Programmare in Java con GNU/Linux

di Giacomino Timillero

1 Introduzione

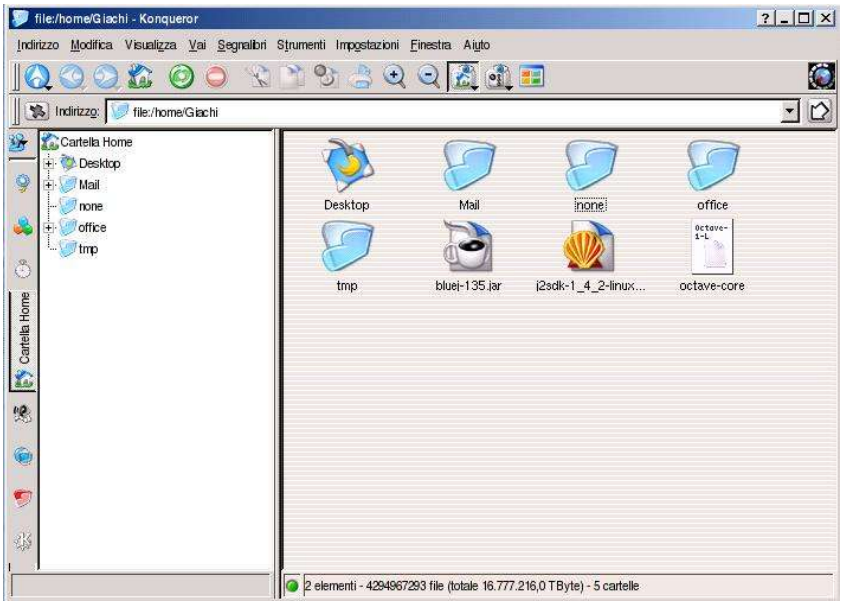
Questa guida è indirizzata a chi intende realizzare programmi in Java all'interno di un sistema operativo GNU/linux .

Questa guida ha lo scopo di indicare al lettore in modo chiaro ed elementare quali programmi utilizzare, come reperirli e come installarli per programmare in Java.

Questa guida non si propone di insegnare a programmare in Java , quindi non verranno trattati concetti inerenti la programmazione.

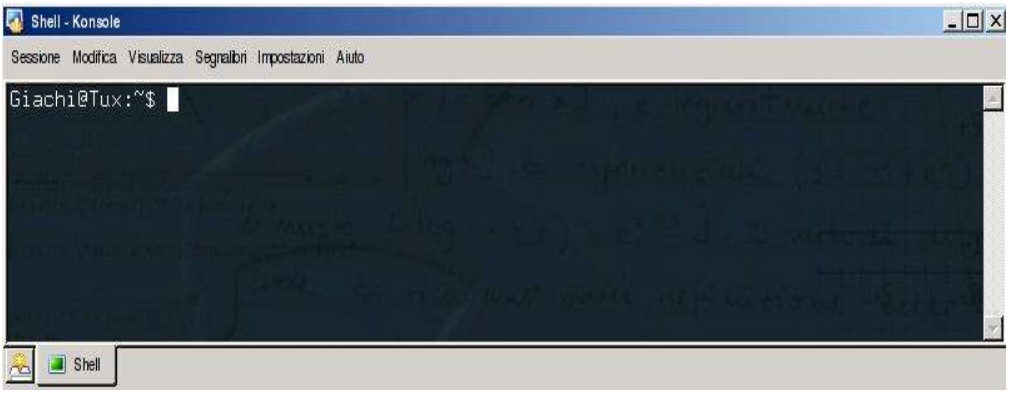
2 Programmi necessari

I programmi di cui avremo bisogno sono 2 : Java 2 sdk e BlueJ.
Il primo, chiamato anche J2SE , consente di compilare i programmi realizzati in java . Lo si può scaricare gratuitamente dal sito http://java.sun.com.
Attualmente il collegamento è http://java.sun.com/j2se/1.4.2/download.html
ed il file da scaricare si chiama j2sdk-1_4_2_04-linux-i586.bin, di dimensioni pari a 34,17MB. Non bisogna far confusione con il programma J2re, che può eseguire programmi in java ma non li può compilare.
Il secondo programma è un'ambiente di sviluppo che funziona appoggiandosi al Java2sdk. Lo scopo principale di BlueJ è quello di fornire al programmatore un'interfaccia semplice per la programmazione.
Il programma è disponibile per vari sistemi operativi; la versione per linux ha un nome sul tipo bluej-xxx.jar , dove xxx è il numero della versione.
Lo potete prelevare dal sito http://www.bluej.org .
Salviamo i due file nella nostra directory home, come in figura.



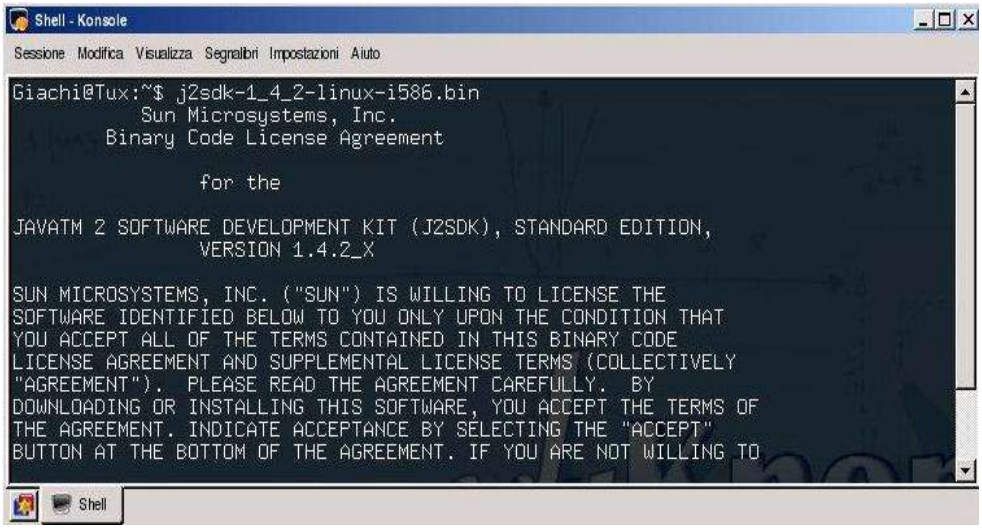
3 Installare j2sdk

Apriamo un terminale (Shell) cliccando sull'apposita icona (se disponibile) presente nel panello, in basso a sinistra , oppure cercandolo nel menù dei programmi. Il terminale apparirà con una schermata simile a quella sottostante, indicando il nome dell'utente e della macchina, e presentando un cursore in attesa di informazioni.

