15
Programmazione di rete

15.1 Link all'interno di applet

URL – Uniform Resurce Locator

Classe URL
costruttori:
URL(String protocollo, String hostname, int porta, String filename);
URL(String protocollo, String hostname, String filename);
URL(URL url, String pathname);
URL(String stringaUrl); //controllare l'eccezione MalformedUrlException

metodi:
getDocumentBase(): restituisce l'URL del file HTML corrente
getCodeBase(): restituisce l'URL del file .class dell'applet

getAppletContext().showDocument(theURL);

Esempio

```
import java.net.URL;      import java.net.MalformedURLException;

class Bookmark {
   String name;           URL url;

   Bookmark(String name, String theURL) {
      this.name = name;
      try { this.url = new URL(theURL); }
      catch ( MalformedURLException e) {
         System.out.println("Bad URL: " + theURL);
      }        }       }
```

```
import java.awt.*;                    import java.awt.event.*;
```

```
import java.net.MalformedURLException;   import java.net.URL;

public class ButtonLink11 extends java.applet.Applet  implements ActionListener {

  Bookmark bmlist[] = new Bookmark[3];

  public void init() {
     bmlist[0] = new Bookmark("Laura's Home Page", "http://www.lne.com/lemay/");
     bmlist[1] = new Bookmark("Gamelan", "http://www.gamelan.com");
     bmlist[2]= new Bookmark("Java Home Page", "http://java.sun.com");

     setLayout(new GridLayout(bmlist.length,1, 10, 10));
     for (int i = 0; i < bmlist.length; i++) {
        Button b = new Button(bmlist[i].name);
        b.addActionListener(this);add(b);
     }
  }

  public void actionPerformed(ActionEvent e) {
     if (e.getSource() instanceof Button)
        LinkTo(((Button)e.getSource()).getLabel());
  }

  void LinkTo(String name) {
     URL theURL = null;
     for (int i = 0; i < bmlist.length; i++)
       {if (name.equals(bmlist[i].name))  theURL = bmlist[i].url;}
     if (theURL != null) getAppletContext().showDocument(theURL);
  }
}
```

15.2 Indirizzi Internet

Classe InetAddress

metodi:

```
public String getHostName()
public byte[] getBytes()
public String getHostAddress()
public static InetAddress getLocalHost() throws UnknownHostException

public static InetAddress getByName(String host) throws UnknownHostException
public static InetAddress[] getAllByName(String host)
        throws UnknownHostException
```

Esempio

```
/**@version 1.00 1999-08-28   @author Cay Horstmann */

import java.net.*;

public class InetAddressTest
{  public static void main(String[] args)
   {
try {
        if (args.length > 0)
        {  String host = args[0];
           InetAddress[] addresses  = InetAddress.getAllByName(host);
           for (int i = 0; i < addresses.length; i++)  System.out.println(addresses[i]);
        }
        else
        {  InetAddress localHostAddress = InetAddress.getLocalHost();
           System.out.println(localHostAddress);
        }
     }
     catch (Exception e)
     {  System.out.println("Error: " + e);}
   }
}
```

15.3 Connessioni tramite socket

socket – elemento astratto del software di rete che consente al programma
di ricevere trasmettere dati

Costruttori:

Socket(InetAddress host, int porta)
Socket(String host, int porta)

Metodi:
public InputStream getInputStream() throws IOException
public OutputStream getOutputStream() throws IOException
public void close() throws IOException

public InetAddress getInetAddress()
public int getPort()
public int getLocalPort()
public synchronized void setSoTimeout(int timeout) throws SocketException
public synchronized int getSoTimeout() throws SocketException

Esempio

```
/* @version 1.10 1997-06-27 @author Cay Horstmann*/
import java.io.*;    import java.net.*;
public class SocketTest
{
  public static void main(String[] args)
   {  try
     {  Socket s = new Socket("time-A.timefreq.bldrdoc.gov", 13);

        BufferedReader in = new BufferedReader
          (new InputStreamReader(s.getInputStream()));
        boolean continua = true;
        while (continua)
        {  String line = in.readLine();
          if (line == null) continua = false;
          else  System.out.println(line);
        }
     }
     catch (IOException e)
     {  System.out.println("Error" + e);}
   }
```

```
}
```

controllo del timeout tramite thread

Esempio

```
/*version 1.00 1999-08-27 author Cay Horstmann */

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

public class SocketOpenerTest
{  public static void main(String[] args)
   {  String host;
      if (args.length > 0) host = args[0];
      else host = "www.yourcompany.com";

      int port;
      if (args.length > 1) port = Integer.parseInt(args[1]);
      else port = 80;

      int timeout = 5000;
      Socket s = SocketOpener.openSocket(host, port, timeout);

      if (s == null) System.out.println("The socket could not be opened.");
      else System.out.println(s);
   }
}

class SocketOpener implements Runnable //utilizzabile in ogni applicazione
{
   private String host;
   private int port;
   private Socket socket;

   public SocketOpener(String aHost, int aPort)
   {  socket = null;
      host = aHost;
      port = aPort;
   }
```

```
public static Socket openSocket(String aHost, int aPort, int timeout)
{
  SocketOpener opener = new SocketOpener(aHost, aPort);
     Thread t = new Thread(opener);
     t.start();
     try {  t.join(timeout); }
     catch (InterruptedException exception)   {}
     return opener.getSocket();
   }

   public void run()
   {  try {socket = new Socket(host, port);}
     catch (IOException exception) {}
   }

   public Socket getSocket()
   {return socket;}
};
```

15.4 Implementazione server

Classe ServerSocket – elemento astratto del software di rete che consente al programma di ricever e trasmettere dati

