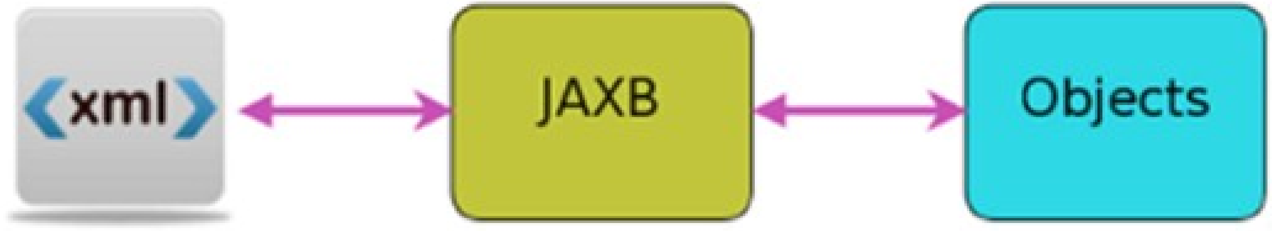
# JAXB basics

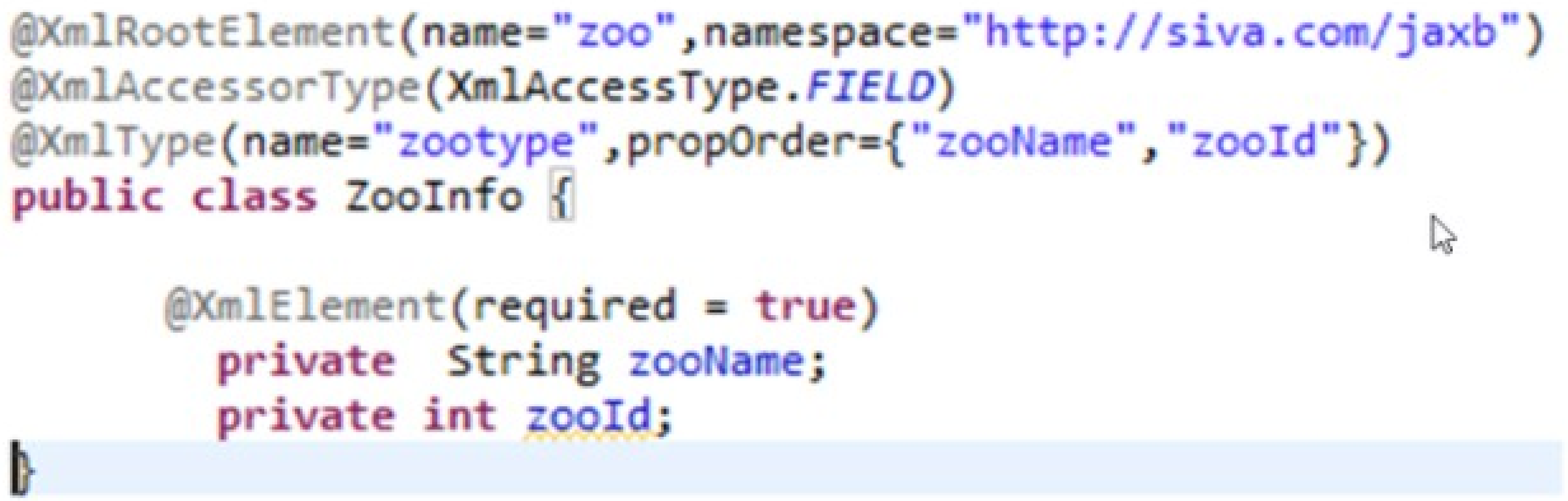
## What is JAXB ?

- JAXB is an acronym of **Java Architecture for XML Binding.**
- JAXB provides API to access and process a XML document.
- We can read or write XML files using JAXB.



