User Data Protocol (2)

```
.....
try {
    Thread.sleep(100000);

    so = new DatagramSocket(porta);

    so.receive(pacRic);
}
catch......
```

cosa succede se un altro processo fa la send quando questo processo e' nella sleep?

```
......
try {
    so = new DatagramSocket(porta);

    Thread.sleep(100000);

    so.receive(pacRic);
}
catch......
```

cosa succede se un altro processo fa la send quando questo processo e' nella sleep?

Filtraggio pacchetti in entrata

```
.......
so = new DatagramSocket(portaloc);

so.connect(hostrem, portarem);

while (....) {

    so.receive(pacRic);

}

if (so.isConnected()) {

    System.out.println("porta remota " + so.getPort() +
    " host remoto " + so.getInetAddress().getHostName());

    so.disconnect();
}
```

scarta tutti i pacchetti che non provengono da hostrem e portarem

Input e Output Streams

FileOutputStream

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

byte[] buf = (new String("ciao a tutti")).getBytes();
byte[] buf2 = (new String("arrivederci")).getBytes();

try {
    FileOutputStream fos = new FileOutputStream("prv.txt");

    fos.write(buf, 0, buf.length);
    fos.write(buf2, 0, buf2.length);

    fos.close();
}
catch(FileNotFoundException e) {
    System.out.println("errore di creazione file");}
catch(IOException e) {
    System.out.println("errore di IO sulla write");}
```

Filtro DataOutputStream

```
import java.io.*;

.....

byte[] buf = (new String("ciao a tutti")).getBytes();
int i = 33;
long l = 55l;
boolean bo = true;
float f = 44.4f;
double d= 33.3d;

try {
    DataOutputStream dos = new DataOutputStream(
                           new FileOutputStream("prv"));

    dos.write(buf, 0, buf.length);
    dos.writeByte(i);
    dos.writeChar(i);
```

Filtro DataOutputStream (cont)

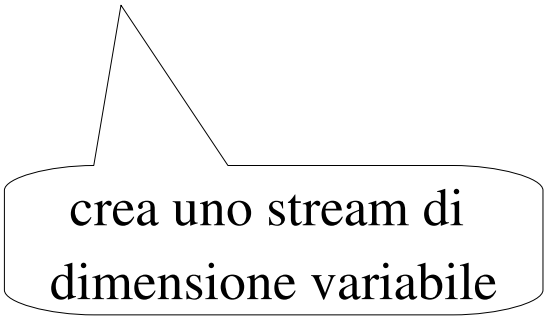
```
        dos.writeInt(i);
        dos.writeLong(l);
        dos.writeBoolean(bo);
        dos.writeFloat(f);
        dos.writeDouble(d);

        dos.close();
    }
    catch(FileNotFoundException e){
        System.out.println("errore di crazione file");
    }
    catch(IOException e) {
        System.out.println("errore di IO sulla write");
    }
```

Filtro GZIPOutputStream

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

.....

try {
    DataOutputStream dos = new DataOutputStream(
    new GZIPOutputStream(new FileOutputStream("prv.Gz")));

    dos.write(buf, 0, buf.length);
    dos.writeChar(i);
    dos.writeInt(i);
    dos.writeBoolean(bo);
    dos.writeDouble(d);

    ......
}
catch.....
```

ByteArrayOutputStream

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

byte[] buf = (new String("ciao a tutti")).getBytes();
int i = 33;
double d = 33.3d;

DataOutputStream dos = new DataOutputStream(
                                    new ByteArrayOutputStream());

dos.write(buf, 0, buf.length);

dos.writeInt(i);

dos.writeDouble(d);
```



ByteArrayOutputStream (cont)

```
byte[] bufSped = bos.toByteArray();
```

Copia il contenuto dello stream nel vettore di byte

```
System.out.println("lo stream è lungo " + bos.size());

DatagramPacket dp = new DatagramPacket(bufSped,
                                       bufSped.length, host, portarem);
```

Attenzione alla lunghezza del vettore!!!

```
bos.reset();
```

svuota lo stream

```
bos.close();
```

non necessaria

Filtro BufferedOutputStream

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

try {
    BufferdOutputStream bos = new BufferedOutputStream(
                              new FileOutputStream("prv.txt")));

    bos.write(buf, 0, buf.length);

    bos.flush();

    bos.close();
}
catch.....
```

di default viene allocato un buffer di 512 bytes

scrive nel buffer, quando il buffer è pieno i dati vengono copiati nel file

copia i dati dal buffer nel file

copia i dati dal buffer nel file e lo chiude

BufferedWriter

```
String s = new String("ciao");

try {

    BufferedWriter bw = new BufferedWriter(
                                new FileWriter("provaw.txt"));

    bw.write(s, 0, s.length());

    bw.close();
}
catch(IOException e) {
    System.out.println("errore di IO");
}
```

BufferedWriter (2)

