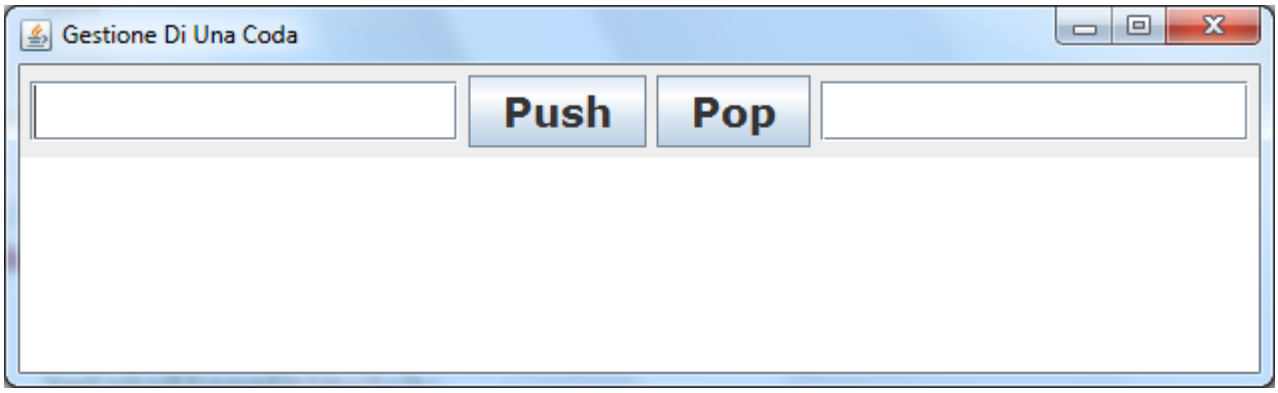
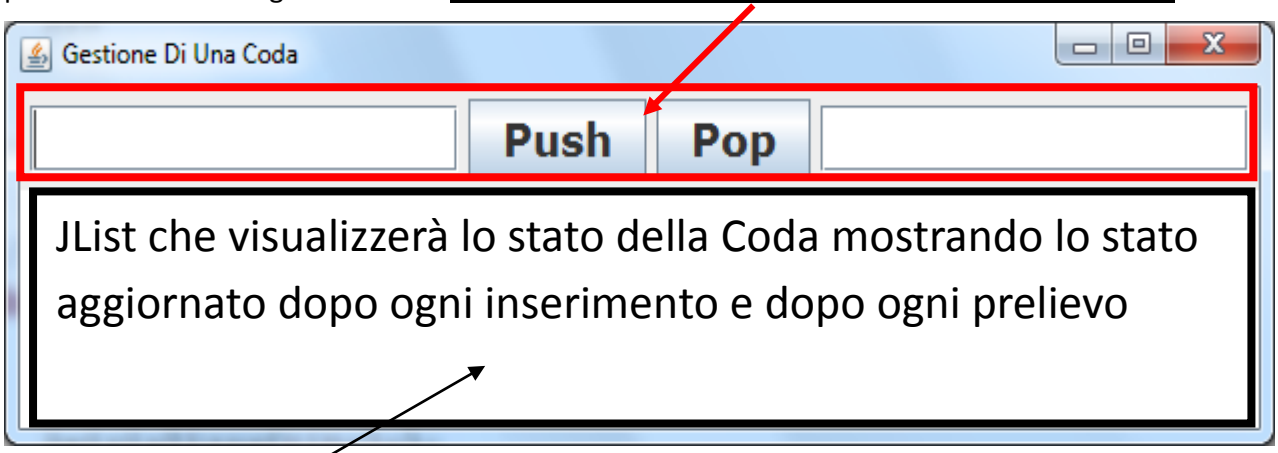
Esercitazione N°10 Gestione di una Coda con utilizzo di vettori

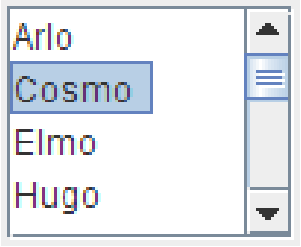
In questa esercitazione vogliamo realizzare una applicazione che implementi il funzionamento di una pila mediante un vettore. In particolare noi utilizzeremo un vettore di stringhe per semplificare le operazioni di input e di output sulla finestra, queste operazioni infatti avvengono tutte mediante stringhe come già sappiamo. l' applicazione avrà l'aspetto mostrato di seguito:



La costruzione della finestra sarà fatta utilizzando come LayoutManager il border Layout e posizionando nella regione nord un **pannello che contiene le due JTextField e i due JButton**



e nalla Regione Centro un JList



Una JList è un componente in grado di visualizzare un elenco di oggetti. All'atto della sua dichiarazione viene specificato il tipo di oggetto che visualizzerà come elemento di lista. Se il numero di elementi contenuti supera la sua dimensione mostra automaticamente una barra di scorrimento. La dichiarazione che noi andremo ad effettuare nel codice della finestra Sarà:

```
private JList<String> lstCoda;
```