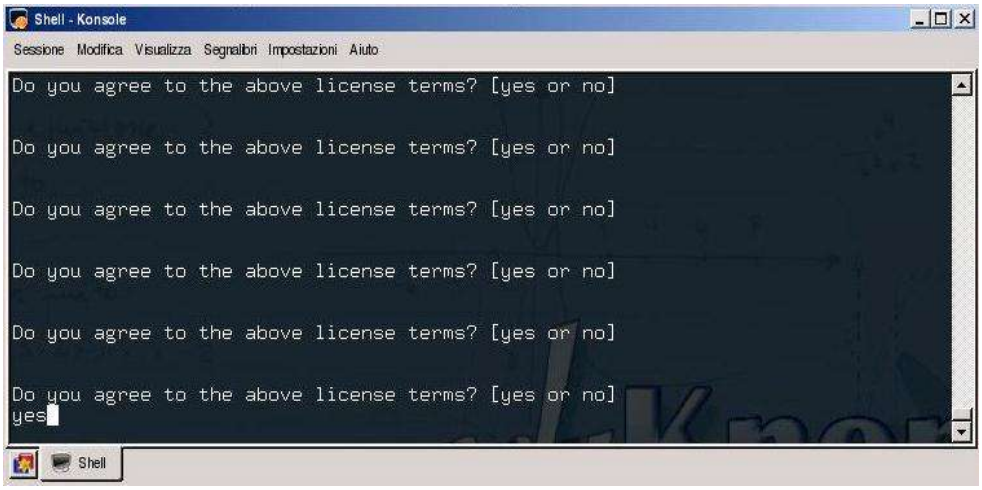


Scriviamo il nome del file j2sdk da installare:

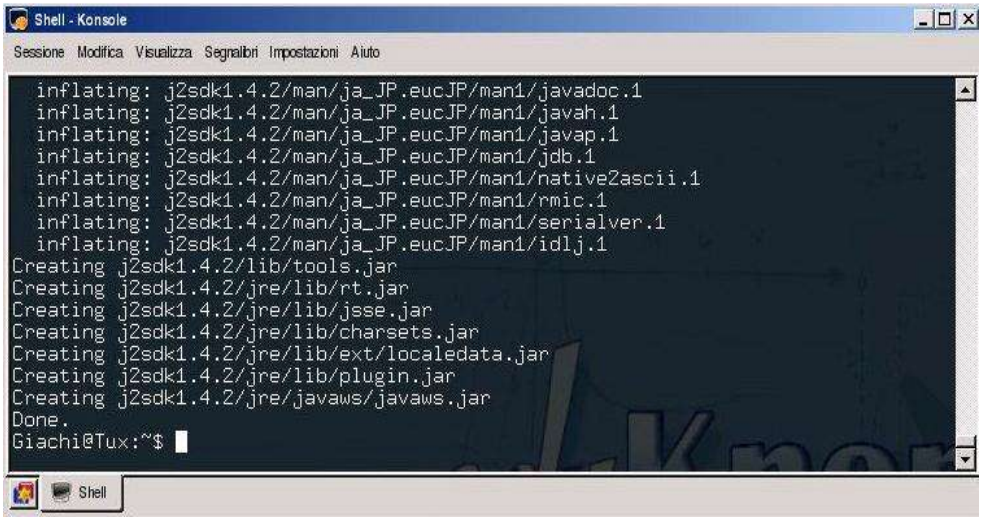
Premiamo quindi “invio” : il programma visualizzerà la licenza d'uso.



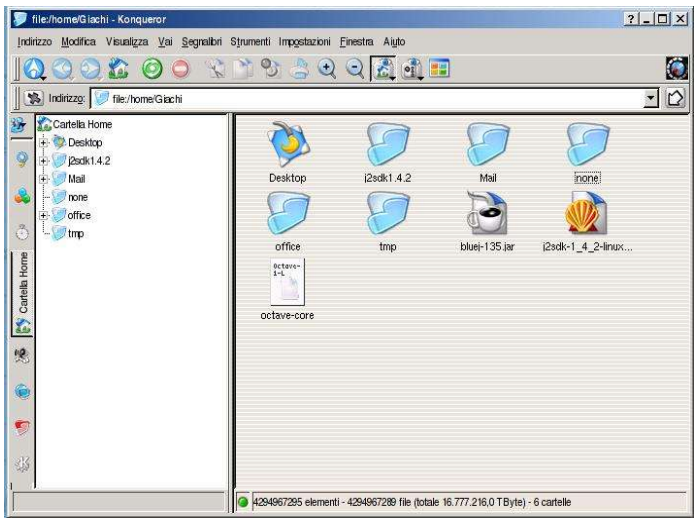
Teniamo premuto il tasto “invio” finché non ci verrà chiesto se accettiamo o no i termini della licenza.



Digitiamo quindi “yes” e premiamo “invio”: il programma inizierà l'installazione.



Concluso il processo potremo osservare che nella nostra directory “home” ora appare una nuova cartella chiamata j2sdk1.x.x :

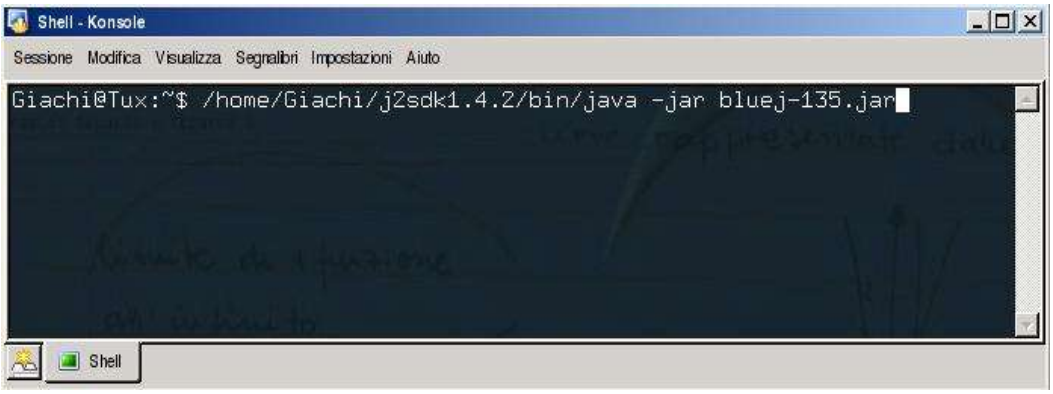


4 Installare BlueJ

Ora andremo ad installare l'ambiente di sviluppo BlueJ . Il file bluej-xxx.jar , che prima avevamo scaricato nella directory /home , utilizzerà proprio il pacchetto java appena installato per essere eseguito.
Da terminale digiterete quindi