Costruttori:
public ServerSocket(int port) throws IOException

Metodi:

public Socket accept() throws IOException
public void close() throws IOException

public InetAddress getInetAddress()
public int getLocalPort() //porta su cui il socket è in ascolto
public synchronized void setSoTimeout(int timeout) throws SocketException
public synchronized int getSoTimeout() throws IOException

```
ServerSocket s= new ServerSocket(8189);
Socket incoming = s.accept();
BufferedReader in = new BufferedReader
        (new InputStreamReader(incoming.getInputStream()));
      PrintWriter out = new PrintWriter
        (incoming.getOutputStream(), true /* autoFlush */);
```

Esempio
/*@version 1.10 1997-06-27 @author Cay Horstmann */

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

public class EchoServer
{  public static void main(String[] args )
   {  try
      {  ServerSocket s = new ServerSocket(8189);

        Socket incoming = s.accept( );

         BufferedReader in = new BufferedReader
           (new InputStreamReader(incoming.getInputStream()));
         PrintWriter out = new PrintWriter
           (incoming.getOutputStream(), true /* autoFlush */);

         out.println( "Hello! Enter BYE to exit." );

         boolean done = false;
         while (!done)
         {  String line = in.readLine();
           if (line == null) done = true;
           else
           {  out.println("Echo: " + line);
             if (line.trim().equals("BYE")) done = true;
           }
         }
         incoming.close();
      }
      catch (Exception e)
      {  System.out.println(e);}
   }
}
//Il client può essere telnet indirizzo 127.0.0.1 porta 8189
```

Oppure:

```
/*@version 1.10 1997-06-27 @author Cay Horstmann */
```

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

public class EchoClient
{
private static final int PORTNUM = 8189;

public static void main(String[] args) {
   Socket socket=null;
   BufferedReader in=null;                PrintWriter out=null;
   String address;

   // Check the command-line args for the host address
   if (args.length != 1) {System.out.println("Usage: java EchoClient <address>");
                          return;}
   else
   address = args[0];

   // Initialize the socket and streams
   try {
     socket = new Socket(address, PORTNUM);
     in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
     out = new PrintWriter(socket.getOutputStream());
   }
   catch (IOException e)
{System.err.println("Exception: couldn't create stream socket");
         System.exit(1);}

// Process user input and server responses
   try { StringBuffer  str = new StringBuffer(128);
     String  inStr;
     int  c;

     while ((inStr = in.readLine()) != null) { //usa caratteri UNICODE
       System.out.println(inStr);
       while ((c = System.in.read()) != '\n') str.append((char)c);
       System.out.println("Client: " + str);
       out.print(str.toString());      //usa caratteri UNICODE
       out.flush();
       if ((str.toString()).trim().equals("BYE")) break;
       str.setLength(0);}
     // Cleanup
     out.close();      in.close();      socket.close();
   }
```

```
    catch (IOException e)
{ System.err.println("Exception: I/O error trying to talk to server");}
  }
}
```

servire più client:

```
ServerSocket s = new ServerSocket(8189);

while(true)
{  Socket incoming = s.accept( );
   Thread t = new ThreadedEchoHandler(incoming);
   t.start();
}
```

Esempio

```
/* @version 1.10 1997-06-27 * @author Cay Horstmann*/
import java.io.*;    import java.net.*;

public class ThreadedEchoServer
{  public static void main(String[] args )
   {  int i = 1;
      try
      {  ServerSocket s = new ServerSocket(8189);
         for (;;)
         {  Socket incoming = s.accept( );
            System.out.println("Spawning " + i);
            new ThreadedEchoHandler(incoming, i).start();
            i++;}
      }
      catch (Exception e)      {  System.out.println(e);}
   }
}
```

```
class ThreadedEchoHandler extends Thread
{ private Socket incoming;          private int counter;

 public ThreadedEchoHandler(Socket i, int c)  { incoming = i; counter = c; }

  public void run()
  { try
    {  BufferedReader in = new BufferedReader
         (new InputStreamReader(incoming.getInputStream()));
       PrintWriter out = new PrintWriter
         (incoming.getOutputStream(), true /* autoFlush */);

        out.println( "Hello! Enter BYE to exit." );

       boolean done = false;
       while (!done)
       {  String str = in.readLine();
         if (str == null) done = true;
         else  {  out.println("Echo (" + counter + "): " + str);
           if (str.trim().equals("BYE"))   done = true;
         }
       }
       incoming.close();}
    catch (Exception e)  {  System.out.println(e); }
  }}
```

15.5 Applicazioni client-server

CLIENT:

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

public class Trivia {
private static final int PORTNUM = 1234;

public static void main(String[] args) {
   Socket socket=null;
   DataInputStream in=null;              DataOutputStream out=null;
   String address;

   // Check the command-line args for the host address
   if (args.length != 1) {System.out.println("Usage: java Trivia <address>");
                          return;}
   else
   address = args[0];

   // Initialize the socket and streams
   try {
     socket = new Socket(address, PORTNUM);
     in = new DataInputStream(new BufferedInputStream(socket.getInputStream()));
     out =new DataOutputStream(new BufferedOutputStream(socket.getOutputStream()));
   }
   catch (IOException e)
        {System.err.println("Exception: couldn't create stream socket");
        System.exit(1);}
```