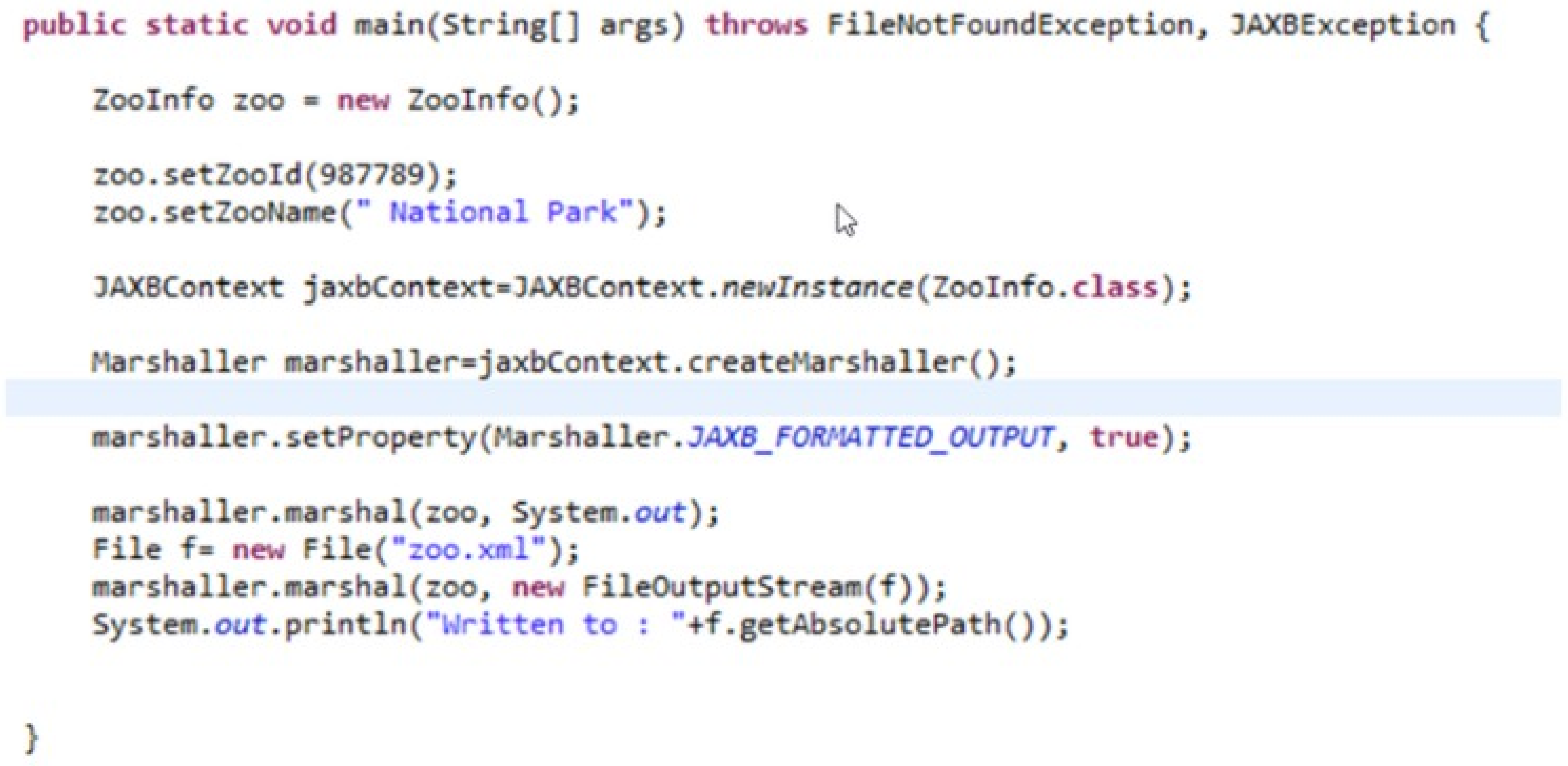
- Unlike SAX, DOM parsers,   to use JAXB the developer need not be aware of XML parsing techniques.

