

**Procedimento:**

- Creo il DB con il terminale.
- Creo l'interfaccia "Applicazione" come indicato dall'esercizio.
- Creo la classe "Esame1" che è il main.
- Creo la classe "ApplicazioneSwing" che sarà l'implementazione di "Applicazione".
- Creo la classe "utente" che mi rappresenta l'oggetto con cui si dovrà lavorare (nome, password).
- Dato che dovrò comunicare con il DB creo l'interfaccia "Modello" implementata dalla classe "DatabaseManager".

**Scrittura codice:**

- Ora che ho le classi, inizio scrivendo il codice di "Utente" (dato che è semplice, con setter e getter che si possono generare in automatico e il costruttore).
  - Scrivo il codice dell'"ApplicazioneSwing" creando l'interfaccia base (bottoni, campi di testo, etc senza listener) così ho qualcosa da eseguire/vedere e da (eventualmente) sistemare prima di proseguire. (ovviamente dovrò anche creare un oggetto di classe "ApplicazioneSwing" in "Esame1", il quale usando "panelloLogin()" aprirà la finestra dell'applicazione).
  - Creo i vari listener che svolgano le funzioni indicate dal testo dell'esercizio.
- Creo tutti i listener completi tranne quelli riguardanti le lingue e "valida()" il quale deve accedere al DB per controllare la presenza dell'utente.
- Vado quindi a creare il metodo "cercaUtente" nell'interfaccia "Modello" ed ad implementarlo nella classe "DatabaseManager".
  - Completo il metodo "valida()" che avevo lasciato in sospeso.

Punti 4 e 5 - lingue (potrebbe non essere svolto nel modo migliore):

- Creo la comboBox e il rispettivo listener con "Italiano" e "English".
- Per aggiungere la nuova lingua mi basta:  
Aggiungere all'elenco della comboBox, il "Klingon"  
Aggiungere al file "configurazione" le traduzioni.  
Niente di più; fine.

**ECLIPSE:**

```
esame1
    Esame1.java
esame1.database
    DatabaseManager.java
esame1.modello
    Modello.java
    Utente.java
esame1.swing
    ApplicazioneSwing.java
esame1.vista
    Applicazione.java

configurazione.properties
```

### Utente.java

```
package esame1.modello;

public class Utente {
    String nome;
    String password;

    public Utente(String nome, String password) {
        this.nome = nome;
        this.password = password;
    }

    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
}
```

### Esame1.java

```
package esame1;
import esame1.swing.ApplicazioneSwing;

public class Esame1 {
    public static void main(String[] args) {
        ApplicazioneSwing applicazione = new ApplicazioneSwing();
        applicazione.pannelloLogin();
    }
}
```

### Applicazione.java

```
package esame1.vista;
public interface Applicazione {

    public void pannelloLogin();

    public boolean valida(
        String nome,
        String password );

    public void esci();
}
```

## ApplicazioneSwing.java

```
package esame1.swing;
import java.awt.BorderLayout;
...

public class ApplicazioneSwing implements Applicazione {
    int conta = 0;
    @Override
    public void pannelloLogin() {
        try {

            // PRINCIPALE
            JPanel pan_principale = new JPanel();
            pan_principale.setLayout(new BorderLayout());

            // BASSO
            JPanel pan_basso = new JPanel();
            pan_principale.add(pan_basso, BorderLayout.SOUTH);

            // Bottone Ok
            JButton ok = new JButton();
            ok.setText("OK");
            pan_basso.add(ok, BorderLayout.WEST);
            ok.setEnabled(false);

            // Bottone Annulla
            JButton annulla = new JButton();
            annulla.setText("Annulla");
            pan_basso.add(annulla, BorderLayout.EAST);

            // CENTRO
            JPanel pan_centro = new JPanel();
            pan_centro.setLayout(new GridLayout(1, 2));
            pan_principale.add(pan_centro, BorderLayout.CENTER);

            // Logo
            ImageIcon logo = new ImageIcon("res/Logo.jpg");
            Image scaledImage = logo.getImage().getScaledInstance(130,130,Image.SCALE_DEFAULT);
            logo.setImage(scaledImage);
            JLabel img = new JLabel(logo);
            pan_centro.add(img, 0);

            //CENTRO - DESTRA
            JPanel pan_destra = new JPanel();
            pan_destra.setLayout(new GridLayout(5,1));
            pan_centro.add(pan_destra,1);

            //Campi di testo
            JLabel nomeLbl = new JLabel("Nome utente");
            JTextField nomeTxt = new JTextField();
            JLabel passwordLbl = new JLabel("Parola chiave");
            JTextField passwordTxt = new JTextField();
```

```

pan_destra.add(nomeLbl,0);
pan_destra.add(nomeTxt,1);
pan_destra.add(passwordLbl,2);
pan_destra.add(passwordTxt,3);

// ComboBox
String[] linguaLista = new String[] {"Italiano", "English", "Klingon"};
JComboBox<String> linguaCombo = new JComboBox<String>(linguaLista);
pan_principale.add(linguaCombo,BorderLayout.NORTH);

// FINESTRA
JFrame finestra = new JFrame();
finestra.setTitle("Entra nel sistema");
finestra.setLocation(0, 0);
finestra.setSize(400, 240);
finestra.add(pan_principale,BorderLayout.NORTH);
finestra.setVisible(true);

// LISTENER
// Ok (valida)
ok.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        valida(nomeTxt.getText(), passwordTxt.getText());
    }
});

// Annulla (esci)
annulla.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        esci();
    }
});

// Nome (disattiva bottone Ok)
nomeTxt.addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent e) {
        //Vuoto
    }
    @Override
    public void keyReleased(KeyEvent e) {
        abilitaBottoneOk(ok, nomeTxt, passwordTxt);
    }
    @Override
    public void keyPressed(KeyEvent e) {
        //Vuoto
    }
});

