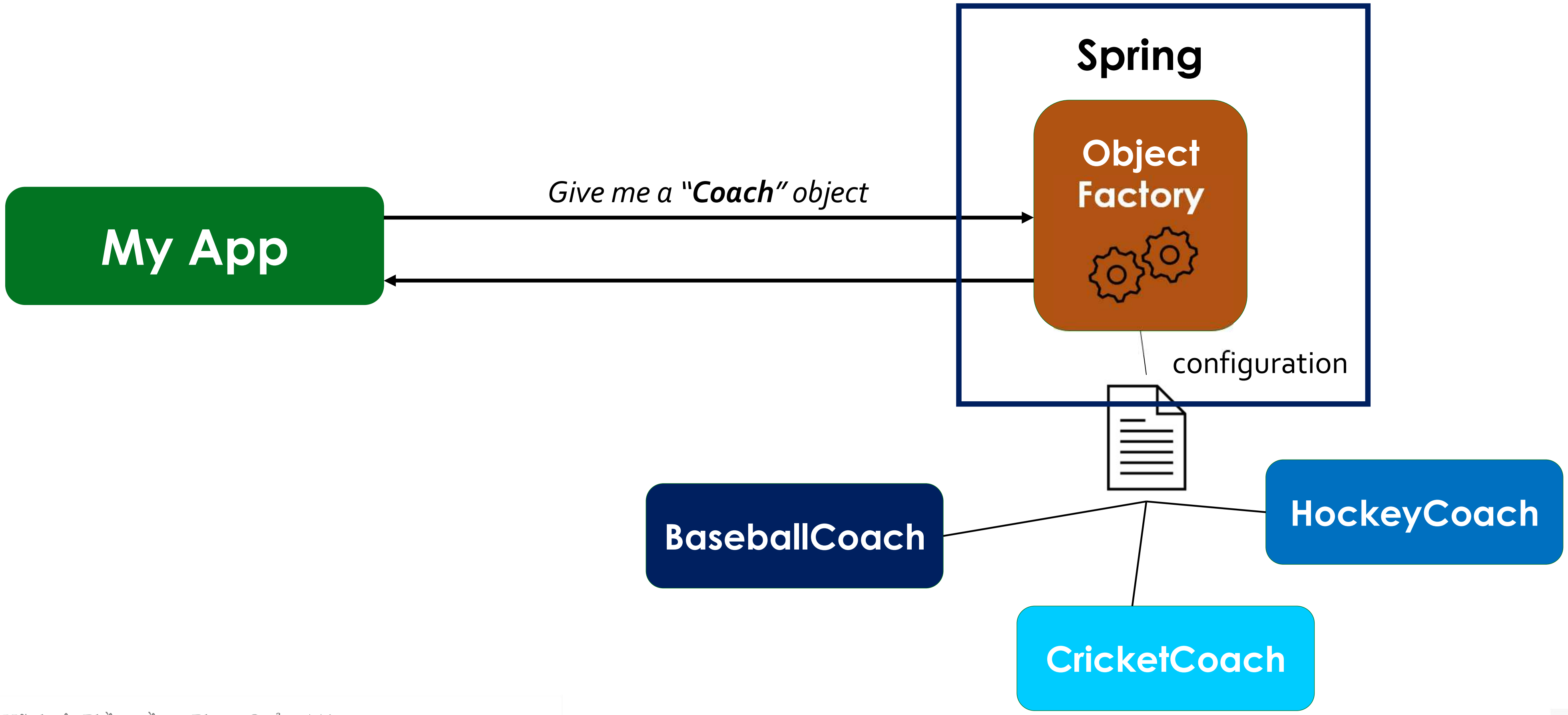
» The app should be **configurable**

IoC – Coding Scenario

- The app should be **configurable (I don't want to edit code in MyApp class)**
- **Ideal solution**