```
// Process user input and server responses
   try { StringBuffer  str = new StringBuffer(128);
     String  inStr;
     int  c;

     while ((inStr = in.readUTF()) != null) { //usa caratteri UNICODE
       System.out.println("Server: " + inStr);
       if (inStr.equals("Bye.")) break;
       while ((c = System.in.read()) != '\n') str.append((char)c);
       System.out.println("Client: " + str);
       out.writeUTF(str.toString());      //usa caratteri UNICODE
       out.flush();
       str.setLength(0);}
     // Cleanup
     out.close();      in.close();      socket.close();
   }
   catch (IOException e)
        { System.err.println("Exception: I/O error trying to talk to server");}
  }
}
```

SERVER:

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

public class TriviaServer extends Thread {
  private static final int  PORTNUM = 1234;
  private ServerSocket   serverSocket;

  private static final int  WAITFORCLIENT = 0;
  private static final int  WAITFORANSWER = 1;
  private static final int  WAITFORCONFIRM = 2;
  private int state = WAITFORCLIENT;

  private int  numQuestions;       private int  num = 0;
  private String[] questions;       private String[]  answers;
  private Random  rand = new Random(System.currentTimeMillis());


  public TriviaServer() {
    super("TriviaServer");
    try {
      serverSocket = new ServerSocket(PORTNUM);
      System.out.println("TriviaServer up and running...");
    }
    catch (IOException e)
          {System.err.println("Exception: couldn't create socket");
          System.exit(1);}
  }

  public static void main(String[] args) {
    TriviaServer server = new TriviaServer();
    server.start();
  }
```

```
public void run() {
   Socket  clientSocket= null;

   // Initialize the arrays of questions and answers
   if (!initQnA())
        {System.err.println("Error: couldn't initialize questions and answers");
        return;}

   // Look for clients and ask trivia questions
   while (true) {       // Wait for a client
    if (serverSocket == null) return;
    try { clientSocket = serverSocket.accept();}
    catch (IOException e)
        {System.err.println("Exception: couldn't connect to client socket");
        System.exit(1);}

    // Perform the question/answer processing
    try {
      DataInputStream is =
new DataInputStream(new BufferedInputStream(clientSocket.getInputStream()));
      DataOutputStream os =
new DataOutputStream(new BufferedOutputStream(clientSocket.getOutputStream()));

        String inLine, outLine;

       // Output server request
      outLine = processInput(null);
      os.writeUTF(outLine);          //usa caratteri UNICODE
      os.flush();

      // Process and output user input
      while ((inLine = is.readUTF()) != null) { //usa caratteri UNICODE
        outLine = processInput(inLine);
      os.writeUTF(outLine);          //usa caratteri UNICODE
      os.flush();
        if (outLine.equals("Bye."))           break;
      }

      // Cleanup
      os.close();       is.close();    clientSocket.close();
    }
    catch (Exception e) { System.err.println("Exception: " + e);
      e.printStackTrace();      }
```

```
}   }
```

```
private boolean initQnA() {
  boolean isQ = true;
  int index = 0;
  BufferedReader inStream =null;


  try {
    inStream  = new BufferedReader (new FileReader("QnA.txt"));
    String riga;
    riga= inStream.readLine();
    numQuestions =  Integer.parseInt(riga);
    numQuestions /= 2;
    questions = new String[numQuestions];
    answers = new String[numQuestions];
    while ((riga = inStream.readLine()) != null)
      {
      if (isQ)
        {questions[index] = riga;
        isQ = false; }
      else
        {answers[index] = riga;
        isQ = true;
        index ++;     //aggiorno dopo che ho inserito il secondo elemento della coppia
          }
      }
      inStream.close();
  }
  catch (FileNotFoundException e)
       {System.err.println("Exception: couldn't find the fortune file");
        return false;}
  catch (IOException e)
       {System.err.println("Exception: I/O error trying to read questions");
       return false;}
  return true;
 }
```

```
String processInput(String inStr) {
   String outStr="";
   switch (state) {
   case WAITFORCLIENT:
     // Ask a questio
     outStr = questions[num];
     state = WAITFORANSWER;
     break;
   case WAITFORANSWER:
     // Check the answer
     if (inStr.trim().equalsIgnoreCase(answers[num])) //trim elimina spazi pre e postfissi
       outStr = "That's correct! Want another? (y/n)";
     else
       outStr = "Wrong, the correct answer is "+answers[num]+". Want another? (y/n)";
     state = WAITFORCONFIRM;
     break;
   case WAITFORCONFIRM:
     // See if they want another question
     if (inStr.trim().equalsIgnoreCase("y")) {  //trim elimina spazi pre e postfissi
       num = Math.abs(rand.nextInt()) % questions.length;
       outStr = questions[num];
       state = WAITFORANSWER;
      }
     else
         { outStr = "Bye.";
          state = WAITFORCLIENT;}
    }
   return outStr;
  }      }
```

ESEMPIO FILE

```
4
What caused the craters on the moon?
meteorites
How far away is the moon (in miles)?
239000
```

