

The provided code represents a simple example of using the Spring Framework's dependency injection feature. It consists of two Java classes (`A` and `B`) representing beans and an XML configuration file (`refconfig.xml`) defining the beans and their dependencies.

Class `A` has a `name` property and a reference to an instance of class `B` (`objB`). Class `B` has properties for `name`, `address`, and `phone`. Both classes have getter and setter methods for their properties.

The XML configuration file defines two beans: `bref` of type `B` and `aref` of type `A`. The `bref` bean is configured with specific values for its properties (`name`, `address`, and `phone`). The `aref` bean is configured with a `name` property value and a reference to the `bref` bean for the `objB` property.

The `Test` class contains the `main` method for testing. It creates an instance of `ApplicationContext` by loading the XML configuration file. It retrieves the `aref` bean from the context and prints out the values of `objB`'s properties (`name`, `phone`, and `address`), as well as the value of `name` in the `A` bean. It also prints the string representation of the `aref` object.

Overall, this code demonstrates the concept of dependency injection in Spring, where the dependencies between beans are managed by the framework based on the configuration file. The configuration file specifies how the beans are created and wired together, allowing for loose coupling and easier maintenance of the application.

Class A:

```
package com.springcore.ref;
public class A {

    private String name;
    private B objB;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public B getObjB() {
        return objB;
    }
    public void setObjB(B objB) {
        this.objB = objB;
    }

    public A() {
        super();
        // TODO Auto-generated constructor stub
    }

    @Override
    public String toString() {
        return "A [name=" + name + ", objB=" + objB + "]";
    }

}
```

Class B:

```
package com.springcore.ref;
```

```
public class B {

    private String name;
    private String address;
    private String phone;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }

    public B() {
        super();
        // TODO Auto-generated constructor stub
    }

    @Override
    public String toString() {
        return "B [name=" + name + ", address=" + address + ", phone=" + phone + "]";
    }

}
```

Config.xml:

```
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:p="http://www.springframework.org/schema/p"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
https://www.springframework.org/schema/beans/spring-beans.xsd">

    <bean class="com.springcore.ref.B" name="bref">
        <property name="name" value="B new name"/>
        <property name="address" value="B new Guwahati"/>
        <property name="phone" value="B new 8638370510"/>
    </bean>

    <bean class="com.springcore.ref.A" name="aref" p:name="A name" p:objB-ref="bref">
        <!-- <property name="name" value="A name"/>
        <property name="objB">
            <ref bean="bref"/>
        </property>
    </bean>

-->
</beans>
```

Main class:

```
package com.springcore.ref;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Test {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        ApplicationContext context = new ClassPathXmlApplicationContext("com/springcore/ref/refconfig.xml");
        A aObj = (A) context.getBean("aref");

        System.out.println(aObj.getObjB().getName());
        System.out.println(aObj.getObjB().getPhone());
        System.out.println(aObj.getObjB().getAddress());

        System.out.println(aObj.getName());

        System.out.println(aObj);
    }
}
```