# Using Annotations

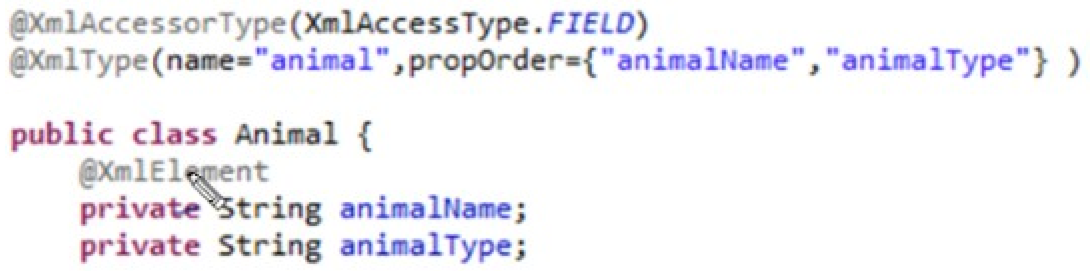


```
@XmlRootElement(name="zoo",namespace="http://siva.com/jaxb")
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name="zootype",propOrder={"zooName","zooId"})
public class ZooInfo {

	@XmlElement(required = true)
	  private  String zooName;
	  private int zooId;
}
```

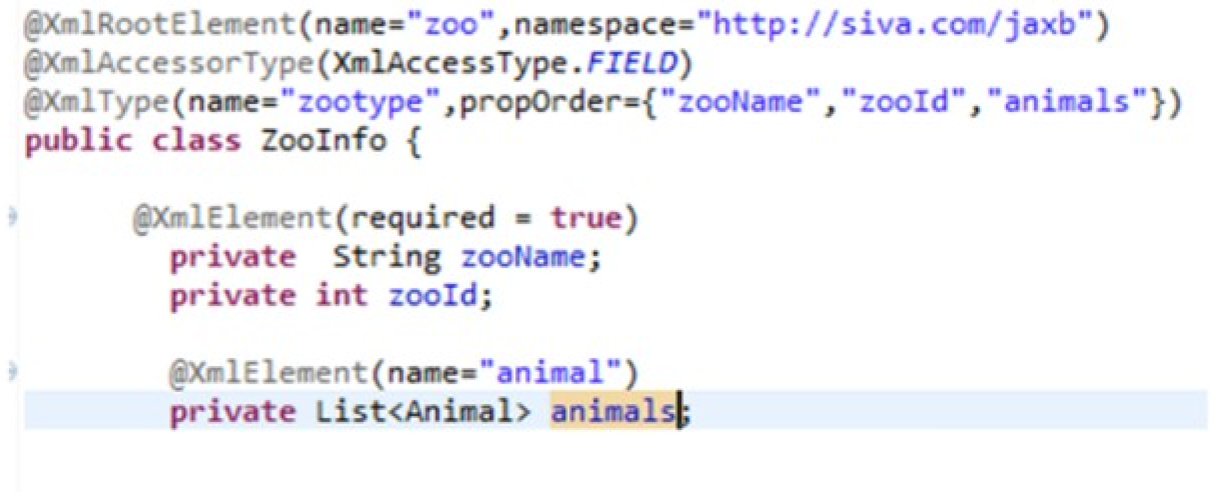
## Marshalling Example



```
public static void main(String[] args) throws FileNotFoundException, JAXBException {

    ZooInfo zoo = new ZooInfo();

    zoo.setZooId(987789);
    zoo.setZooName(" National Park");

    JAXBContext jaxbContext=JAXBContext.newInstance(ZooInfo.class);

    Marshaller marshaller=jaxbContext.createMarshaller();

    marshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);

    marshaller.marshal(zoo, System.out);
    File f= new File("zoo.xml");
    marshaller.marshal(zoo, new FileOutputStream(f));
    System.out.println("Written to : "+f.getAbsolutePath());


}
```