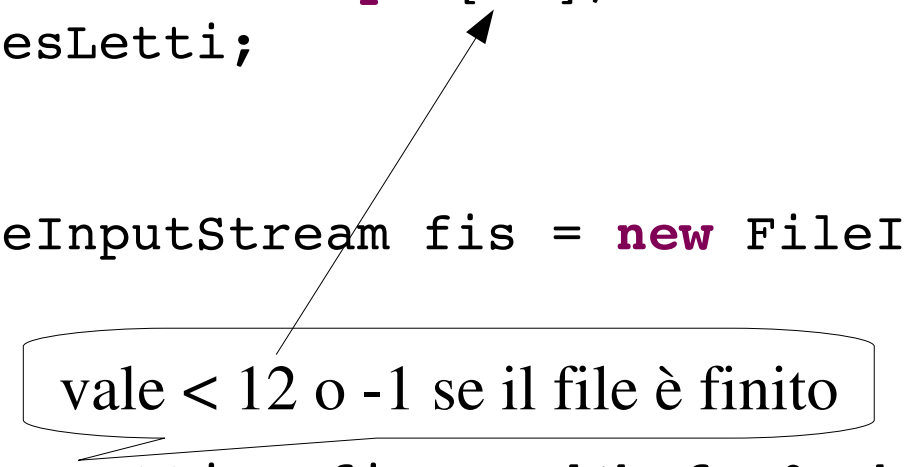
```
char[] c = new char[15];
.... inizilizzazione di c ....

try {

    BufferedWriter bw = new BufferedWriter(
                                new FileWriter("provaw.txt"));

    bw.write(c, 0, c.length());

    bw.close();
}
catch(IOException e) {
    System.out.println("errore di IO");
}
```

Concatenazione di Filtri

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

try {

    DataOutputStream dos = new DataOutputStream(
                           new BufferedOutputStream(
                           new GZIPOutputStream(
                           new FileOutputStream("prv.Gz"))));

    dos.writeInt(i);
    dos.write(buf, 0, buf.length);

    dos.flush();

    dos.close();
}
catch ....
```

FileInputStream

```
byte[] buf = new byte[12];
int bytesLetti;

try {
    FileInputStream fis = new FileInputStream("prv.txt");

    bytesLetti = fis.read(buf, 0, buf.length);

    fis.close();
}
catch(FileNotFoundException e) {
    System.out.println("errore file non trovato");}
catch(IOException e) {
    System.out.println("errore di IO sulla read");}
```



Filtro DataInputStream

```
byte[] buf = new byte[12];
int i;
long l;
int b;
char c;
boolean bo;
float f;
double d;


try {
    DataInputStream dis = new DataInputStream(
                          new FileInputStream("prv"));


    dis.read(buf, 0, buf.length);
    b = dis.readByte();
    c = dis.readChar();
```

Filtro DataInputStream (cont)

```
        i = dis.readInt();
        l = dis.readLong();
        bo = dis.readBoolean();
        f = dis.readFloat();
        d = dis.readDouble();

        dis.close();
    }
    catch(FileNotFoundException e) {
        System.out.println("errore file non trovato");}
    catch(IOException e) {
        System.out.println("errore di IO sulla read");}
```

Filtro GZIPInputStream

```
int i;
byte[] buf = new byte[12];

try {
    DataInputStream dis = new DataInputStream(
      new GZIPInputStream(new FileInputStream("prv.Gz")));

    i = dis.readInt();

    dis.read(buf, 0, buf.length);

    dis.close();
}
catch....
```

ByteArrayInputStream

```
try
{
    so.receive(dp);
}
catch ....

try {
    DataInputStream dis = new DataInputStream(new
          ByteArrayInputStream(dp.getData(),0,dp.getLength()));

    int i = dis.readInt();

    double d = dis.readDouble();

    byte[] buf = new byte[12];
    dis.read(buf, 0, buf.length);

}
catch(IOException e) {
    System.out.println("errore sulla datainputstream");}
```

Filtro BufferedInputStream

```
byte[] buf = new byte[14];
int byteLetti;

try {
    BufferedInputStream bis = new BufferedInputStream(
                                  new FileInputStream("prv.txt"));

    byteLetti = bis.read(buf, 0, buf.length);
}
catch....
```

di default viene allocato un buffer di 2048 bytes

vengono letti dal file 2048 bytes, 14 vengono messi in buf ed il resto in un buffer per le prossime read

```
int bytePresenti = bis.available();


long bytesDaSaltare = 33l;

bis.skip(bytesDaSaltare);
```

BufferedReader

```
try {
    BufferedReader br = new BufferedReader(
                                    new FileReader("prv.txt"));
    if (br.ready()) {

        String s = br.readLine();

        System.out.println(s);
    }

    br.close();
}
catch(IOException e) {
    System.out.println("errore sulla readLine");
}
```

BufferedReader (2)

```
char[] c = new char[15];
int letti;

try {
    BufferedReader br = new BufferedReader(
                                    new FileReader("prv.txt"));
    if (br.ready()) {

        letti = br.read(c, 0, c.length);

        System.out.println(new String(c, 0, letti));
    }

    br.close();
}
catch(IOException e) {
    System.out.println("errore sulla read");
}
```