```

```

// Password (disattiva bottone Ok)
passwordTxt.addKeyListener(new KeyListener() {
    @Override
    public void keyTyped(KeyEvent e) {
        //Vuoto
    }
    @Override
    public void keyReleased(KeyEvent e) {
        abilitaBottoneOk(ok, nomeTxt, passwordTxt);
    }
    @Override
    public void keyPressed(KeyEvent e) {
        //Vuoto
    }
});

// ComboBox (cambio lingue)
linguaCombo.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JComboBox<String> combo = (JComboBox<String>) e.getSource();
        String linguaScelta = (String) combo.getSelectedItem();
        String lingua = "Lingua." + linguaScelta + ".";

        ResourceBundle resourceBundle = ResourceBundle.getBundle("configurazione");

        annulla.setText(resourceBundle.getString(lingua + "annulla"));
        ok.setText(resourceBundle.getString(lingua + "ok"));
        nomeLbl.setText(resourceBundle.getString(lingua + "nome"));
        passwordLbl.setText(resourceBundle.getString(lingua + "password"));
        finestra.setTitle(resourceBundle.getString(lingua + "titolo"));
    }
});

// Finestra (chiusura)
finestra.addWindowListener(new WindowAdapter() {
    @Override
    public void windowClosing(WindowEvent e) {
        esci();
    }
});

} catch(Throwable throwable) {
    throwable.printStackTrace();
}
}

```

```

@Override
public boolean valida(String nome, String password) {
    System.out.println("valida");
    conta++;

    Utente utente = new Utente(nome, password);
    DatabaseManager databaseManager = new DatabaseManager();

    if(databaseManager.cercaUtente(utente)) {
        System.out.println("Utente trovato!");
        System.exit(0);
        return true;
    }
    else {
        System.out.println("Utente NON trovato!");
        if(conta<3) {
            return true;
        }
        else {
            System.exit(0);
            return true;
        }
    }
}

```

```

@Override
public void esci() {
    System.out.println("esci");
    System.exit(0);
}

```

```

public void abilitaBottoneOk(JButton ok, JTextField nomeTxt, JTextField passwordTxt) {
    if(nomeTxt.getText().isEmpty() || passwordTxt.getText().isEmpty())
        ok.setEnabled(false);
    else
        ok.setEnabled(true);
}
}

```

## Modello.java

```

package esame1.modello;

public interface Modello {
    public boolean cercaUtente(Utente utente);
}

```

## DatabaseManager.java

```
package esame1.database;
import java.sql.Connection;
...
```

```
public class DatabaseManager implements Modello {
```

```
    protected String url;
```

```
    public DatabaseManager() {
```

```
        try {
            ResourceBundle resourceBundle = ResourceBundle.getBundle("configurazione");
            String clazz = resourceBundle.getString("database.class");
            url = resourceBundle.getString("database.url");
            Class.forName(clazz);

```

```
        } catch (Throwable throwable) {
            throwable.printStackTrace();
        }
    }
```

```
    @Override
```

```
    public boolean cercaUtente(Utente utente) {
```

```
        Connection connessione = null;
        PreparedStatement statement = null;
        ResultSet resultSet = null;
```

```
        try {
            connessione = connetti();
            statement = connessione.prepareStatement("SELECT * FROM UTENTI WHERE  
NOME=" + utente.getNome() + " AND PAROLA_CHIAVE=" + utente.getPassword() + "");
            resultSet = statement.executeQuery();

```

```
            if(resultSet.next())
                return true;
            return false;

```

```
        } catch (Throwable throwable) {
            throwable.printStackTrace();
            disconnetti(connessione, statement, resultSet);
        }

```

```
        return false;
    }
```

```
    public void disconnetti(Connection connection, PreparedStatement statement, ResultSet resultSet) {
```

```
        try {
            connection.close();
        } catch (Throwable throwable) {
            //Vuoto
        }
    }
```

```

    try {
        statement.close();
    } catch(Throwable throwable) {
        //Vuoto
    }

    try {
        resultSet.close();
    } catch(Throwable throwable) {
        //Vuoto
    }
}

public Connection connetti() throws SQLException {
    return DriverManager.getConnection(url);
}
}

```

### configurazione.properties

database.class = org.apache.derby.jdbc.ClientDriver  
 database.url = jdbc:derby://localhost/Users/Aldo/DBconDerby/Esame1

Lingua.Italiano annulla = Annulla  
 Lingua.Italiano.ok = Ok  
 Lingua.Italiano.nome = Nome utente  
 Lingua.Italiano.password = Parola chiave  
 Lingua.Italiano.titolo = Entra nel sistema

Lingua.English annulla = Cancel  
 Lingua.English.ok = Ok  
 Lingua.English.nome = Username  
 Lingua.English.password = Password  
 Lingua.English.titolo = Login

Lingua.Klingon annulla = QII  
 Lingua.Klingon.ok = Luq  
 Lingua.Klingon.nome = Pong user  
 Lingua.Klingon.password = Mu'wIj  
 Lingua.Klingon.titolo = Pat 'el