Esempio chat

```
/*   ChatServer
Progetto del Corso di Programmazione e Laboratorio di Programmazione
Anno Accademico 1999/2000   Autori: Gentile Francesco, Lucarelli Marco
Parametri:
  porta= porta cui mettersi in ascolto (default=3000)
  utenti= numero massimo di utenti ammessi (default=0, nessun limite)
  pausa= numero minimo di millisecondi tra due messaggi accettati dallo stesso client
 */
import java.net.*;  import java.io.*; import java.util.*;

public class ChatServer {

protected int utentiConnessi= 0;

public ChatServer(int porta, int utenti, int pausa) throws IOException, InterruptedException
{
     ServerSocket server= new ServerSocket(porta);
     while (true) {
      if ((utentiConnessi < utenti) || (utenti == 0)) {
        Socket client= server.accept();
        System.out.println(client.getInetAddress());
        ChatThread c= new ChatThread(client, this, pausa);
        c.start();
        synchronized(this) {utentiConnessi++;}
       } else Thread.sleep(1000);
     }
   }

//  public void decrementaUtenti() {synchronized(this) {utentiConnessi--;}}
  synchronized public void decrementaUtenti(){utentiConnessi--;}

  public static void main(String args[]) throws IOException, InterruptedException {
    if (args.length < 1) new ChatServer(3000, 0, 0);
    else {
      if (args.length < 2) new ChatServer(Integer.parseInt(args[0]), 0, 0);
      else {
if (args.length < 3) new ChatServer(Integer.parseInt(args[0]), Integer.parseInt(args[1]),
0);
 else new ChatServer(Integer.parseInt(args[0]),
                                Integer.parseInt(args[1]), Integer.parseInt(args[2]));
      }
```

```
    }
  }}
/*  ChatThread  */
import java.net.*;  import java.io.*; import java.util.*;

public class ChatThread extends Thread {
  protected static int lmess= 256; // Massima lunghezza di un messaggio
  protected static int lnome= 16; // Massima lunghezza per il nick di un utente
  protected static char separatore= '\n';
  protected static int pausa;
  protected static ChatServer cs;


  protected static String elenconomi= "" + separatore;  // "\n"
  protected static Vector threads= new Vector();

/* i campi statici sono campi di classe e quindi comuni a tutte le istanze che vengono
create*/

  protected Socket s;
  protected DataInputStream i;
  protected DataOutputStream o;
  protected String nome= "";



  public ChatThread(Socket s, ChatServer cs, int pausa) throws IOException {
    this.s= s;              this.cs= cs;         this.pausa= pausa;
    i= new DataInputStream(new BufferedInputStream(s.getInputStream()));
    o= new DataOutputStream(new BufferedOutputStream(s.getOutputStream()));
  }

  protected void ferma() {
    cs.decrementaUtenti();
    synchronized(threads) {
      threads.removeElement(this);
      elenconomi= "" + separatore;
      Enumeration e= threads.elements();
      while (e.hasMoreElements()) {
        ChatThread oct= (ChatThread) e.nextElement();
        elenconomi= elenconomi + oct.nome + separatore;
      }
    }
```

```
    trasmetti(nome + " è uscito dalla chat.", "Tutti", true);
    try {s.close();}
      catch (IOException ex) {}
  }









public void run() {
   try {   //legge il nuovo nome
   while (nome == "") {
      nome= i.readUTF();
      if ((nome.length() > lnome) || (nome.indexOf(separatore) != -1))
                  { ferma();  return;}
      }
// evita nomi duplicati: se uno si aggiunge con lo stesso nome di una persona
 // già presente aggiunge al nome tanti asterischi fino a che il nome
 // diventa diverso da tutti gli altri
   synchronized(threads) {         //aggiunge il nuovo thread
   while (elenconomi.indexOf(separatore + nome + separatore) != -1) {nome+= "*";}
      elenconomi+= nome + separatore;
      threads.addElement(this);
      }
   trasmetti(nome + " si è unito alla chat.", "Tutti", true);


   while (true) {  //si mette in ascolto sul canale di input
      String msg= i.readUTF();
//il client ha mandato una stringa della forma:
//Destinatario + Separatore + messaggio
      int pos= msg.indexOf(separatore) + 1;
if ((msg.length() <= lmess) && (pos != 0) && (msg.indexOf(separatore, pos) == -1))
trasmetti(nome + "> " + msg.substring(pos, msg.length()), msg.substring(0, pos-1), false);
      try { sleep(pausa);} catch (InterruptedException ex) {}

  }
   } catch (IOException ecc) {
   ferma();
   return;
}}
```

```
protected void trasmetti(String messaggio, String destinatario, boolean trasmettiElenco) {
  synchronized(threads) {
    Enumeration e= threads.elements();
    while (e.hasMoreElements()) {
    ChatThread oct= (ChatThread) e.nextElement();
    if (((oct.nome.equals(destinatario)) || (destinatario.equals("Tutti"))) &&
                        ((oct != this) || trasmettiElenco)){
    try {
      if (trasmettiElenco)
              oct.o.writeUTF(messaggio + elenconomi);
      else
              oct.o.writeUTF(messaggio);
      oct.o.flush();
      } catch (IOException ex) {
    oct.ferma();
    oct.interrupt();
    }}}}}}
```

```
/* ChatClient */
import java.net.*;  import java.io.*; import java.awt.*;
 import java.awt.event.*;

public class ChatClientFrame extends Frame
        implements Runnable, ActionListener, WindowListener{

  Socket s=null;
  protected DataInputStream i=null;
  protected DataOutputStream o=null;

  protected String nome= "";

  protected Label l;
  protected TextArea output;
  protected TextField input;
  protected List scelta;

 protected Thread ascolta= null;
  protected char separatore= '\n';
  static String host = null;
  static String porta = null;



public void windowOpened(WindowEvent e) {}
public void windowActivated(WindowEvent e) {}
public void windowDeactivated(WindowEvent e) {}
public void windowIconified(WindowEvent e) {}
public void windowDeiconified(WindowEvent e) {}
public void windowClosed(WindowEvent e) {}


public void windowClosing(WindowEvent e) {
       termina();
         setVisible(false);
         dispose();
         System.exit(0);

}
```

```
public  ChatClientFrame() {

   Panel pannello=new Panel();
   pannello.setLayout(new BorderLayout());

   setFont(new Font("Helvetica", Font.PLAIN, 14));
   setLayout(new BorderLayout());
   output= new TextArea("", 10, 10, TextArea.SCROLLBARS_VERTICAL_ONLY);
   add(output,"Center");
   output.setEditable(false);

   scelta= new List(20, false);
   scelta.add("Tutti");
   scelta.select(0);
   pannello.add(scelta,"Center");
   add(pannello,"East");

   l= new Label("Destinatario:");
   pannello.add(l,"North");

   input= new TextField();
   input.setEditable(false);
   input.addActionListener(this);
   add(input,"South");

   addWindowListener(this);
   ascolta= new Thread(this);
   ascolta.start();
  }

public static void main (String args []){
     if (args.length >= 1)
         host= args[0];
     else
       host= "localhost";
     if (args.length >= 2)
         porta= args[1];
     else
        porta= "3000";
         ChatClientFrame finestra = new ChatClientFrame();
         finestra.setSize(800,500);
         finestra.setVisible(true);
}
```

```
 public void run() {  //qui leggiamo i messaggi che manda il server
   try {
     output.append(" G/L Chat  di Gentile Francesco e Lucarelli Marco\n\n");
     output.append("Puoi inviare messaggi alla chat scrivendo nella riga in basso e poi battendo INVIO. Puoi scegliere dalla lista a destra il destinatario dei messaggi che invii.\n\n");
     output.append("Connessione al server in corso... ");
     s= new Socket(host, Integer.parseInt(porta));
     i= new DataInputStream(new BufferedInputStream(s.getInputStream()));
     o= new DataOutputStream(new BufferedOutputStream(s.getOutputStream()));
     output.append("ok\n\nScrivi il tuo nome:\n");
     input.setEditable(true);
     input.requestFocus();
     String selezionato= "";
     String riga= "";
     String str;
     String parole [];
     int  n;

     while (true) {
      riga= i.readUTF();
      if (riga.indexOf(separatore)== -1)
                {output.append(riga + '\n');}   //   “marco > Ciao!”
       else
         {                    //”marco si è collegato alla chat\nstefano\nanna\nmarco”
         parole = riga.split("\\n");
        output.append(parole[0] + '\n');
        selezionato= scelta.getSelectedItem();
        scelta.removeAll();
        scelta.add("Tutti");
        scelta.select(0);
          for (n = 1; n < parole.length; n++){
           str= parole[n];
           scelta.add(str);
           if (str.equals(selezionato))
                scelta.select(n);
           }}}
     }
    catch (Exception ecc) {
     output.append("ERRORE!!!");
```

```
      output.append(ecc.getMessage());
      termina();
      return;
    } }

public void actionPerformed(ActionEvent e) {  //qui inviamo messaggi al server
  if ((e.getSource() instanceof TextField) && ((TextField) e.getSource() == input) &&
          ((input.getText().trim() != "")))
{
      String msg;
      String testo = input.getText();
      if (nome == "") {
        nome= testo;
        output.setText("");
        msg= nome;
      }
else {
        output.append("> " + testo + '\n');
        msg= scelta.getSelectedItem() + separatore + testo;
      }
      try {
        o.writeUTF(msg);
        o.flush();
      } catch (IOException ecc) {}
      input.setText("");
    }
  }

public void termina() {
    if (ascolta != null)  ascolta.interrupt();
    ascolta= null;
    try { if (o != null) o.close();}
    catch (IOException ecc) {ecc.printStackTrace();}
    try { if (i != null) i.close();}
    catch (IOException ecc) {ecc.printStackTrace();}
    try { if (s != null) s.close();}
    catch (IOException ecc) {ecc.printStackTrace();}
  }


}
```