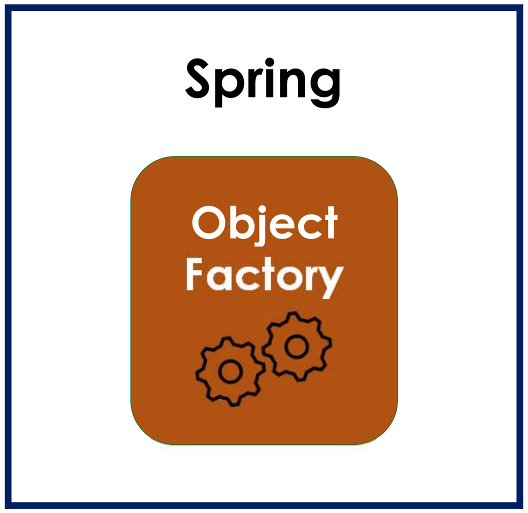
IoC – Coding Scenario



Spring Container

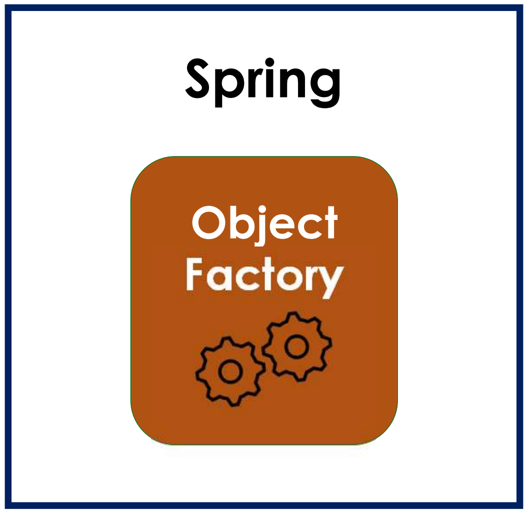
Primary functions:

- **Create** and **manage** objects (IoC)
- **Inject** object's **dependencies** (DI)