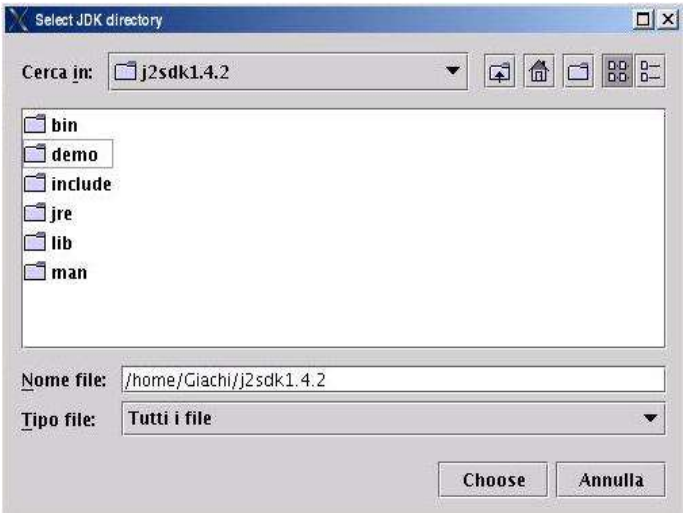
/home/*nome_utente*/j2sdk1.x.x/bin/java -jar bluej-xxx.jar

in modo analogo all'esempio sottostante:



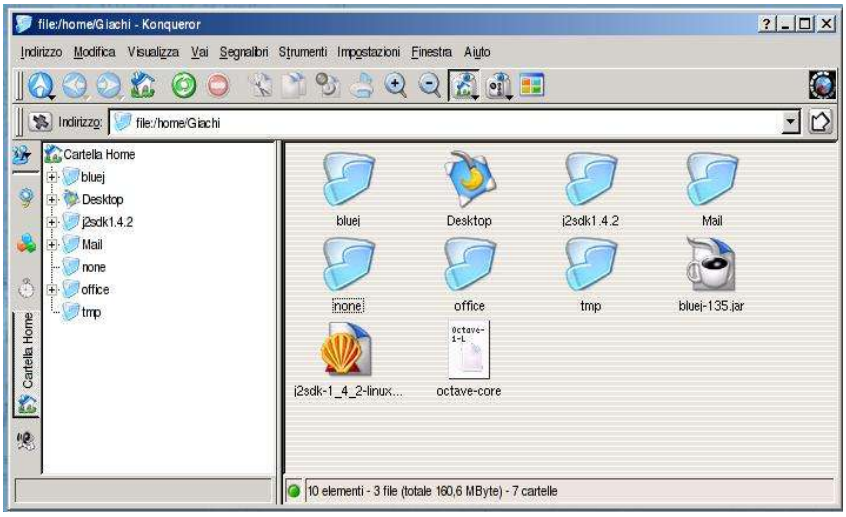
Premendo “invio” inizierà l'installazione del programma.

Ci verrà chiesta la directory in cui andrà installato il programma (“Directory to install to:”) e la directory in cui si trova il compilatore java (java JDK directory).
Nel primo caso lasciamo pure il percorso che appare automaticamente, mentre nel secondo caso clicchiamo sul tasto “Browse”: si aprirà un'altra finestra in cui dovremo specificare la directory j2sdk1.x.x che si era originata in precedenza.



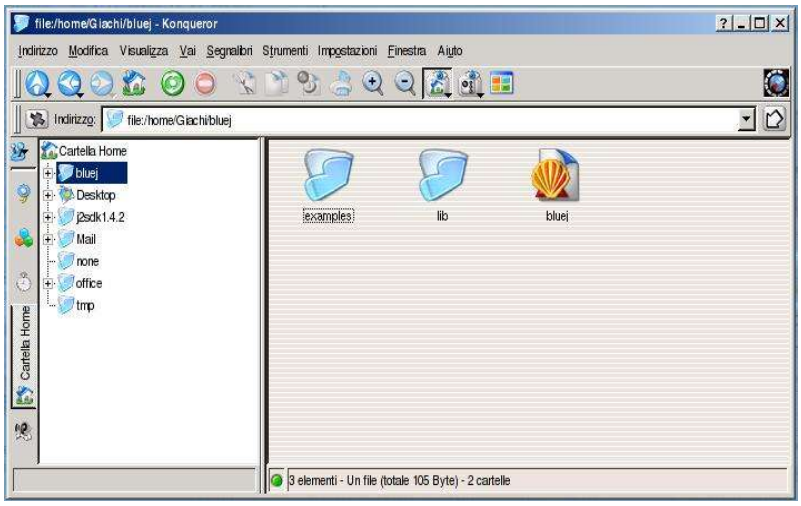
Premiamo quindi su “Choose” e la finestra si chiuderà.

Ora potremo dare avvio all'installazione cliccando su “install”. Alla fine troveremo nella directory “home” una nuova cartella chiamata “bluej”.

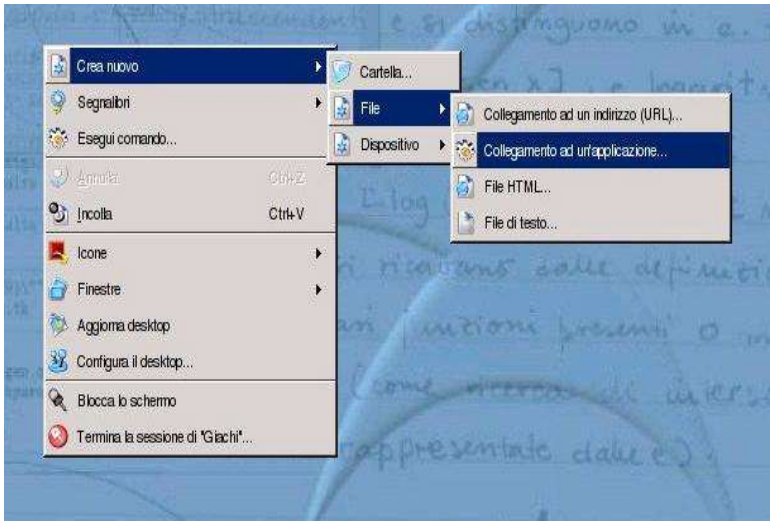


5 Creare un collegamento a BlueJ

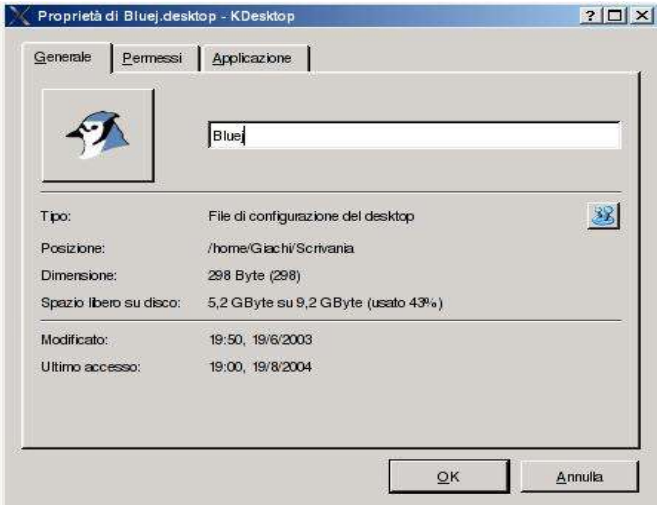
Entrando all'interno della cartella “bluej” precedentemente creata noteremo un file chiamato “bluej”, che costituisce l'eseguibile ; ci basterà quindi cliccare su questo file per avviare il programma.



