Object Oriented Analysis and Design with Java

UE19CS353

Prof. Sindhu R Pai
Department of Computer Science and Engineering

Acknowledgements: Significant portions of the information in the slide sets presented through the course in the class, are extracted from the prescribed text books, information from the Internet. Since these are only intended for presentation for teaching within PESU, there was no explicit permission solicited. We would like to sincerely thank and acknowledge that the credit/rights remain with the original authors/creators only

PES UNIVERSITY ONLINE

UE19CS353: Object Oriented Analysis and Design using Java

Serialization

Prof. Sindhu R Pai

Department of Computer Science and Engineering

Agenda

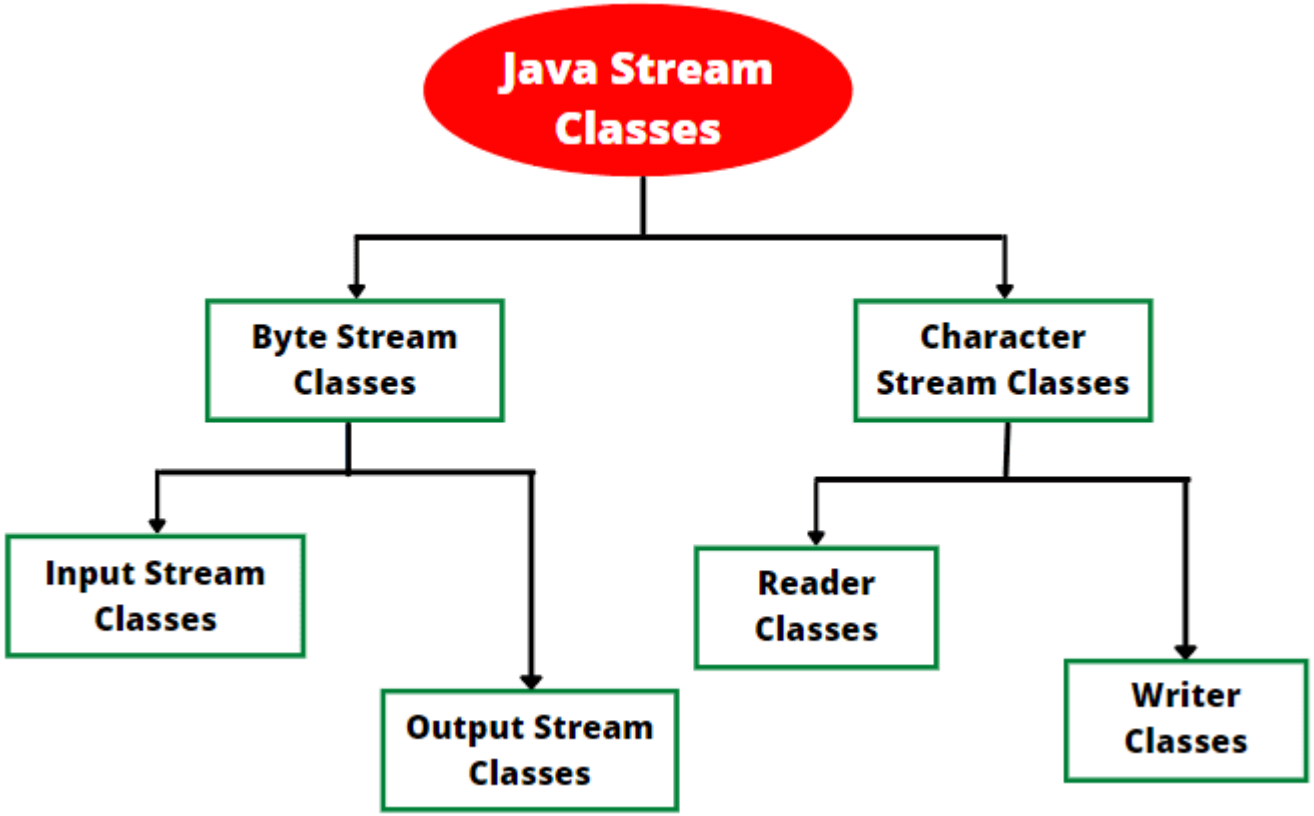
- Introduction
- References

Introduction

- A mechanism to transform an **object into a byte stream**. This object written into a byte stream does not contain the actual code.
- To do this, the class of that object needs to implement the interface **Serializable**.
- Uses **reflection** internally to scrape all the data from the object's fields that need to be serialized.
- Private and final fields are also included.
- If a field contains an object, that object is serialized recursively.
- Getters and setters are not used when serializing an object

Byte Stream

- Nothing but an ordered sequence of bytes
- We can **store video, audio, characters**, etc., by using ByteStream classes.
- These classes are part of the **java.io package**.



Serializable interface

- Is a marker interface (has no data member and method)
- It is used to "mark" Java classes so that the objects of these classes may get a certain capability
- The Serializable interface must be implemented by the class whose object needs to be persisted
- The String class and all the wrapper classes implement the java.io.Serializable interface by default.

Serialization

- The ObjectOutputStream class is used to write primitive data types, and Java objects to an OutputStream.
- Only objects that support the java.io.Serializable interface can be written to streams.

Constructor

1) public ObjectOutputStream(OutputStream out) throws IOException {}	It creates an ObjectOutputStream that writes to the specified OutputStream.

Important Methods

Method	Description
1) public final void writeObject(Object obj) throws IOException {}	It writes the specified object to the ObjectOutputStream.
2) public void flush() throws IOException {}	It flushes the current output stream.
3) public void close() throws IOException {}	It closes the current output stream.

De-Serialization

- An ObjectInputStream deserializes objects and primitive data written using an ObjectOutputStream.

Constructor

1) public ObjectInputStream(InputStream in) throws IOException {}	It creates an ObjectInputStream that reads from the specified InputStream.

Important Methods

Method	Description
1) public final Object readObject() throws IOException, ClassNotFoundException{}	It reads an object from the input stream.
2) public void close() throws IOException {}	It closes ObjectInputStream.

Coding part

- Create an array of 4 objects and write to a file using serialization concept.

Deserialize the contents of the file and display it on the terminal.

•Need to code this using ObjectOutputStream and inputstream

THANK YOU

Prof. Sindhu R Pai

Department of Computer Science and Engineering

sindhurpai@pes.edu

+91 8277606459