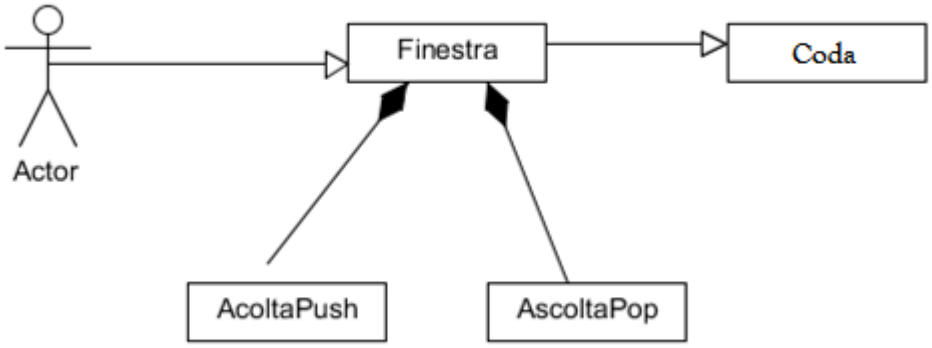
in pratica tra **< > viene** dichiarato il tipo di oggetto e lstPila è la nostra variabile di tipo JList.
Il costruttore della JList è il seguente:

list = new JList(dati); dove data è il vettore di oggetti del tipo dichiarato

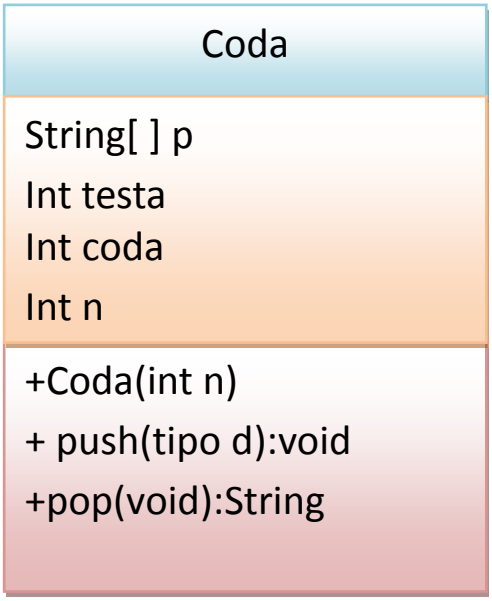
ed in ogni istante si può aggiornare il vettore di oggetti in essa contenuto invocando il metodo:

```
lstCoda.setListData(dati);
```

Lo schema delle Classi della nostra applicazione dovrebbe essere il seguente:



In pratica abbiamo la classe finestra, che costituisce l'interfaccia grafica con cui interagisce l'utente, che contiene le due sottoclassi per la gestione degli eventi sui due bottoni push e pop, e che usa la classe Coda che abbiamo già visto con il seguente TDA:



Quindi nel codice della Finestra la dichiarazione delle variabili sarà:

```
package coda;
import java.awt.BorderLayout;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;
 public class Finestra extends JFrame{
    private JFrame frame;
    private JButton btnPush;
    private JButton btnPop;
    private JTextField txfIn;
    private JTextField txfOut;
    private JPanel pnlUsaCoda;
    private JList<String> lstCoda;
    private Coda coda;
```

Il costruttore della finestra sarà:

```
//Creazione dell'applicazione.
        public Finestra(String titolo)
    {
        super(titolo);
        coda = new Coda(10);
        inizializza(titolo);
        this.setVisible(true);
    }
```

E l'inizializzazione grafica della finestra sarà:

```
//Inizializzazione dell'aspetto e del contenuto del frame.
private void inizializza()
{
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.getContentPane().setLayout(new BorderLayout());
    pnlUsaCoda= new JPanel();
    btnPush = new JButton("Push");
    btnPush.setFont(new Font("Verdana",Font.BOLD,20));
    btnPush.addActionListener(new btnPush_Click() );
    btnPop = new JButton("Pop");
    btnPop.setFont(new Font("Verdana",Font.BOLD,20));
    btnPop.addActionListener(new btnPop_Click() );
    txfIn= new JTextField(10);
    txfIn.setFont(new Font("Verdana",Font.BOLD,20));
    txfOut= new JTextField(10);
    txfOut.setFont(new Font("Verdana",Font.BOLD,20));
    pnlUsaCoda.add(txfIn);
    pnlUsaCoda.add(btnPush);
    pnlUsaCoda.add(btnPop);
    pnlUsaCoda.add(txfOut);
    lstCoda = new JList<String>(coda.p);
    lstCoda.setFont(new Font("Verdana",Font.BOLD,20));
    this.add(pnlUsaCoda, BorderLayout.NORTH);
    this.add(lstCoda, BorderLayout.CENTER);
    this.pack();
}
```

Dove l'istanza della JList viene fatta mediante il costruttore che riceve anche la lista di oggetti, che nel nostro caso è il vettore di stringhe p contenuto nella classe pila. Essendo questo vettore all'inizio vuoto la finestra si presenta all'apertura in questo modo:

Ed in seguito a inserimenti automaticamente si allarga e si restringe,grazie al codice delle due classi ascoltatrici

Per quanto riguarda gli ascoltatoti sono simili a quelli visti per la pila:

```
        * Classe che realizza l'ascoltatore dell'evento click sul bottone Push
        * implementando il metodo actionPerformed
        */
        private class btnPush_Click implements ActionListener{

            public void actionPerformed(ActionEvent e){
                coda.push(txfIn.getText());
                lstCoda.setListData(coda.p);
                ridimensiona();
            }
        }
        /*
            * Classe che realizza l'ascoltatore dell'evento click sul bottone Pop
            * implementando il metodo actionPerformed
            */
            private class btnPop_Click implements ActionListener{

                public void actionPerformed(ActionEvent e){
                    String l;
                    l=coda.pop();
                    txfOut.setText(l);
                    lstCoda.setListData(coda.p);
                    ridimensiona();
                }
            }
public void ridimensiona(){
this.pack();
}
}
```