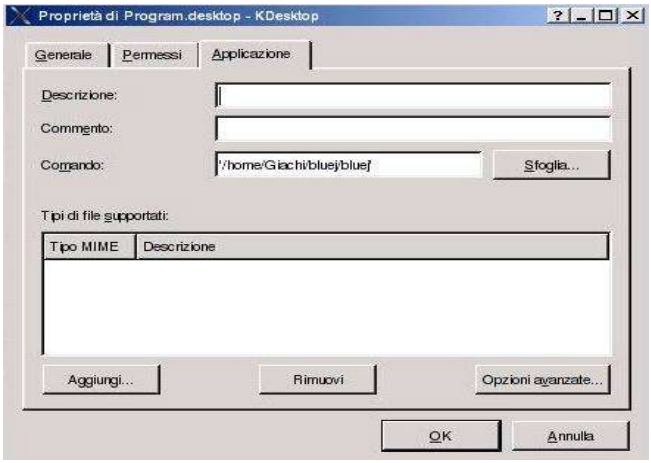
Se utilizzerete spesso Bluej, può risultare scomodo dover ogni volta andare a scovare questo file.
Potete quindi creare un collegamento sulla scrivania.
Nella maggior parte delle distribuzioni GNU/Linux l'operazione si esegue cliccando col tasto destro del mouse in un'area libera del desktop e selezionando la creazione di un nuovo collegamento ad un'applicazione.



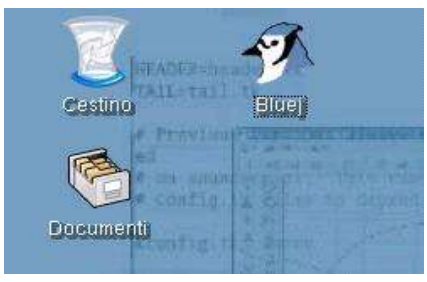
Nell'immagine di esempio successiva , ottenuta utilizzando la distribuzione “eduknoppix” (www.eduknoppix.org), potete notare che si è aperta una finestra con 3 linguette.



In quella denominata “Generale” possiamo inserire il nome che vogliamo attribuire all'applicazione ed impostare, cliccando sul quadratino, l'icona che verrà visualizzata sulla scrivania; quella che vediamo nell'esempio l'ho realizzata rubandola (con un programma per fotografare lo schermo ed uno per realizzare icone) dalla schermata che appare avviando Bluej .
Generalmente in GNU/linux le icone vengono conservate in /usr/share/icons, quindi in tale cartella potremo trovarne una che eventualmente fa al caso nostro.
Fatto ciò clicchiamo su “Applicazione”

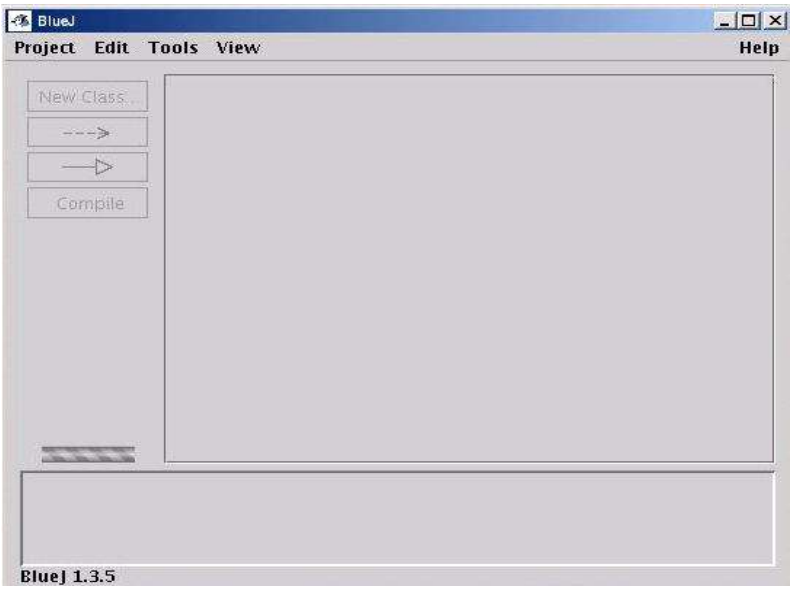


Sulla riga “Comando” dovremo specificare, sfruttando l'opzione “Sfoglia”, il percorso con cui si accede all'eseguibile di BlueJ menzionato prima.
Quindi clicchiamo su “OK” : ora sul Desktop avremo la vostra bella icona di BlueJ!