15.6 Connessioni URL

openStream(): apre il flusso di dati associato ad un URL

Esempio

```
import java.io.*;     import java.net.*;  //legge il codice HTML della pagina

public class UrlTest
{  public static void main(String[] args)
   {  try {
         URL url = new URL("http://www.unipg.it");
         InputStream in = url.openStream();

         BufferedReader d = new BufferedReader(new InputStreamReader(in));

         //DataInput d = new DataInputStream(new BufferedInputStream(in));
         String line;
         while ((line = d.readLine()) != null)
         {System.out.println(line);}
         }
      catch (MalformedURLException e)
        {System.out.println("Bad URL");}
     catch (IOException e)
        {System.out.println("Error" + e);}
   }
}
```

Classe URLConnection

1) Creazione oggetto:
URLConnection connessione = url.openConnection();

2) impostazione proprietà:

```
public void setDoInput(boolean)                public boolean getDoInput()
public void setDoOutput(boolean)               public boolean getDoOutput()
public void setAllowUserInteraction(boolean)
public boolean getAllowUserInteraction()
//visualizzazione finestra dialogo per password
public static void setDefaultAllowUserInteraction(boolean)
public static boolean getDefaultAllowUserInteraction()
public void setUseCaches(boolean)              public boolean getUseCaches()
```

// cerca prima nella cache del browser: solo per applet
public void setRequestProperty(String key, String value) //imposta particolari valori
public String getRequestProperty(String key)
public void setIfModifiedSince(long) public long getIfModifiedSince()
//considera solo i dati modificati dopo una certa dati
3) connessione:
 connessione.connect();

4) richiesta informazioni di intestazione del server:
 public String getContentEncoding()
 public long getExpiration()
 public long getDate()
 public long getLastModified()
 public String getHeaderField(String name)
 public String getHeaderFieldKey(int n)
//restituisce la chiave dell'n-esimo capo di intestazione
 public String getHeaderField(int n) //restituisce l'n-esimo capo di intestazione

5) apertura flusso di input
connessione.setDoInput(true); //questo comunque è ivalore di default
public InputStream getInputStream() throws IOException

 5 bis) apertura flusso di output
connessione.setDoOutput(true);
public OutputStream getOutputStream() throws IOException

Connessione a pagina protetta da password:

String input = nomeUtente +":"+password;
String encoding = base64Encode(input);
Connection.setRequestProperty("Authorization","Basic"+encoding);

per accedere a un file via ftp di costruisce un URL del tipo:

ftp://nomeUtente:password@ftp.server.com/pub/file.txt

Esempio

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

public class GetRaven2 extends java.applet.Applet implements Runnable {
  URL theURL;
  Thread runner;
  TextArea ta = new TextArea("Getting text...");
  String file = "raven.txt";

  public void init() {
   setLayout(new GridLayout(1,1));
   theURL = getDocumentBase();
   String nome = theURL.toString();
   System.out.println(nome);
   int pos = nome.lastIndexOf('/');
   try { this.theURL = new URL(nome.substring(0,pos+1).concat(file)); }
    catch ( MalformedURLException e)
      {System.out.println("Bad URL: " + theURL);}
   add(ta);
  }

  public Insets getInsets() {return new Insets(10,10,10,10);}

 public void start() {if (runner == null) {runner = new Thread(this); runner.start();}}
 public void stop() { if (runner != null)    {runner.stop();  runner = null;} }

  public void run() {
    URLConnection conn = null;
    BufferedReader data = null;
    String line;
    StringBuffer buf = new StringBuffer();

    try {
      conn = this.theURL.openConnection();
     conn.setDoInput(true); //non indispensabile perché vero di default
     conn.connect();
      ta.setText("Connection opened...");

      data = new BufferedReader(new InputStreamReader(conn.getInputStream()));
      ta.setText("Reading data...");
      while ((line = data.readLine()) != null)
```

```
                {buf.append(line + "\n");}
     ta.setText(buf.toString());
    }
   catch (IOException e) {  System.out.println("IO Error:" + e.getMessage());}
  }
}
```

Esempio

```
/** version 1.00 1999-08-27 author Cay Horstmann*/
import java.io.*;    import java.net.*;  import java.util.*;

public class URLConnectionTest
{  public static void main(String[] args)
   {  try
         {String urlName;
        if (args.length > 0)  urlName = args[0];
        else urlName = "http://java.sun.com";

        URL url = new URL(urlName);
        URLConnection connection = url.openConnection();

        // set username, password if specified on command line
        if (args.length > 2)
        {  String username = args[1];
           String password = args[2];
           String input = username + ":" + password;
            String encoding = new sun.misc.BASE64Encoder().encode(input.getBytes());
           connection.setRequestProperty("Authorization", "Basic " + encoding);
        }

        connection.connect();

        // print header fields
        int n = 1;
        String key;
        while ((key = connection.getHeaderFieldKey(n)) != null)
        {  String value = connection.getHeaderField(n);
           System.out.println(key + ": " + value);
           n++;
        }

        // print convenience functions
        System.out.println("----------");
        System.out.println("getContentType: " + connection.getContentType());
        System.out.println("getContentLength: "  + connection.getContentLength());
        System.out.println("getContentEncoding: " + connection.getContentEncoding());
        System.out.println("getDate: " + connection.getDate());
        System.out.println("getExpiration: " + connection.getExpiration());
        System.out.println("getLastModifed: " + connection.getLastModified());
```

```
        System.out.println("----------");
        BufferedReader in = new BufferedReader(new
          InputStreamReader(connection.getInputStream()));

        // print first ten lines of contents

        String line;
        n = 1;
        while ((line = in.readLine()) != null && n <= 10)
        {  System.out.println(line);
          n++;
        }
        if (line != null) System.out.println(". . .");
      }
      catch (IOException exception)
      {  System.out.println("Error: " + exception)  }
    }
}
```

15.7 Invio dati a pagina Web

<form name=mapForm2
action="http://us.rd.yahoo.com/maps/home/submit_a/*-http://maps.yahoo.com/maps"
method=get>
<input maxlength=40 size=25 name="addr" value="">
<input maxlength=40 size=25 name="csz" value="">

metodo GET:

Esempio
Indirizzo:
http://maps.yahoo.com/py/maps.py
campi (specificati col TAG <INPUT>):
addr
csz
spazi → +
caratteri non alfanumerici → %codice ASCII di due cifre esadecimali
codifica URL:
http://maps.yahoo.com/py/maps.py?addr=1+Infinite+Loop&csz=Cupertino+Ca

==========

metodo POST:

lavora sul flusso URLConnection
parametri :
nome&valore

i valori vanno codificati secondo il modello URL:

Esempio

http://www.census.gov/ipc/www/idbprint.html

```
<form method=post action="/cgi-bin/ipc/idbsprd">

<br><select name="tbl" size=8 >
<option value="001">001 Total Midyear Population
<option value="002">002 Urban Population as a Percent of Total Population
. . .
</select>

<p><B>Select one or more countries:</B>
<br><select multiple name="cty" size=8>
<option value="AF">Afghanistan
<option value="AL">Albania
 . . .
</SELECT>


<p><B>Year Selection</B>
<br><input name="optyr" type=radio value=latest checked>Latest available year.
<br><input name="optyr" type=radio value=all >All available years.
<br><input name="optyr" type=radio value=selected >Selected years:

(select individual years and/or a year range below)
<p>Select one or more years (see below for selection by interval):
<p><select multiple name="yr" size=8>
<option>1950
<option>1951
. . .
</select>
```

File proprietà:

```
URL=http://www.census.gov/cgi-bin/ipc/idbsprd
tbl=001
cty=CH
optyr=latest checked


tbl=001&cty=CH&optyr=latest checked
```

Esempio

```
/* @version 1.00 1999-08-28 @author Cay Horstmann */

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

public class PostTest
{  public static void main(String[] args)
   {  try
      {  String fileName;
         if (args.length > 0)     fileName = args[0];
         else                     fileName = "PostTest.properties";

          Properties props = new Properties();
         FileInputStream in = new FileInputStream(fileName);
         props.load(in);

         URL url = new URL(props.getProperty("URL"));
         props.remove("URL");
         String r = doPost(url, props);
         System.out.println(r);
      }
      catch (IOException exception)
      {  System.out.println("Error: " + exception); }
   }

   public static String doPost(URL url, Properties nameValuePairs) throws IOException
   {  URLConnection connection = url.openConnection();
      connection.setDoOutput(true);

      PrintWriter out = new PrintWriter(connection.getOutputStream());