Configuring Spring Container

- » XML configuration file
- » Java Annotations
- » Java Source Code

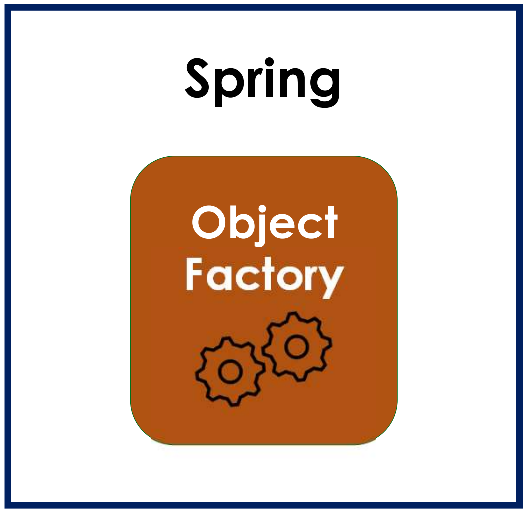


Spring Development Process

» **Configuring** your Spring beans

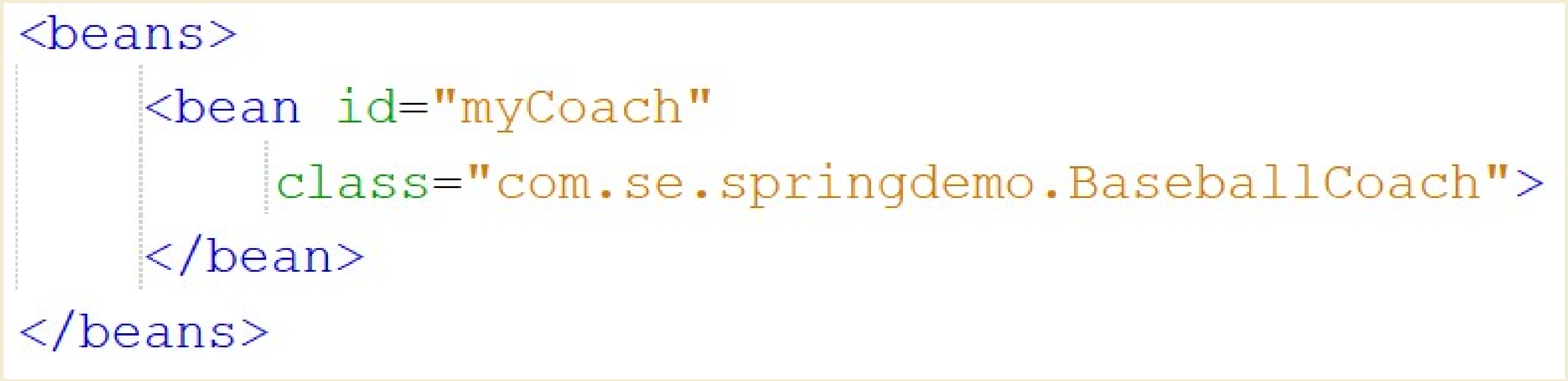
» **Creating** a Spring container

» **Retrieving** the beans from the container



Step 2: Configuring your Spring beans

File: applicationContext.xml:



```
<beans>
    <bean id="myCoach"
        class="com.se.springdemo.BaseballCoach">
    </bean>
</beans>
```

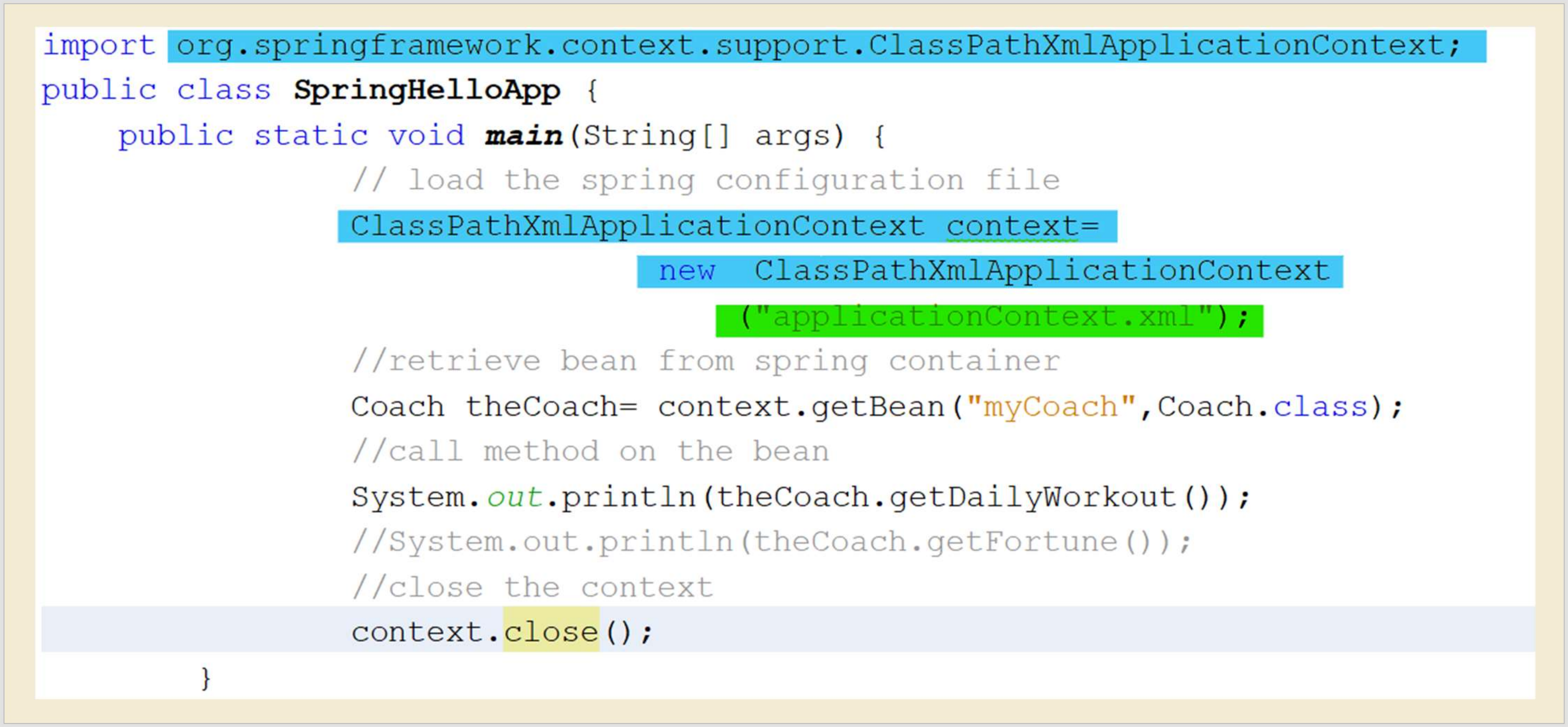
Step 2: Creating a Spring Container

Spring container is generally known as an **ApplicationContext**

Specialized implimentations:

