

PROTOCOLLO UDP: INVIO DI OGGETTI

Laura Ricci

24 Ottobre 2005

UDP: SERIALIZZAZIONE DI OGGETTI

Esercizio:

Il Server *ServerScuola* mantiene informazioni circa le assenze giustificate e quelle non giustificate effettuate dagli alunni e le invia ad un *ClientGenitore*, tramite un collegamento UDP.

- definizione di un oggetto messaggio serializzabile contenente due interi (assenze giustificate, assenze non giustificate)
- spedizione/ricezione oggetti mediante *ObjectInput/OutputStream*

Nota: La versione presentata è notevolmente semplificata per mettere in evidenza i problemi principali. Sviluppare la versione completa.

SERIALIZAZIONE DEGLI OGGETTI

```
import java.io.*;  
class messaggio implements Serializable  
    {private int nassenzeg;  
      private int nassenzeng;  
      messaggio(int x, int y)  
          {this.nassenze = x;  
            this.nassenzeng= y; }  
      int getx ( )  
          {return nassenzeg;};  
      int gety ( )  
          {return nassenzeng;}  
    }
```

IL MITTENTE – SERVER ASSENZE

```
import java.net.*;
import java.io.*;
import java.util.*;

public class ServerAssenze
{
    public static void main(String Args[]) throws Exception
    { /* inizializzazioni indirizzo e porta del client-destinatario
      InetAddress ia = InetAddress.getByName("LocalHost");
      int port= 1300;
      DatagramSocket ds = new DatagramSocket( );
      ByteArrayOutputStream bout=new ByteArrayOutputStream();
      byte [] data=new byte[256] ;
      DatagramPacket dp= new DatagramPacket(data, data.length, ia, port);
```

IL MITTENTE - SERVER ASSENZE

```
for (int i=1;i<10;i++)
{
    int na=i;
    int nr=i;
    messaggio m=new messaggio(na,nr);
    ObjectOutputStream dout = new ObjectOutputStream(bout);
    dout.writeObject(m);
    dout.flush();
    data =bout.toByteArray();
    dp.setData(data);
    dp.setLength(data.length);
    ds.send(dp);
    bout.reset();
}
}
```

IL MITTENTE - SERVER ASSENZE

Note:

- E' necessario costruire un nuovo *ObjectOutputStream* per ogni oggetto inviato.
- provare a spostare l'istruzione
ObjectOutputStream dout = new *ObjectOutputStream*(bout)
fuori dal ciclo *for*: il destinatario non riesce a ricostruire l'oggetto serializzato (*StreamCorruptedException*).
- posso eliminare la *bout.reset()* se sposto l'istruzione
ByteArrayOutputStream bout=new *ByteArrayOutputStream*();
all'interno del ciclo *for*

IL CLIENT ASSENZE

```
import java.net.*;
```

```
import java.io.*;
```

```
import java.util.*;
```

```
public class ClientAssenze{
```

```
public static void main (String Args[]) throws Exception
```

```
{ /* inizializzazioni – indirizzo e porta su cui riceve
```

```
InetAddress ia = InetAddress.getByName("LocalHost");
```

```
int port=1300;
```

```
DatagramSocket ds=new DatagramSocket(port);
```

```
byte buffer[]=new byte[256];
```

```
DatagramPacket dpin= new DatagramPacket(buffer, buffer.length);
```

IL CLIENT ASSENZE

```
for (int i=1;i<10;i++)  
  
    {ds.receive(dpin);  
    ByteArrayInputStream bais= new ByteArrayInputStream(dpin.getData());  
    ObjectInputStream ois= new ObjectInputStream(bais);  
    messaggio m = (messaggio) ois.readObject();  
    System.out.println(m.getx());  
    System.out.println(m.gety());  
    }
```

Provare a vedere cosa accade se si elimina dal server l'istruzione *bout.reset()*;