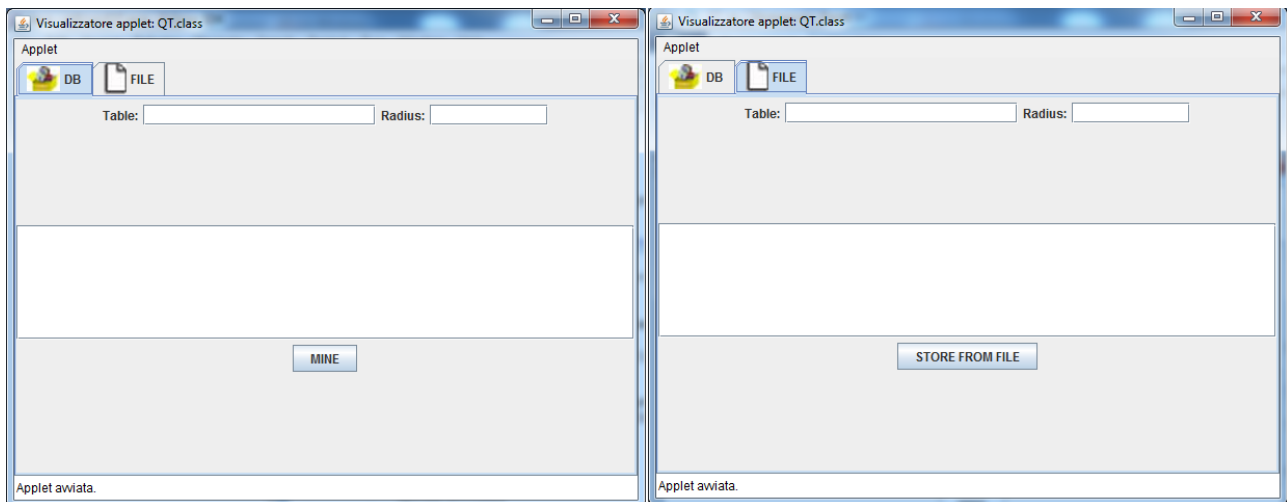


Esercitazione 8 - Java Swing e Applet

Definire la classe **QT** nel package client. Tale classe definisce una applet eseguibile in un browser web

(la visualizzazione dell'applet, tramite applet viewer, è riportata nel seguito)



Classe **QT**

Attributi

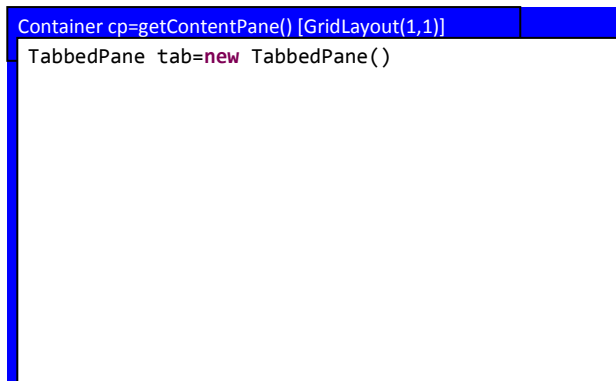
```
private ObjectOutputStream out;  
private ObjectInputStream in ;
```

Inner Class TabbedPane

Metodi

```
public void init()
```

Inizializza la componente grafica della applet istanziando un oggetto della classe **JtabbedPane** e aggiungendolo al container della applet. Inoltre avvia la richiesta di connessione al Server ed inizializza i flussi di comunicazione (membri dato **in** e **out**). Ip del server e porta sono acquisiti come parametri dalla pagina HTML.



Classe **privata** **TabbedPane** che estende **JPanel** ed è inner class di QT

Attributi

```
private JPanelCluster panelDB;  
private JPanelCluster panelFile;
```

Inner Class JPanelCluster

Metodi

TabbedPane() // fornito dal docente

Inizializza i membri panelDB e panelFile e li aggiunge ad un oggetto istanza della classe TabbedPane. Tale oggetto (TabbedPane) è quindi inserito nel pannello che si sta costruendo.

private void learningFromDBAction()**throws** SocketException, IOException, ClassNotFoundException