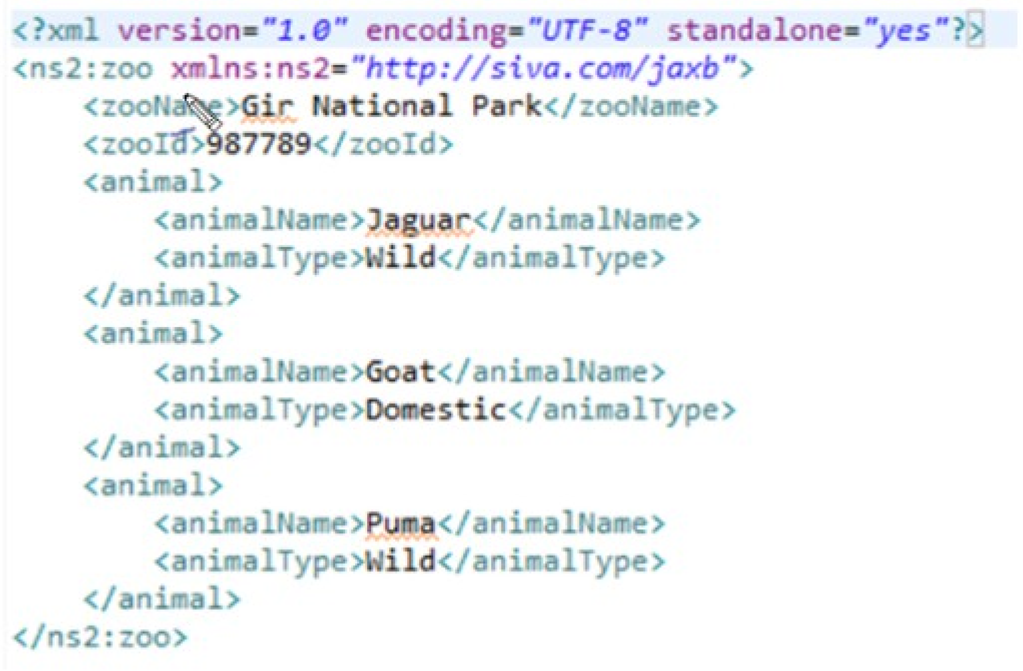


```java
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name="animal",propOrder={"animalName","animalType"} )

public class Animal {
    @XmlElement
    private String animalName;
    private String animalType;
```



```java
@XmlRootElement(name="zoo",namespace="http://siva.com/jaxb")
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name="zootype",propOrder={"zooName","zooId","animals"})
public class ZooInfo {

       @XmlElement(required = true)
         private  String zooName;
         private int zooId;

         @XmlElement(name="animal")
         private List<Animal> animals;
```

