

Main class:

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

public class Drawing {

    public static void main(String[] args) {
        ApplicationContext context = new
ClassPathXmlApplicationContext("NewFile.xml");
        Triangle1 obj =(Triangle1)context.getBean("Triangle");
        obj.draw();

    }

}
```

Class with ApplicationContextHolder interface

```
import org.springframework.beans.BeansException;
import org.springframework.beans.factory.BeanNameAware;
import org.springframework.context.ApplicationContext;
import org.springframework.context.ApplicationContextAware;

public class Triangle1 implements ApplicationContextAware, BeanNameAware {

    private Point pointA;
    private Point pointB;
    private Point pointC;
    private ApplicationContext context =null;

    public Point getPointA() {
        return pointA;
    }

    public void setPointA(Point pointA) {
        this.pointA = pointA;
    }

    public Point getPointB() {
        return pointB;
    }

    public void setPointB(Point pointB) {
        this.pointB = pointB;
    }

    public Point getPointC() {
        return pointC;
    }

}
```

```

    public void setPointC(Point pointC) {
        this.pointC = pointC;
    }

    public void draw() {
        System.out.println("Triangle drawn with ");
        System.out.println("point A =("+getPointA().getX()+",
"+getPointA().getY()+")");
        System.out.println("point B =("+getPointB().getX()+",
"+getPointB().getY()+")");
        System.out.println("point C =("+getPointC().getX()+",
"+getPointC().getY()+")");
    }

    @Override
    public void setApplicationContext(ApplicationContext context) throws
BeansException
    {
        this.context=context;
    }

    @Override
    public void setBeanName(String beanName) {
        System.out.println( "BeanName is " + beanName);
    }
}

```

Create a point class

```

public class Point {
    private int x;
    private int y;
    public int getX() {
        return x;
    }
    public void setX(int x) {
        this.x = x;
    }
    public int getY() {
        return y;
    }
    public void setY(int y) {
        this.y = y;
    }
}

```

Xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
           http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans.xsd">
    <bean id="Triangle" class="spring1.Triangle1">
        <property name="pointA">
            <bean class="spring1.Point">
                <property name="x" value="0"/>
                <property name="y" value="1"/>
            </bean>
        </property>
        <property name="pointB">
            <bean class="spring1.Point">
                <property name="x" value="20"/>
                <property name="y" value="10"/>
            </bean>
        </property>
        <property name="pointC">
            <bean class="spring1.Point">
                <property name="x" value="30"/>
                <property name="y" value="40"/>
            </bean>
        </property>
    </bean>
</beans>
```

output:

```
BeanName is Triangle
Triangle drawn with
point A =(0, 1)
point B =(20, 10)
point C =(30, 40)
```