

1

↳ import java.rmi.Remote; → interface

↳ import java.rmi.RemoteException; → class

↳ allows java objects to invoke methods on objects located on remote systems

RMI enables distributed computing and remote communication between Java programs.

Interface extends Remote indicating that it defines methods that can be invoked remotely

import java.rmi.server.*; → importing all the classes and interfaces both within java.rmi.server package including 'UnicastRemoteObject' class

By extending 'UnicastRemoteObject' the class inherits the functionality to export itself as a remote object

Unicast Remote Object : Primary purpose is to export a remote object and make it available for remote method invocations from client objects.

When a remote object is extended from 'UnicastRemoteObject', it becomes capable of being accessed and invoked remotely by clients using RMI

Multicast Remote Object : It extends UnicastRemoteObject and adds additional functionality for multicast communication.

By extending MulticastRemoteObject a remote object gains the ability to send multicast messages to multiple clients

Registry reg = LocateRegistry.createRegistry(5000);

(2)

↳ This line creates an RMI registry on port 5000. The RMI registry acts as a central repository where objects can be bound to names and located by clients.

PalindromeInterface obj = new PalindromeClass();

↳ An instance of PalindromeClass is created and assigned to PalindromeInterface reference variable 'obj'.

reg.bind("XYZ", obj);

↳ The obj is bound to the registry with the name "XYZ". This allows clients to locate and access the "PalindromeClass" instance remotely using name "XYZ".

PalindromeInterface obj1 = (PalindromeInterface) Naming.lookup

("rmi://localhost:5000/XYZ");

↳ This line uses the 'Naming.lookup()' method to lookup the remote object with the name "XYZ" in the RMI registry located at "rmi://localhost:5000/". The lookup() method returns a remote object reference that is cast to the 'PalindromeInterface' type and assigned to the 'obj1' variable. This allows the client to invoke methods on the remote object.

PalindromeClass() throws RemoteException {

 super();

}

↳ It calls the constructor of the "Unicast Remote Object" class using super() which sets up the necessary RMI related functionality.