Persisting Data

Buddha Jyothiprasad
AUTHOR

@prbuddha http://controlspace.info

Outline

- Analyze options to store data
- Implement save on close
- Learn Method References
- Implement load tasks on startup

Numerous Strategies

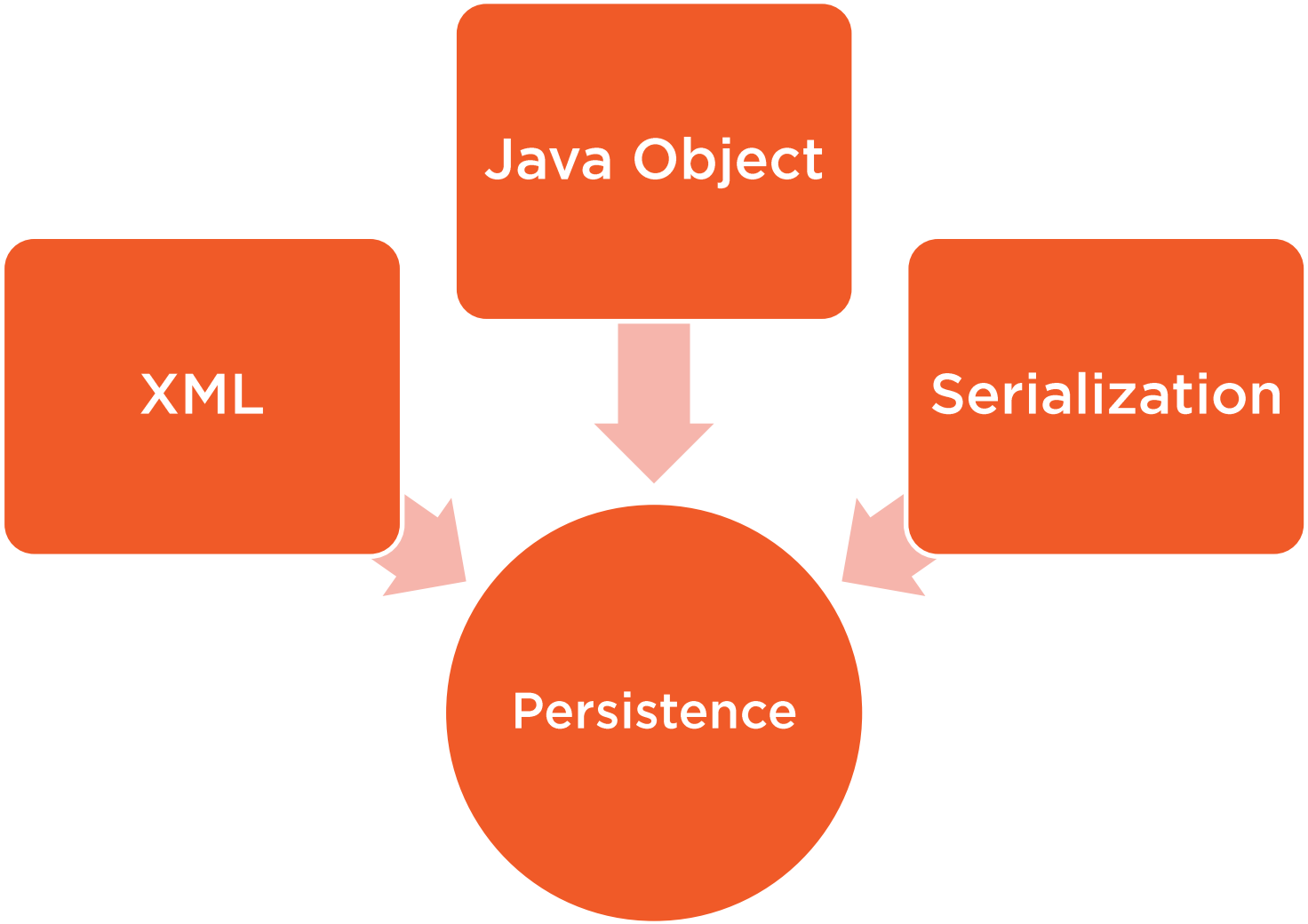
Each task in separate file

One XML holding all tasks

One CSV For storing tasks

Database to store in a table

Our Choice



```
FileOutputStream os = null;
os = new FileOutputStream(“tasks.xml”);

java.beans.XMLEncoder encoder=new XMLEncoder(os);
encoder.writeObject(tasksMap);

encoder.close();
```

XML Serialization

```
FileInputStream is=null;
is = new FileInputStream(“tasks.xml”);

XMLDecoder decoder=new XMLDecoder(is);
tasksMap=(HashMap<Integer, Task>)decoder.readObject();

decoder.close();
```

XML De-Serialization

Demo

Window Close Event

Method References

Writing tasks to XML files

Demo

Reading data on startup

Set initial tasks in Controller

Summary

- Serialization for rescue
- Window Close Event
- Method References
- Save Tasks Map as an XML file
- Read data back on startup

Next Steps

Official Resources

- http://docs.oracle.com/javase/8/javase-clienttechnologies.htm
- https://docs.oracle.com/javase/8/javafx/api
- Ensemble Demo Application

Online Resources

- http://tutorials.jenkov.com
- http://code.makery.ch

Books

- Java FX 8: Introduction by Example, Apress publications