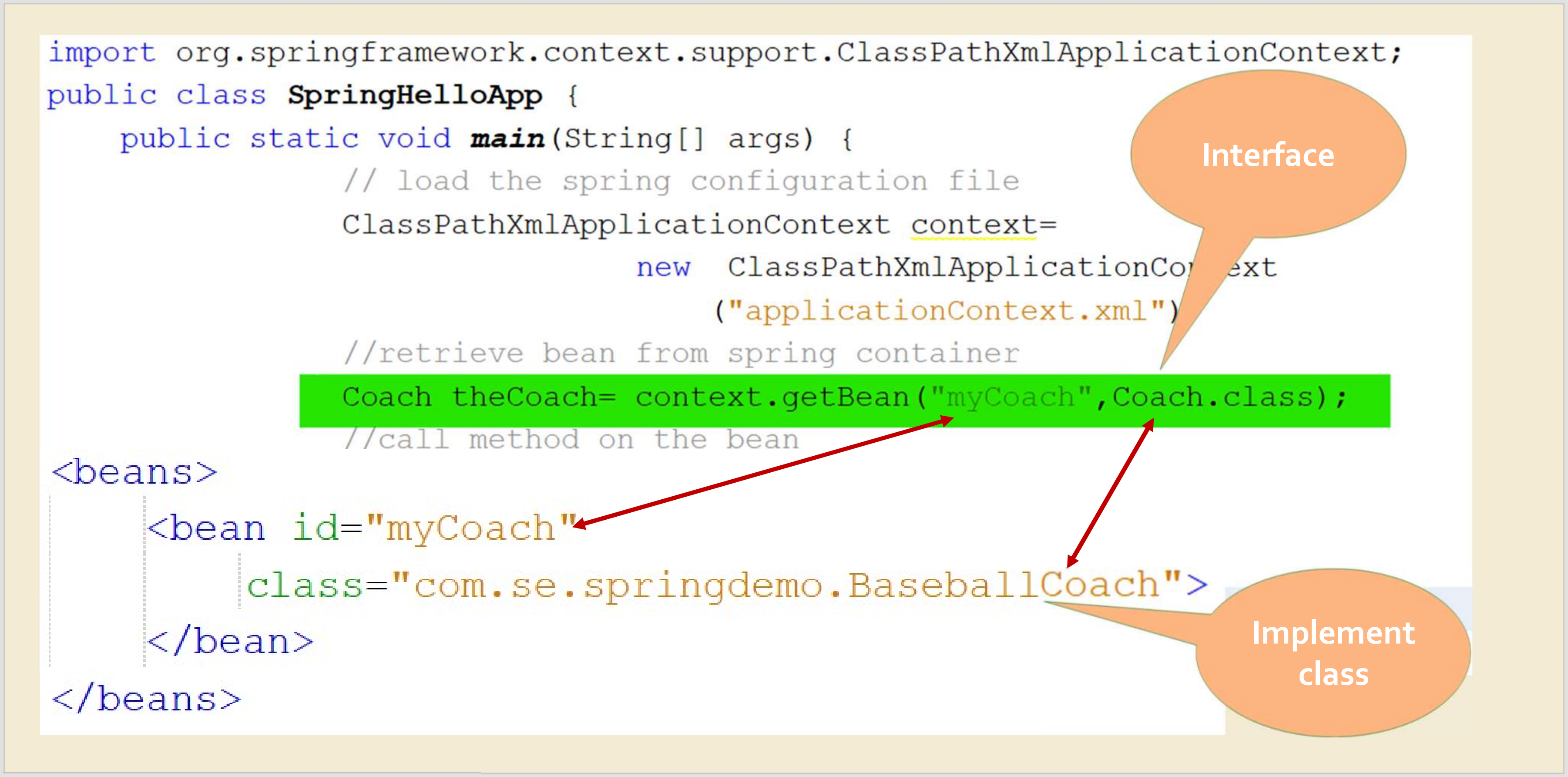
- ClassPathXmlApplicationContext
- AnnotationConfigApplicationContext
- GenericWebApplicationContext