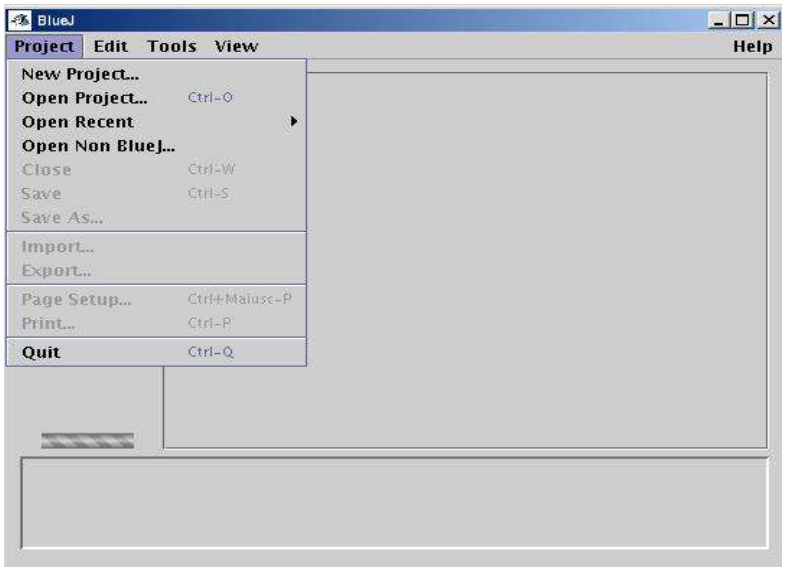


6 Utilizzare BlueJ

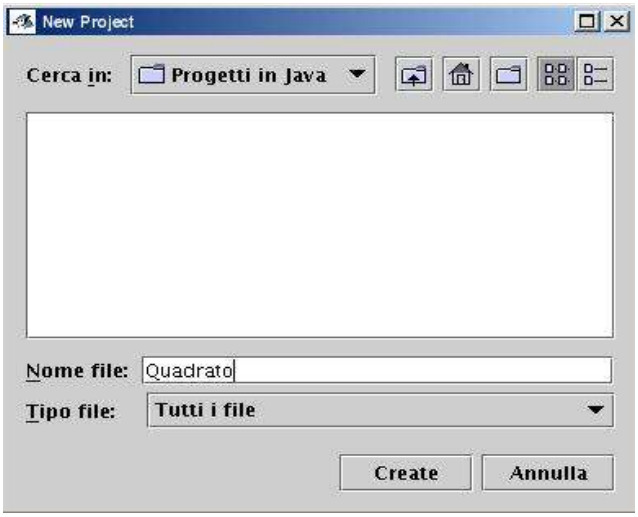
Ora analizzeremo come scrivere e compilare un programma in Java mediante BlueJ, presupponendo che conosciate già i concetti base di programmazione di tal linguaggio.
Per chi desidera approfondire le sue conoscenze su BlueJ può consultare la guida presente all'indirizzo www.**bluej**.org/tutorial/tutorial-italiano.pdf
Aprite BlueJ cliccando sul file /home/*nome_utente*/bluej/bluej menzionato nel precedente paragrafo o sul collegamento che avete creato sulla scrivania.
Terminato l'avvio del programma ci si presenterà una schermata simile a quella sottostante.



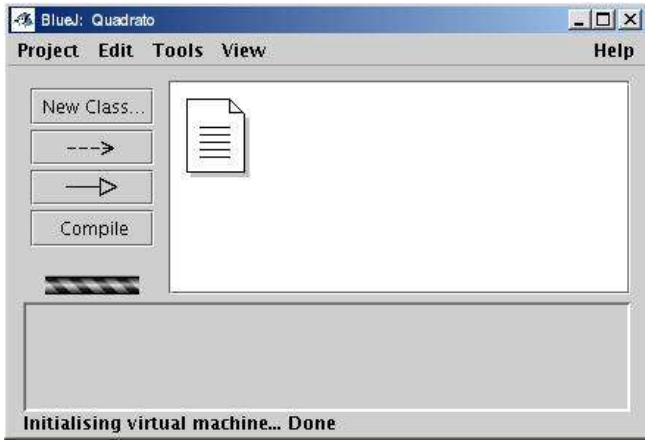
Adesso creiamo un nuovo progetto aprendo il menù “Project” e scegliendo “New Project”.



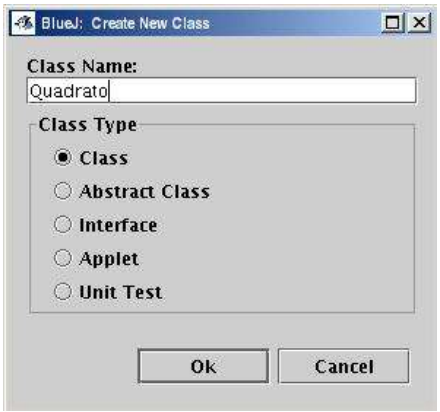
Nella finestra che si aprirà digitiamo il nome del nostro progetto.
Nell'esempio ho chiamato il progetto “Quadrato” e l'ho posto all'interno di una cartella che ho denominato “Progetti in java”



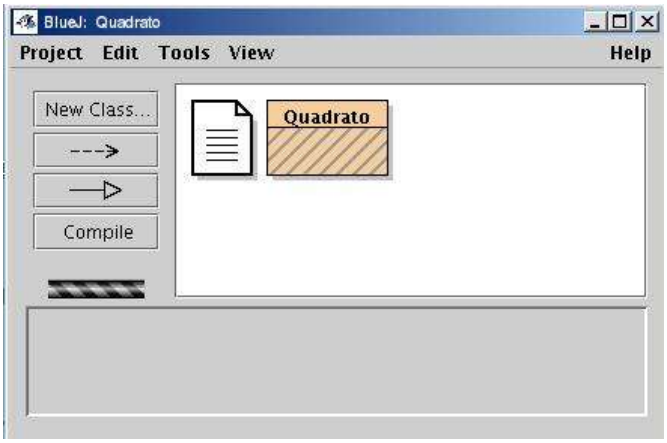
Nella schermata del programma comparirà un foglio stilizzato.



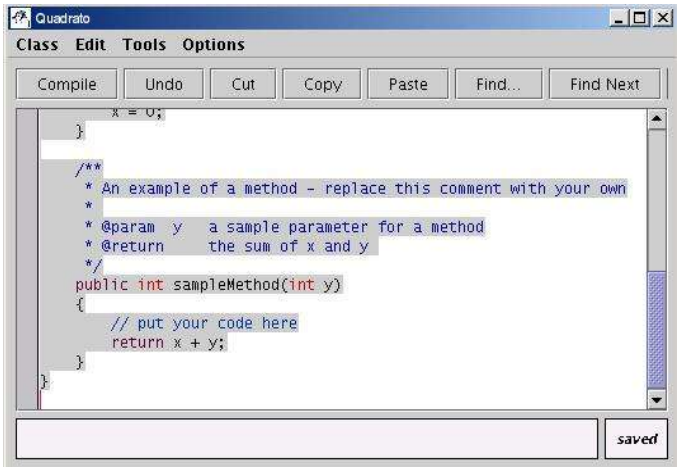
Dentro a questo progetto creiamo ora una nuova classe, cliccando su “New Class...” . Nella finestra che si aprirà inseriamo il nome della classe, mantenendo selezionata “Class”.



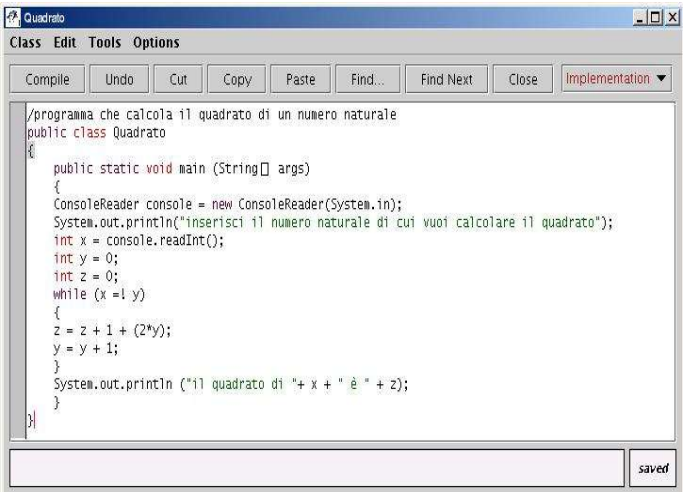
Clicchiamo quindi su “OK”.
Nella schermata principale apparirà un rettangolo col nome della classe appena creata.