      Enumeration enum = nameValuePairs.keys();
```

```
while (enum.hasMoreElements())
    {  String name = (String)enum.nextElement();
       String value = nameValuePairs.getProperty(name);
       char ch;
       if (enum.hasMoreElements()) ch = '&'; else ch = '\n';
       out.print(name + "="  + URLEncoder.encode(value) + ch);
    }

    out.close();

    BufferedReader in=null;
    try
    {  in = new BufferedReader(new
         InputStreamReader(connection.getInputStream()));
    }
    catch (FileNotFoundException exception)
    { /*
//caratteristica 1.2
     InputStream err = ((HttpURLConnection)connection).getErrorStream();
       if (err == null) throw exception;
       in = new BufferedReader(new InputStreamReader(err));
*/
    }
    StringBuffer response = new StringBuffer();
    String line;

    while ((line = in.readLine()) != null)
       response.append(line + "\n");

    in.close();
    return response.toString();
  }
}
```

15.8 Firma di un applet

/** * By default, this applet raises a security exception, unless you configure your policy to allow applets from its location to write to the file "writetest.txt". */

```
import java.awt.*; import java.io.*;
import java.lang.*; import java.applet.*;

public class WriteFile extends Applet {
String myFile = "writetest.txt";
File f = new File(myFile);
DataOutputStream dos;

public void init() {
String osname = System.getProperty("os.name");
 }

public void paint(Graphics g) {
try
{
dos = new DataOutputStream
(new BufferedOutputStream(
new FileOutputStream(myFile),128));
dos.writeChars("Cats can hypnotize you when you least expect it\n");
dos.flush();
g.drawString("Successfully wrote to the file named " + myFile + " -- go take a
look at it!", 10, 10);
}
catch (SecurityException e)
{ g.drawString("writeFile: caught security exception: " + e, 10, 10); }
catch (IOException ioe)
{ g.drawString("writeFile: caught i/o exception",10, 10);
} } }
```

> jar cvf WriteFile.jar *.class

> jar tvf WriteFile.jar
   0 Sat Oct 30 20:55:58 GMT+01:00 2004 META-INF/
   68 Sat Oct 30 20:55:58 GMT+01:00 2004 META-INF/MANIFEST.MF
   1586 Sat Oct 30 20:54:24 GMT+01:00 2004 WriteFile.class

>keytool -genkey -keyalg rsa -alias stefanochiave -keystore archiviostefano.jks
Immettere la password del keystore: pippoa
Specificare nome e cognome
 [Unknown]: Stefano Marcugini
Specificare il nome dell'unitÓ aziendale
 [Unknown]: DipMat
Specificare il nome dell'azienda
 [Unknown]: UniPg
Specificare la localitÓ
 [Unknown]: Pg
Specificare la provincia
 [Unknown]: Pg
Specificare il codice a due lettere del paese in cui si trova l'unitÓ
 [Unknown]: IT
Il dato CN=Stefano Marcugini, OU=DipMat, O=UniPg, L=Pg, ST=Pg, C=IT Þ corretto?
 [no]: si

Immettere la password della chiave per <stefanochiave>
	(INVIO se corrisponde alla password del keystore): pippoc

> keytool -certreq -alias stefanochiave -keystore archiviostefano.jks -sigalg SHA1WithRSA -file richiestacertificato.p10

C:\ejbca20>ra adduser stefano pippoc "C=IT, O=University, CN=Stefano Marcugini"
null null 1 1
Trying to add user:

Username: stefano
Password (hashed only): pippoc
DN: C=IT, O=University, CN=Stefano Marcugini
SubjectAltName: null
Email: null
Type: 1
Token: 3
Certificate profile: 1
End entity profile: 1
User 'stefano' has been added.

Note: If batch processing should be possible,
also use 'ra setclearpwd stefano <pwd>'.

C:\ejbca20>ra finduser stefano
Found user:
username=stefano
password=null
dn="CN=Stefano Marcugini,O=University,C=IT"
email=null
status=10
type=1
token type=3
end entity profile id=1
certificate entity profile id=1
hard token issuer id=0
created=Sat Oct 30 21:59:11 GMT+01:00 2004
modified=Sat Oct 30 21:59:11 GMT+01:00 2004

C:\ejbca20>ra listusers 10
User: foo, "CN=foo,O=AnaTom,C=SE", "null", foo@anatom.se, 10, 1, 1
User: stefano, "CN=Stefano Marcugini,O=University,C=IT", "null", null, 10, 1, 3
C:\ejbca20>ra listusers 40
User: tsa, "CN=UniTSA", "", tsa@radek.pl, 40, 1, 4
User: tsa1, "CN=141.250.5.29,O=Univeristy,C=IT", "null", null, 40, 1, 4
User: filippo, "CN=Filippo,C=IT", "null", null, 40, 1, 1

User: paola, "CN=Paola,C=IT", "null", null, 40, 1, 1
User: admin, "CN=Admin,C=IT", "null", null, 40, 64, 1
User: 505864, "CN=505864,O=AnaTom,C=SE", "null", 505864@anatom.se, 40,
User: swede, "CN=Õõ÷,O=┼─Í,C=SE", "null", swede@anatom.se, 40, 1, 1
User: 568830, "CN=568830,O=AnaTom,C=SE", "", 568830@anatom.se, 40, 1, 2
User: 328536, "CN=328536,O=AnaTom,C=SE", "", 328536@anatom.se, 40, 1, 2
User: tomcat, "CN=127.0.0.1,O=FolignoCA,C=IT", "null", null, 40, 1, 3
User: superadmin, "CN=SuperAdmin", "null", null, 40, 65, 2

Richiesta certificato

> ca processreq stefano pippoc richiestacertificato.p10 stefano.cer

oppute tramite interfaccia web. In questo caso per recuperare il certificato in fornato .cer mi riconnetto richiedendo il certificato dell'utente

C:\ejbca20>ra listusers 10
User: foo, "CN=foo,O=AnaTom,C=SE", "null", foo@anatom.se, 10, 1, 1

C:\ejbca20>ra listusers 40
User: tsa, "CN=UniTSA", "", tsa@radek.pl, 40, 1, 4
User: tsa1, "CN=141.250.5.29,O=Univeristy,C=IT", "null", null, 40, 1, 4
User: filippo, "CN=Filippo,C=IT", "null", null, 40, 1, 1
User: paola, "CN=Paola,C=IT", "null", null, 40, 1, 1
User: admin, "CN=Admin,C=IT", "null", null, 40, 64, 1
User: 505864, "CN=505864,O=AnaTom,C=SE", "null", 505864@anatom.se, 40,
User: swede, "CN=Õõ÷,O=┼─Í,C=SE", "null", swede@anatom.se, 40, 1, 1
User: 568830, "CN=568830,O=AnaTom,C=SE", "", 568830@anatom.se, 40, 1, 2
User: 328536, "CN=328536,O=AnaTom,C=SE", "", 328536@anatom.se, 40, 1, 2
User: tomcat, "CN=127.0.0.1,O=FolignoCA,C=IT", "null", null, 40, 1, 3
User: superadmin, "CN=SuperAdmin", "null", null, 40, 65, 2
User: stefano, "CN=Stefano Marcugini,O=University,C=IT", "null",null,40,1,3

Scaricamento file CA

> keytool -import -alias cacert -file CorsoSprecializzazione.pem -keystore archiviostefano.jks

> keytool -import –trustcacerts -alias stefanochiave -file stefano.cer -keystore archiviostefano.jks
Immettere la password del keystore: pippoa
Immettere la password della chiave per <stefanochiave>pippoc
La risposta del certificato Þ stata installata nel keystore

> jarsigner -keystore archiviostefano.jks WriteFile.jar stefanochiave
Enter Passphrase for keystore: pippoa
Enter key password for chiavestefano: pippoc