Step 2: Creating a Spring Container



```
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class SpringHelloApp {
    public static void main(String[] args) {
                // load the spring configuration file
                ClassPathXmlApplicationContext context=
                                new ClassPathXmlApplicationContext
                                    ("applicationContext.xml");
                //retrieve bean from spring container
                Coach theCoach= context.getBean("myCoach",Coach.class);
                //call method on the bean
                System.out.println(theCoach.getDailyWorkout());
                //System.out.println(theCoach.getFortune());
                //close the context
                context.close();
        }
```

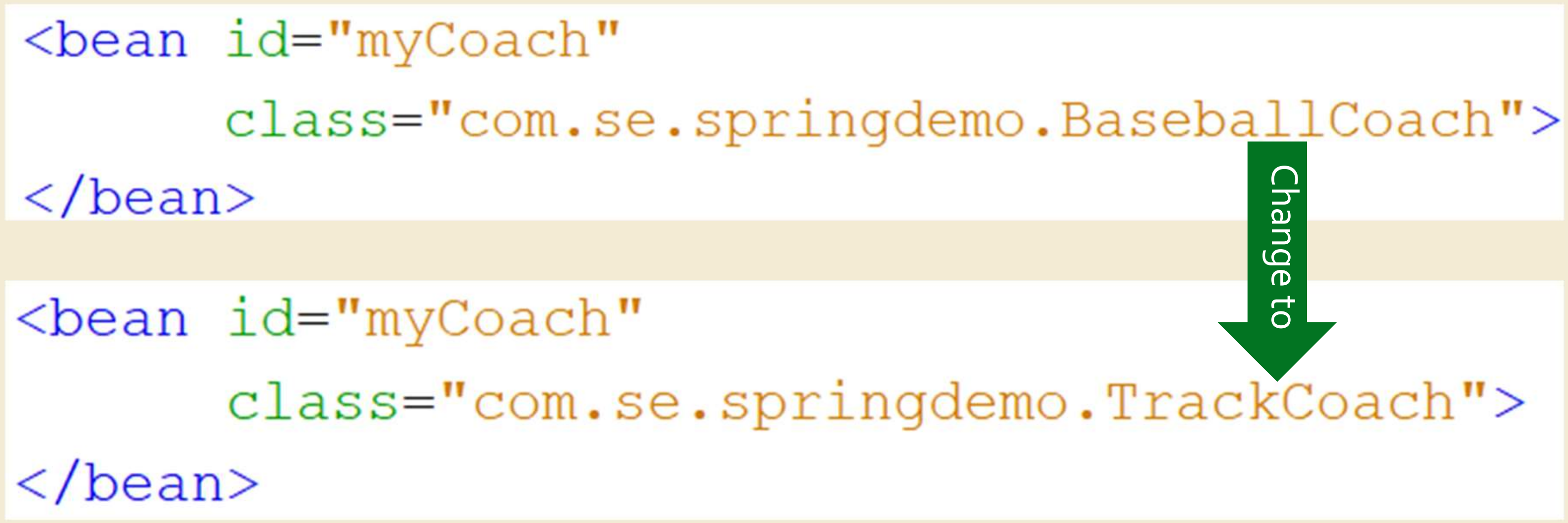
Step 3: Retrieving the beans from the container



```
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class SpringHelloApp {
    public static void main(String[] args) {
                // load the spring configuration file
                ClassPathXmlApplicationContext context=
                                new  ClassPathXmlApplicationCo  ext
                                    ("applicationContext.xml")
                //retrieve bean from spring container
                Coach theCoach= context.getBean("myCoach",Coach.class);
                //call method on the bean
<beans>
    <bean id="myCoach"
        class="com.se.springdemo.BaseballCoach">
    </bean>
</beans>
```

How to change another coach

Only change the implementation class (spring bean) in the config file



```xml
<bean id="myCoach"
      class="com.se.springdemo.BaseballCoach">
</bean>
```

```xml
<bean id="myCoach"
      class="com.se.springdemo.TrackCoach">
</bean>
```

QUESTIONS