Clicchiamo quindi su di esso: vedremo quindi apparire una finestra che recherà il listato di un programma di esempio.



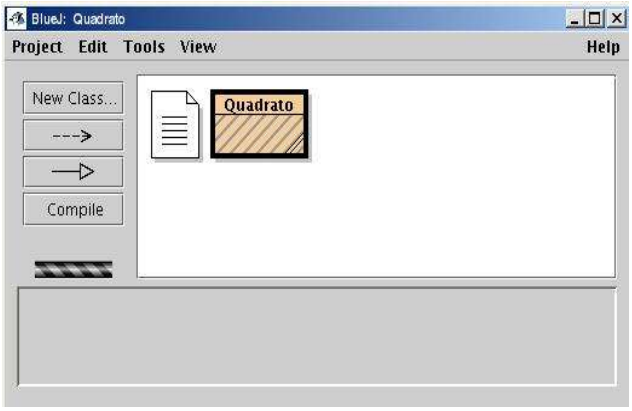
Lo cancelliamo e ci scriviamo al suo posto il nostro programma di esempio che calcola il quadrato di un numero naturale.



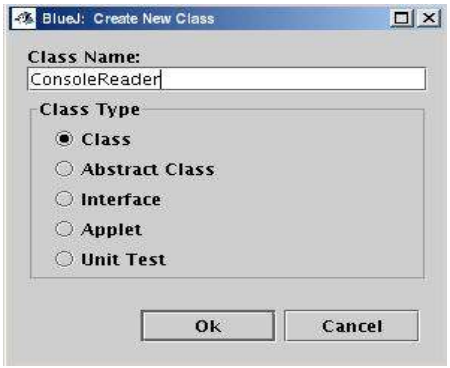
Osservate che quando il cursore si trova subito a destra di una parentesi graffa chiusa BlueJ evidenzia in grigio qual è la parentesi graffa aperta corrispondente. Questa funzione è molto utile perché uno degli errori di stesura più frequenti riguarda le parentesi graffe.

Fatto ciò entriamo nel menù “Class” e salviamo con “Save”.

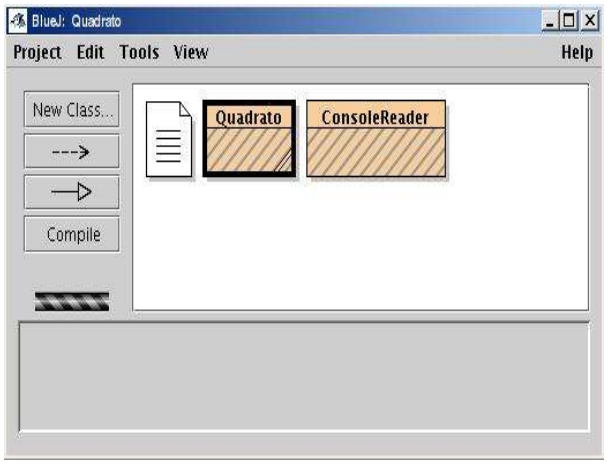
Quindi premiamo “Close” in alto a destra: ritorneremo così al menù principale.



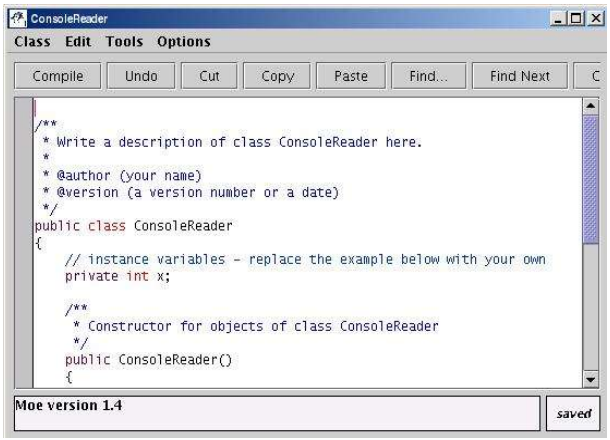
Come avrete notato, nel programma si è invocato un metodo appartenente a ConsoleReader, la celebre classe con cui si può gestire l'immissione di dati da tastiera.
Inseriamo quindi questa classe agendo sul pulsante “New Class...” e digitando “ConsoleReader” (attenzione alle maiuscole!) nella schermata che si aprirà.



Diamo l'OK. Osserveremo che nel menù principale sarà comparso un nuovo rettangolo chiamato appunto “ConsoleReader”.



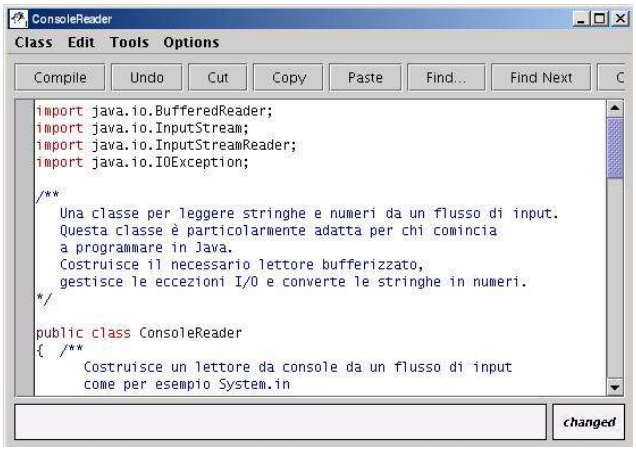
Cliccandolo entreremo dentro a questa classe dove potremo osservare nuovamente che il programma ha creato un listato di aiuto.



