JAC444 / BTP400 Course Object-Oriented Software Development II - Java

Serialization

Segment 3

Input Stream / Output Stream

In this segment you will be learning about:

- Serializable Objects
- FileInputStream / FileOutputStream
- ObjectInputStream / ObjectOutputStream

Object Serialization

- The key to writing an object is to represent its state in a serialized form sufficient to reconstruct the object as it is read.
- Reading and writing objects is a process called object serialization.
- One you need to know how to serialize objects by writing them to an *ObjectOutputStream* and reading them in again using an *ObjectInputStream*.
- An object is serializable only if its class implements the Serializable interface.

```
FileOutputStream out = new FileOutputStream(“name");
ObjectOutputStream s = new ObjectOutputStream(out);
s.writeObject("Today");
s.writeObject(new Date());
s.flush();
```

ObjectInputStream

```
FileInputStream in = new FileInputStream("name");
ObjectInputStream s = new ObjectInputStream(in);
String today = (String)s.readObject();
Date date = (Date)s.readObject();
```

The `readObject` method deserializes the next object in the stream and traverses its references to other objects recursively to deserialize all objects that are reachable from it. In this way, it maintains the relationships between the objects.

`ObjectInputStream` stream implements the `DataInput` interface that defines methods for reading primitive data types.

Customizing Serialization

- An object is serializable only if its class implements the `Serializable` interface.

- `Serializable` is an empty interface, it doesn't contain any method declarations. It is what is called a marker interface.

- The serialization of instances of this class are handled by the `defaultWriteObject` method of `ObjectOutputStream`.

- This method automatically writes out everything required to reconstruct an instance of the class, including the following:
  - Class of the object.
  - Class signature.
  - Values of all non-***transient*** and and non-***static*** members, including members that refer to other objects.

- One can customize serialization for his/her classes by providing two methods for it: `writeObject` and `readObject`.

Externalizable Interface

For complete, explicit control of the serialization process, a class must implement the `Externalizable` interface

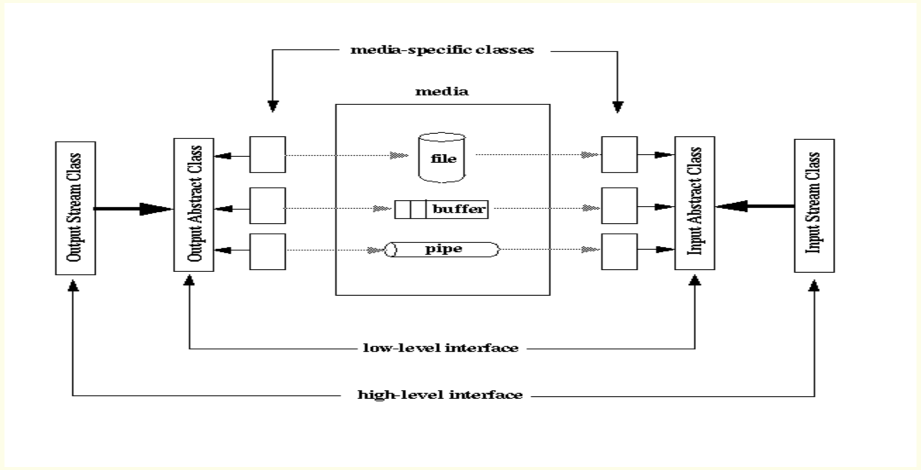
```
package java.io;
public interface Externalizable extends Serializable   {
public void writeExternal(ObjectOutput out) throws IOException;
public void readExternal(ObjectInput in) throws IOException,
                                java.lang.ClassNotFoundException;
}
```

Particularly sensitive classes should not be serialized. To accomplish this, the class should not implement either the `Serializable` or `Externalizable` interface

Conclusions

After completion of this segment you should know:

- How to serialize Java objects.
- How to deserialize and construct Java objects
- How to read/write object using serialization in general.



Conclusions

After completion of this lesson you should know:

Write Java programs using I/O classes