```java
/*
 * Cette classe se connecte à un serveur de temps pour obtenir
 * la date et l'heure courante.
 *
 * Usage : java Client host port
 */
import java.net.*;
import java.io.*;
import java.util.Date;

public class Client {
    public static void main (String [] args) throws Exception {
        int portDest = Integer.parseInt(args[1]);
        // création de la socket sur le 1er port libre de la machine locale
        Socket socket = new Socket();
        socket.bind(new InetSocketAddress("localhost", 0));
        // Connexion avec la machine distante
        socket.connect(new InetSocketAddress(args[0], portDest));
        // récupérer la date
        ObjectInputStream in = new ObjectInputStream(socket.getInputStream());
        System.out.println(in.readObject());
        in.close();
        socket.close();
    }
}
```



```java
/**
 * Cette classe représente une serveur de temps. Elle accepte des
 * connexions de clients et fournir en retour la date et l'heure
 * courante. Le port est passé en paramètre.  Modèle client/serveur
 *
 *  Usage : java TimeServer port
 */
import java.net.*;
import java.io.*;
import java.util.*;

public class TimeServer {
    public static void main (String [] args) throws IOException {
        int port = Integer.parseInt(args[0]);
        // creation de la socket
        ServerSocket socketServeur = new ServerSocket(port);
        while (true) {
            // attendre la prochaine connexion
            Socket socketClient = socketServeur.accept();
            // gérer le client dans un thread
            new GestionDuClient(socketClient);
        }
    }
}

/**
 * Cette classe gère la connexion avec un client particulier et lui renvoie
 * la date et l'heure
 */
class GestionDuClient implements Runnable {
    private Thread proc = new Thread(this);
    private Socket socket;

    public GestionDuClient(Socket s) {
        socket = s;
        proc.start();
    }

    public void run() {
        try (ObjectOutputStream dout = new ObjectOutputStream(this.socket.getOut
putStream()))
            {
                dout.writeObject(new Date());
                this.socket.close();
            }
        catch (IOException e) { System.err.println(e); }
    }
}
```