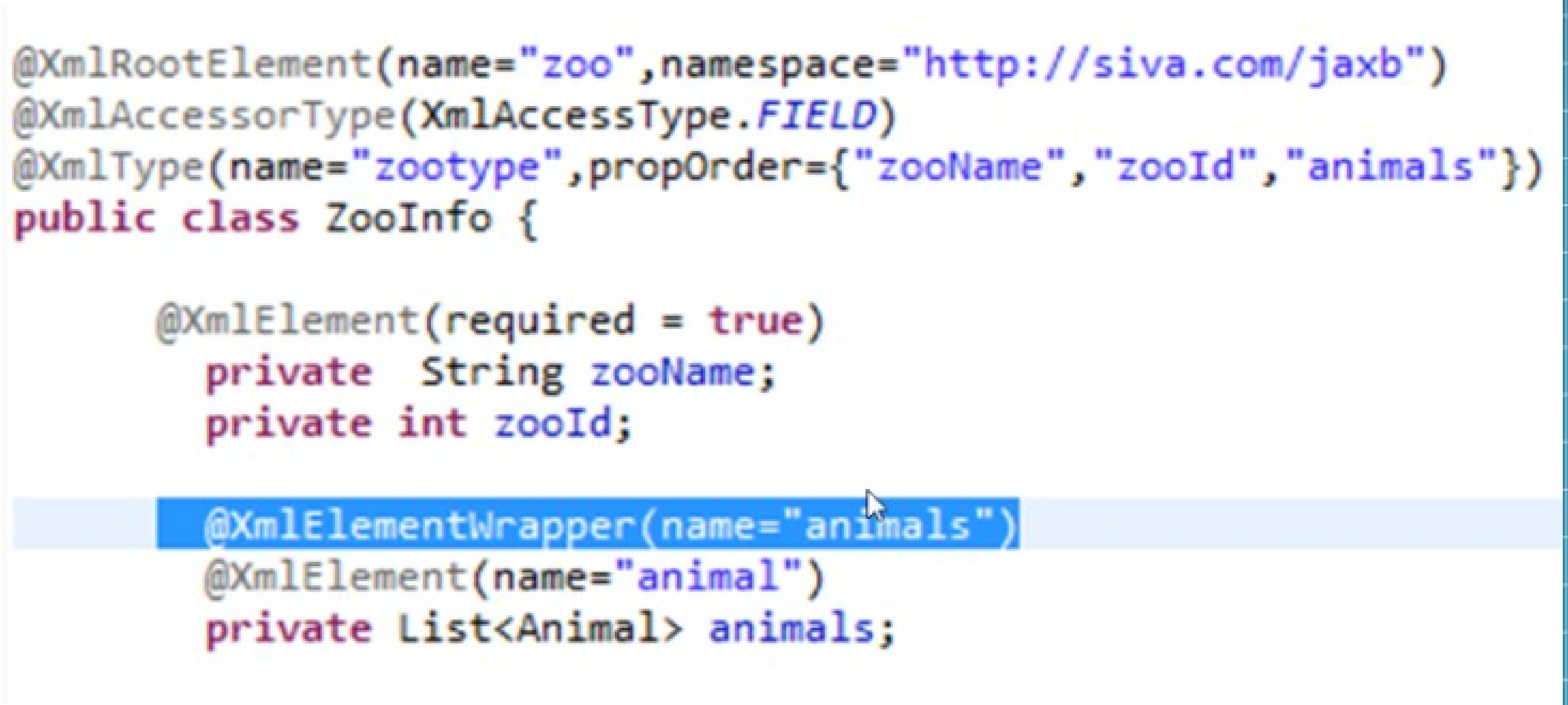
## Output after marshalling



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:zoo xmlns:ns2="http://siva.com/jaxb">
    <zooName>Gir National Park</zooName>
    <zooId>987789</zooId>
    <animal>
        <animalName>Jaguar</animalName>
        <animalType>Wild</animalType>
    </animal>
    <animal>
        <animalName>Goat</animalName>
        <animalType>Domestic</animalType>
    </animal>
    <animal>
        <animalName>Puma</animalName>
        <animalType>Wild</animalType>
    </animal>
</ns2:zoo>
```

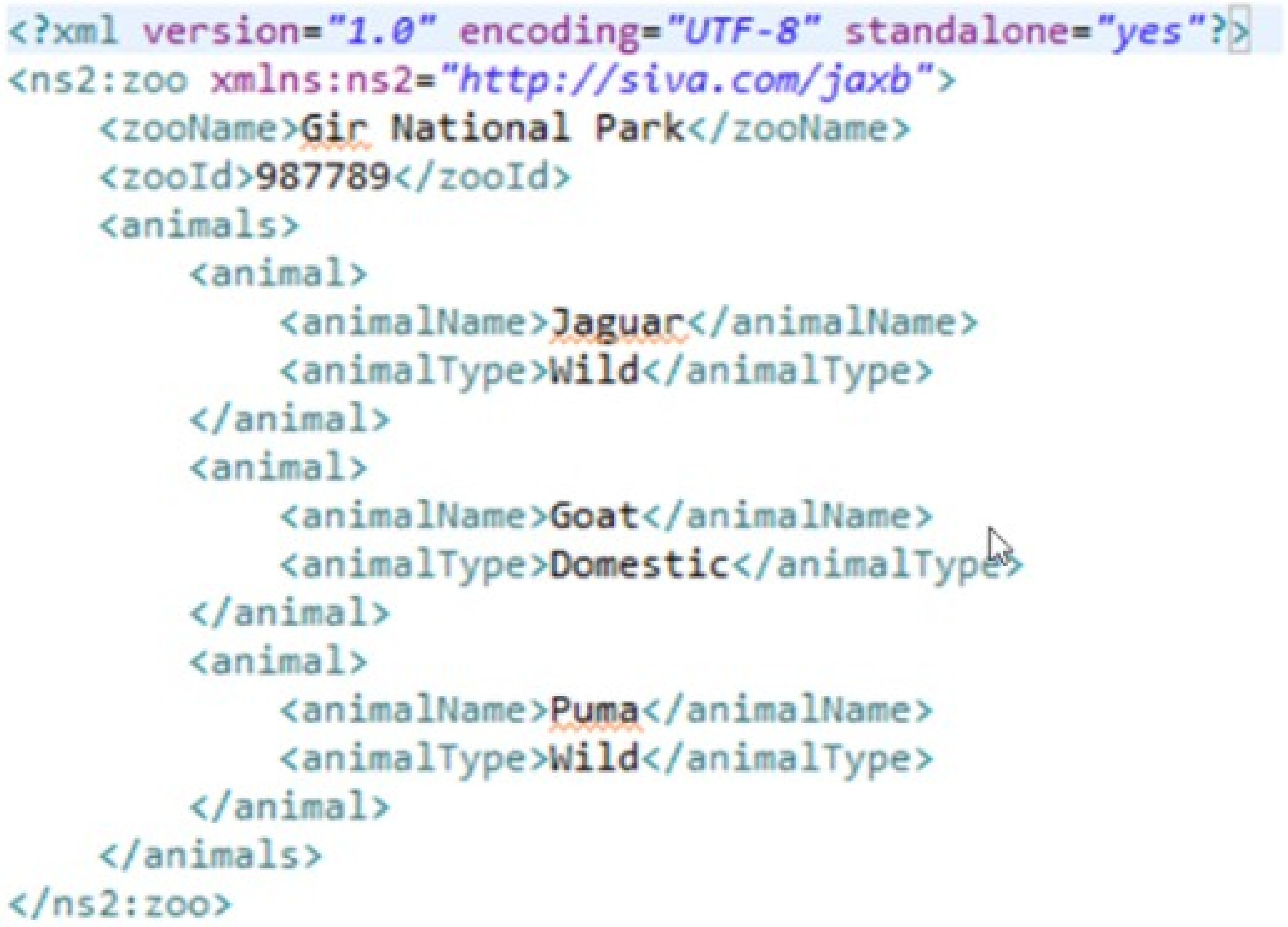
- It would look better if all animal tags are wrapped inside a tag like <animals>

# Using @XmlElementWrapper



```
@XmlRootElement(name="zoo",namespace="http://siva.com/jaxb")
@XmlAccessorType(XmlAccessType.FIELD)
@XmlType(name="zootype",propOrder={"zooName","zooId","animals"})
public class ZooInfo {

      @XmlElement(required = true)
        private  String zooName;
        private int zooId;

        @XmlElementWrapper(name="animals")
        @XmlElement(name="animal")
        private List<Animal> animals;
```

## Output after using @XmlElementWrapper



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:zoo xmlns:ns2="http://siva.com/jaxb">
    <zooName>Gir National Park</zooName>
    <zooId>987789</zooId>
    <animals>
        <animal>
            <animalName>Jaguar</animalName>
            <animalType>Wild</animalType>
        </animal>
        <animal>
            <animalName>Goat</animalName>
            <animalType>Domestic</animalType>
        </animal>
        <animal>
            <animalName>Puma</animalName>
            <animalType>Wild</animalType>
        </animal>
    </animals>
</ns2:zoo>
```