Acquisisce il numero di cluster (**panelDB.parameterText.getText()**). Qualora il valore acquisito non fosse un numero reale maggiore di zero visualizza un messaggio di errore all'interno di una JOptionPane (vedi esempio).

```
...  
double radius;  
try{  
    radius=new Double(panelDB.parameterText.getText()).doubleValue();  
    if (radius<=0)  
        throw new NumberFormatException("Radius<=0");  
}  
catch(NumberFormatException e){  
    JOptionPane.showMessageDialog(this,e.toString());  
    return;  
}  
...
```

Acquisisce in nome della tabella (da [panelDB.tableText](#)). Trasmette al server il comando 0 e il nome della tabella. Resta in attesa della risposta del server. In caso di risposta diversa da "OK", visualizza un messaggio di errore in una JOptionPane e termina l'esecuzione del metodo.

Altrimenti, invia il comando 1 e il raggio dei cluster da scoprire e aspetta la risposta del server. In caso di risposta diversa da "OK", visualizza un messaggio di errore in una JOptionPane e termina l'esecuzione del metodo.

Altrimenti legge il numero di cluster scoperti e i cluster così come sono trasmessi da server e li visualizza in [panelDB.clusterOutput](#). Invia al server il comando 2 e aspetta la risposta del server. In caso di risposta diversa da "OK", visualizza un messaggio di errore in una JOptionPane e termina l'esecuzione del metodo, altrimenti visualizza un messaggio che confermi il successo della attività in una JOptionPane.

```
private void learningFromFileAction() throws SocketException,  
IOException, ClassNotFoundException
```

Acquisisce il nome della tabella (da [panelFile.tableText](#)) e il raggio dei cluster (da [panelFile.parameterText](#)). Invia al server il comando 3, il nome della tabella e il raggio dei cluster e aspetta la risposta del server. In caso di risposta diversa da "OK", visualizza un messaggio di errore in una JOptionPane e termina l'esecuzione del metodo, altrimenti visualizza, in una JOptionPane, un messaggio che confermi il successo della attività.

Classe [JPanelCluster](#) che estende [JPanel](#)

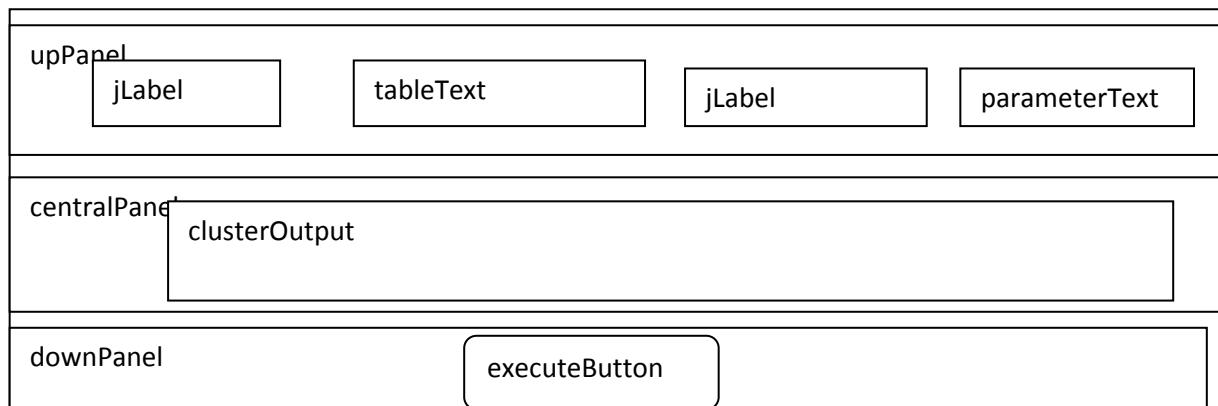
Attributi

```
private JTextField tableText=new JTextField(20);  
private JTextField parameterText=new JTextField(10);  
private JTextArea clusterOutput=new JTextArea();  
private JButton executeButton;
```

Metodi

[JPanelCluster](#)(String buttonName, java.awt.event.ActionListener a)

Comportamento: Inizializza il pannello in accordo allo schema riportato nel seguito (utilizzare opportunamente i layout). Aggiunge l'ascoltatore [a](#) al bottone [executeButton](#).



NOTE:

1. Il package `SERVER` non viene modificato
2. Le classi `Keyboard` e `MainTest` del package `client` possono essere rimosse.