Marker interface in Java

Difficulty Level : Medium Last Updated : 12 Jul, 2021

It is an empty interface (no field or methods). Examples of marker interface are Serializable, Cloneable and Remote interface. All these interfaces are empty interfaces.

```
public interface Serializable
{
  // nothing here
}
```

Examples of Marker Interface which are used in real-time applications :

1. **Cloneable interface** : Cloneable interface is present in java.lang package. There is a method clone() in Object class. A class that implements the Cloneable interface indicates that it is legal for clone() method to make a field-for-field copy of instances of that class.
   Invoking Object's clone method on an instance of the class that does not implement the Cloneable interface results in an exception CloneNotSupportedException being thrown. By convention, classes that implement this interface should override Object.clone() method.
   Refer here for more details.

```
// Java program to illustrate Cloneable interface
import java.lang.Cloneable;

// By implementing Cloneable interface
// we make sure that instances of class A
// can be cloned.
class A implements Cloneable
{
    int i;
    String s;

    // A class constructor
    public A(int i,String s)
    {
        this.i = i;
        this.s = s;
    }
    // Overriding clone() method
    // by simply calling Object class
    // clone() method.
    @Override
    protected Object clone()
    throws CloneNotSupportedException
    {
        return super.clone();
    }
}

public class Test
{
    public static void main(String[] args)
        throws CloneNotSupportedException
    {
        A a = new A(20, "GeeksForGeeks");

        // cloning 'a' and holding
        // new cloned object reference in b

        // down-casting as clone() return type is Object
        A b = (A)a.clone();

        System.out.println(b.i);
        System.out.println(b.s);
    }
}
```

Output:

```
20
GeeksForGeeks
```

2. **Serializable interface** : Serializable interface is present in java.io package. It is used to make an object eligible for saving its state into a file. This is called Serialization.
   Classes that do not implement this interface will not have any of their state serialized or deserialized. All subtypes of a serializable class are themselves serializable.

```
// Java program to illustrate Serializable interface
import java.io.*;
```

```
// By implementing Serializable interface
// we make sure that state of instances of class A
// can be saved in a file.
class A implements Serializable
{
    int i;
    String s;

    // A class constructor
    public A(int i,String s)
    {
        this.i = i;
        this.s = s;
    }
}

public class Test
{
    public static void main(String[] args)
      throws IOException, ClassNotFoundException
    {
        A a = new A(20,"GeeksForGeeks");

        // Serializing 'a'
        FileOutputStream fos = new FileOutputStream("xyz.txt");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(a);

        // De-serializing 'a'
        FileInputStream fis = new FileInputStream("xyz.txt");
        ObjectInputStream ois = new ObjectInputStream(fis);
        A b = (A)ois.readObject();//down-casting object

        System.out.println(b.i+" "+b.s);

        // closing streams
        oos.close();
        ois.close();
    }
}
```

Output:

```
20 GeeksForGeeks
```

3. **Remote interface** : Remote interface is present in java.rmi package. A remote object is an object which is stored at one machine and accessed from another machine. So, to make an

object a remote object, we need to flag it with Remote interface. Here, Remote interface serves to identify interfaces whose methods may be invoked from a non-local virtual machine.Any object that is a remote object must directly or indirectly implement this interface. RMI (Remote Method Invocation) provides some convenience classes that remote object implementations can extend which facilitate remote object creation.

This article is contributed by **Gaurav Miglani**. If you like GeeksforGeeks and would like to contribute, you can also write an article using write.geeksforgeeks.org or mail your article to review-team@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.
Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.