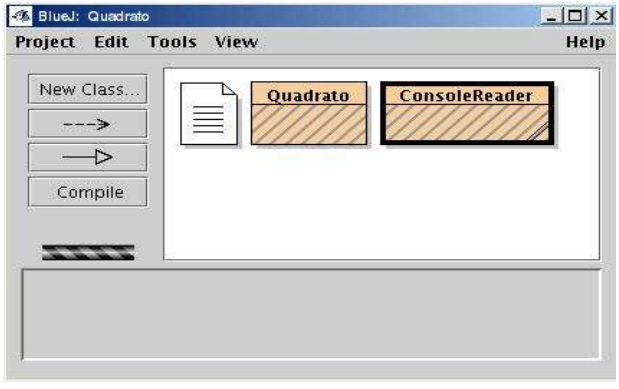
Cancelliamolo ed inseriamo al suo posto questo listato, che potete prelevare all'indirizzo http://www.dei.unipd.it/~shamano/FI1/ConsoleReader.java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.IOException;
/**
   Una classe per leggere stringhe e numeri da un flusso di input.
   Questa classe è particolarmente adatta per chi comincia
   a programmare in Java.
   Costruisce il necessario lettore bufferizzato,
   gestisce le eccezioni I/O e converte le stringhe in numeri.
*/
public class ConsoleReader
{  /**
      Costruisce un lettore da console da un flusso di input
      come per esempio System.in
      @param inStream un flusso di input
   */
   public ConsoleReader(InputStream inStream)
   {  reader = new BufferedReader
         (new InputStreamReader(inStream));
   }
   /**
      Legge una riga di input e la converte in un integer.
      La riga di input non può contenere altro che un integer.
      Non sono neppure ammessi spazi vuoti.
```

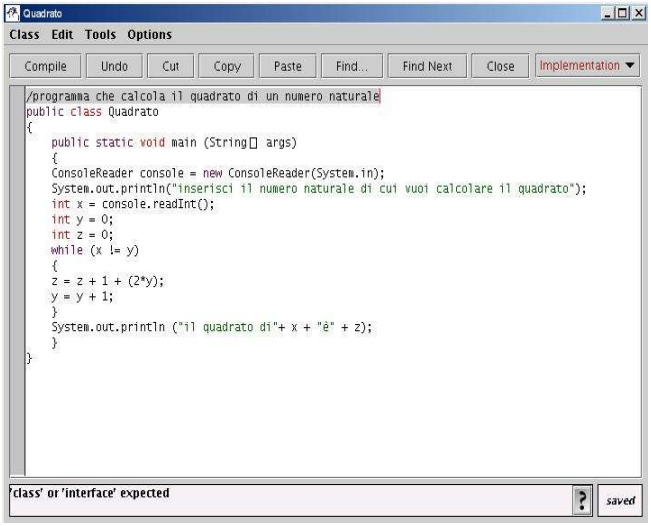
```
    @return l'integer digitato dall'utente
  */
  public int readInt()
  {  String inputString = readLine();
    int n = Integer.parseInt(inputString);
    return n;
  }
  /**
    Legge una riga di input e la converte in un numero in virgola
    Mobile. La riga di input non può contenere altro che un
    numero. Non sono neppure ammessi spazi vuoti.
    @return il numero digitato dall'utente
  */
  public double readDouble()
  {  String inputString = readLine();
    double x = Double.parseDouble(inputString);
    return x;
  }
  /**
    Legge una riga di input. Nel caso (improbabile) di
    una eccezione I/O, il programma si arresta.
    @return la riga di input digitata dall'utente, null
    alla fine dell'input
  */
  public String readLine()
  {  String inputLine = "";
    try
    {  inputLine = reader.readLine();
    }
    catch(IOException e)
    {  System.out.println(e);
      System.exit(1);
    }
    return inputLine;
  }