```
>jarsigner -verify  -verbose -certs WriteFile.jar


136 Mon Nov 01 00:55:02 GMT+01:00 2004 META-INF/MANIFEST.MF
        189 Mon Nov 01 00:55:02 GMT+01:00 2004 META-
INF/STEFANOC.SF
       1590 Mon Nov 01 00:55:02 GMT+01:00 2004 META-
INF/STEFANOC.RSA
          0 Sun Oct 31 23:24:04 GMT+01:00 2004 META-INF/
sm      1586 Sun Oct 31 03:34:54 GMT+01:00 2004 WriteFile.class

      X.509, C=IT, O=University, CN=Stefano1
      X.509, C=IT, O=FolignoCA, CN=CorsoSprecializzazione


  s = signature was verified
  m = entry is listed in manifest
  k = at least one certificate was found in keystore
  i = at least one certificate was found in identity scope

jar verified.


<APPLET CODE="WriteFile.class" archive="WriteFile.jar" WIDTH=800
HEIGHT=250>
```

15.9 Connessione a una terza pagina Web

```
/* version 1.10 1999-08-27       * author Cay Horstmann  */

import java.net.*;  import java.io.*;              import java.util.*;
import java.awt.*;  import java.awt.event.*;  import java.applet.*;

public class WeatherApplet extends Applet  implements ActionListener
{
private TextArea weather;   private java.awt.List state;     private java.awt.List report;




 public void init()
   {
     setLayout(new BorderLayout());
      Panel listPanel = new Panel();

      state = new java.awt.List(6,false);
      for (int i = 0; i < 50; i++)  state.addItem(states[i][0]);
      listPanel.add(state);

      report = new java.awt.List(6,false);
      for (int i = 0;  i < 12; i++)  report.addItem(reports[i][0]);
      listPanel.add(report);
     add( "North", listPanel);

      weather = new TextArea(20, 80);
      weather.setFont(new Font("Courier", Font.PLAIN, 12));
      add("Center", weather);

     Button reportButton = new Button("Get report");
      reportButton.addActionListener(this);
      add("South", reportButton);
   }

  public String getItem(java.awt.List list, String[][] items)
```

```
  {  return items[list.getSelectedIndex()][1];}

  public void actionPerformed(ActionEvent evt)
  {  weather.setText("");
     getWeather(getItem(state, states), getItem(report, reports));
  }

  private String[][] states =
     {  { "Alabama", "al" },
        { "Alaska", "ak" },
. . .
     };

  private String[][] reports =
     {  { "Hourly (State Weather Roundup)", "hourly" },
        { "State Forecast", "state" },
        { "Zone Forecast", "zone" },
. . .
     };
  // Put together the URL query and go get the data from it
  public void getWeather(String state, String report)
  {
   String r = new String();
     try
     {  String queryBase = getParameter("queryBase");
        String query = queryBase + state + "/" + report + ".html";
        URL url = new URL(query);
        BufferedReader in = new BufferedReader(new
           InputStreamReader(url.openStream()));

        String line;
        while ((line = in.readLine()) != null)
           weather.append(removeTags(line) + "\n");
     }
     catch(IOException e)
     {  showStatus("Error " + e); }
  }

  public static String removeTags(String s)
  {  while (true)
     {  int lb = s.indexOf('<');
        if (lb < 0) return s;
        int rb = s.indexOf('>', lb);
```

```
        if (rb < 0) return s;
        s = s.substring(0, lb) + " " + s.substring(rb + 1);
      }
    }
}
```

Per le violazioni di sicurezza:
- File di policy WeatherApplet.policy:
grant
{ permission java.net.SocketPermission "iwin.nws.noaa.gov:80", "connect";};

appletviewer -J-Djava.security.policy=WeatherApplet.policy WeatherApplet.html

- registazione, verifica della firma, acettazione dei prvilegi richiesti

- server proxy: il client accede all'applet che richiede al server proxy di accedere alle altre pagine.