  private BufferedReader reader;
}
```



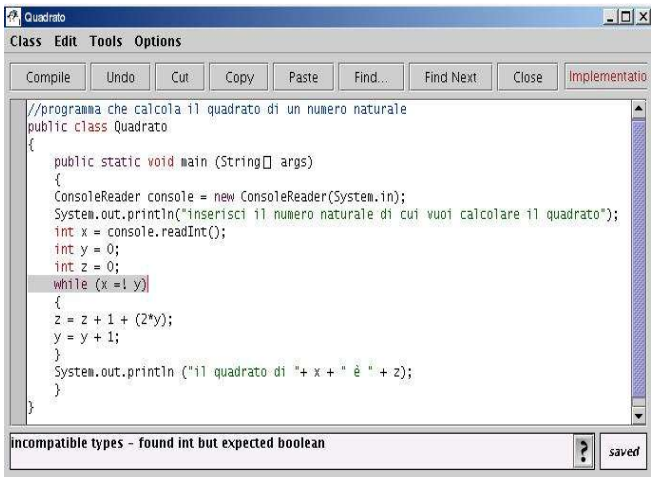
Coma fatto prima salviamo la classe agendo sul comando “Save” del menù “Class”, quindi clicchiamo su “Close” in alto a destra.
Torneremo così alla schermata principale.



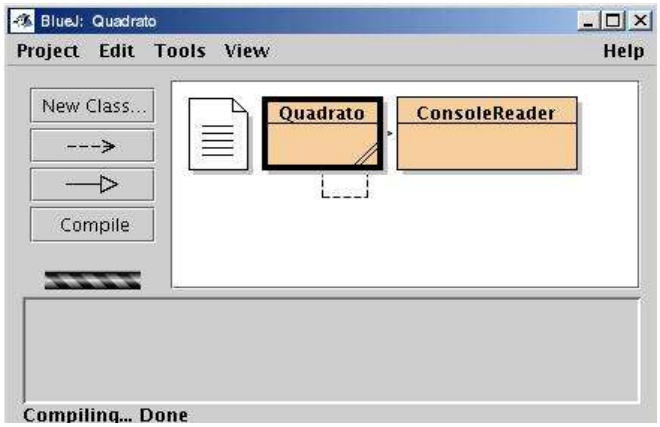
Abbiam così terminato la stesura del programma e siamo pronti per compilarlo: clicchiamo sul pulsante “Compile”.
BlueJ notificherà la presenza di un errore aprendo la classe in cui esso si trova ed evidenziandolo di grigio. Inoltre in basso verrete informati sulla natura dell'errore (**'class' or 'interface' expected**)



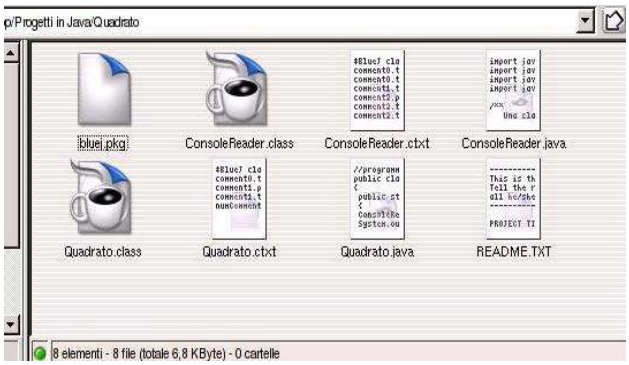
Nel commento mancava una barra. Correggiamo lo sbaglio e salviamo; quindi chiudiamo la classe con “Close” in modo da tornare nella schermata principale.
Agiamo nuovamente sul tasto “Compile”.
Questa volta verrà segnalato un'errore di incompatibilità.



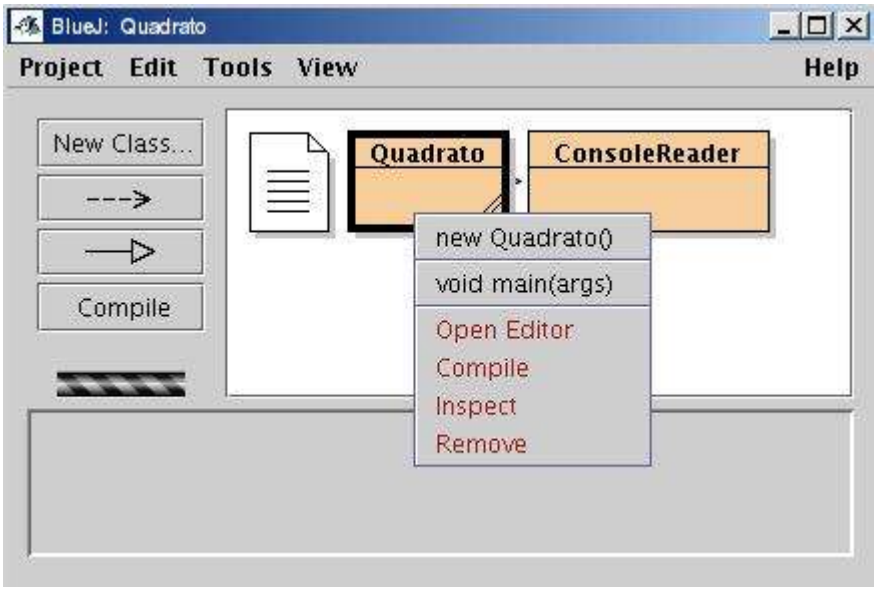
In realtà abbiam solo eseguito un errore nella battitura: abbiam scritto “**=!**” al posto di “**!=**”.
Correggiamo l'errore, salviamo, e premiamo nuovamente “Compile”:
finalmente il processo andrà a buon fine.



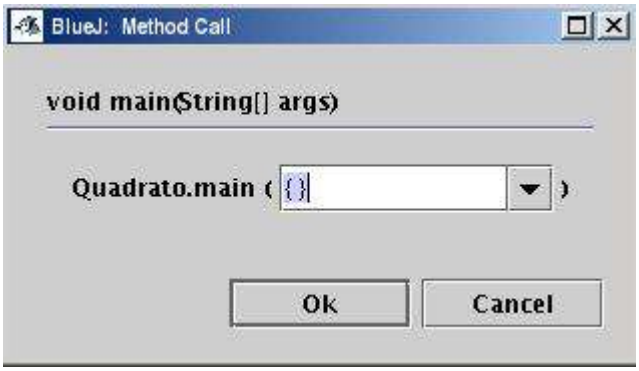
Se andiamo a guardare dentro alla cartella “Quadrato” noteremo la presenza di alcuni file tra cui quelli che abbiam appena compilato: Quadrato.class e ConsoleReader.class .



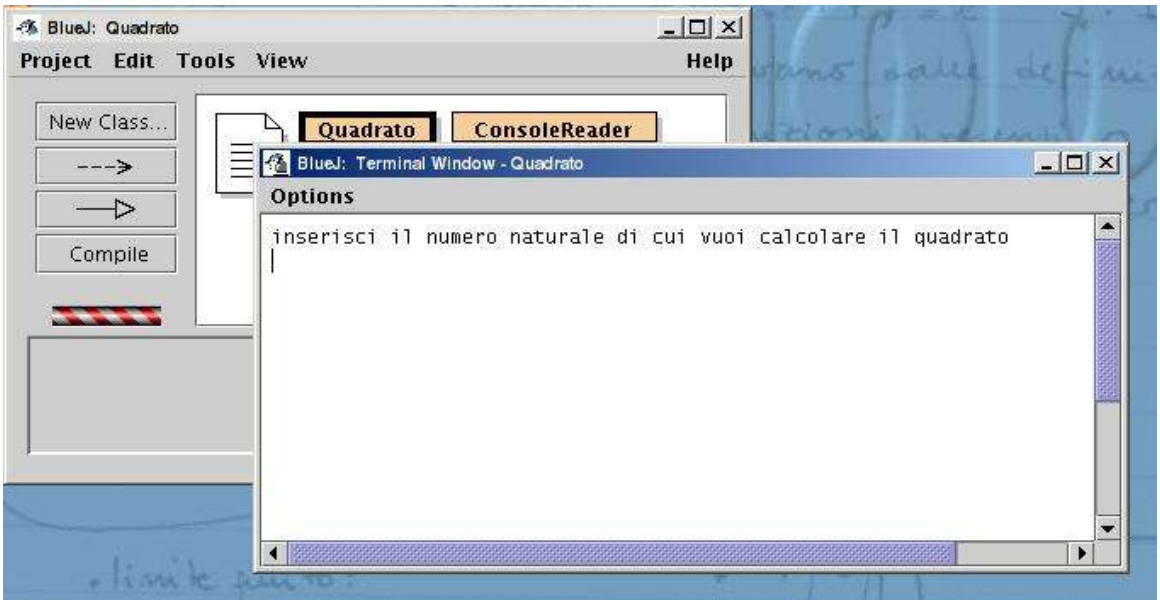
Adesso possiamo testare il nostro programma.
Posizioniamoci col puntatore del mouse sopra alla classe dove si trova il metodo main, quindi sopra alla classe “Quadrato”; cliccando col tasto destro apparirà un menù: agiamo sulla voce “void main(args)”.



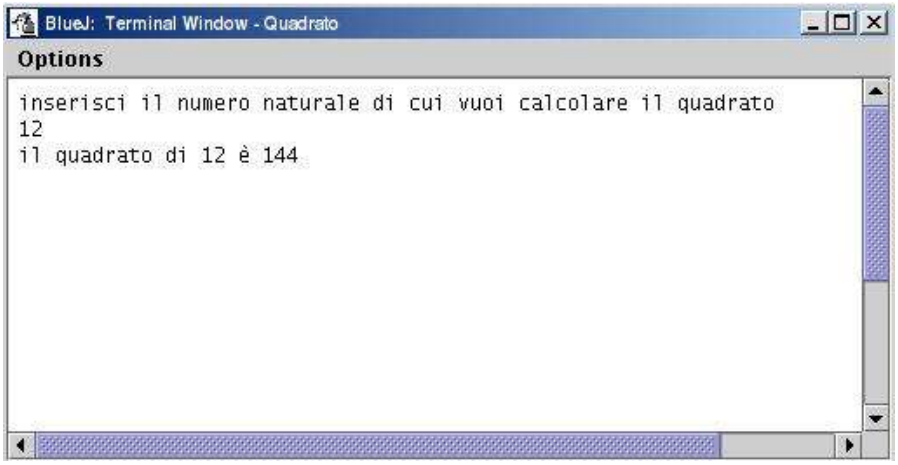
Nella finestra che si apre lasciamo così com'è e clicchiamo su “OK”.



In una nuova schermata vedremo così girare il nostro programma, che ci chiederà di inserire un numero naturale.



Inserendone uno verrà riportato il suo quadrato.



Abbiamo così concluso il nostro programma .