

## Database MySQL "Gestionale"

Si crei un'applicazione swing che preveda l'implementazione delle seguenti funzionalità: · Una procedura di login e di registrazione

· Una serie di funzionalità che consentano di inserire, modificare, rimuovere degli articoli da un inventario

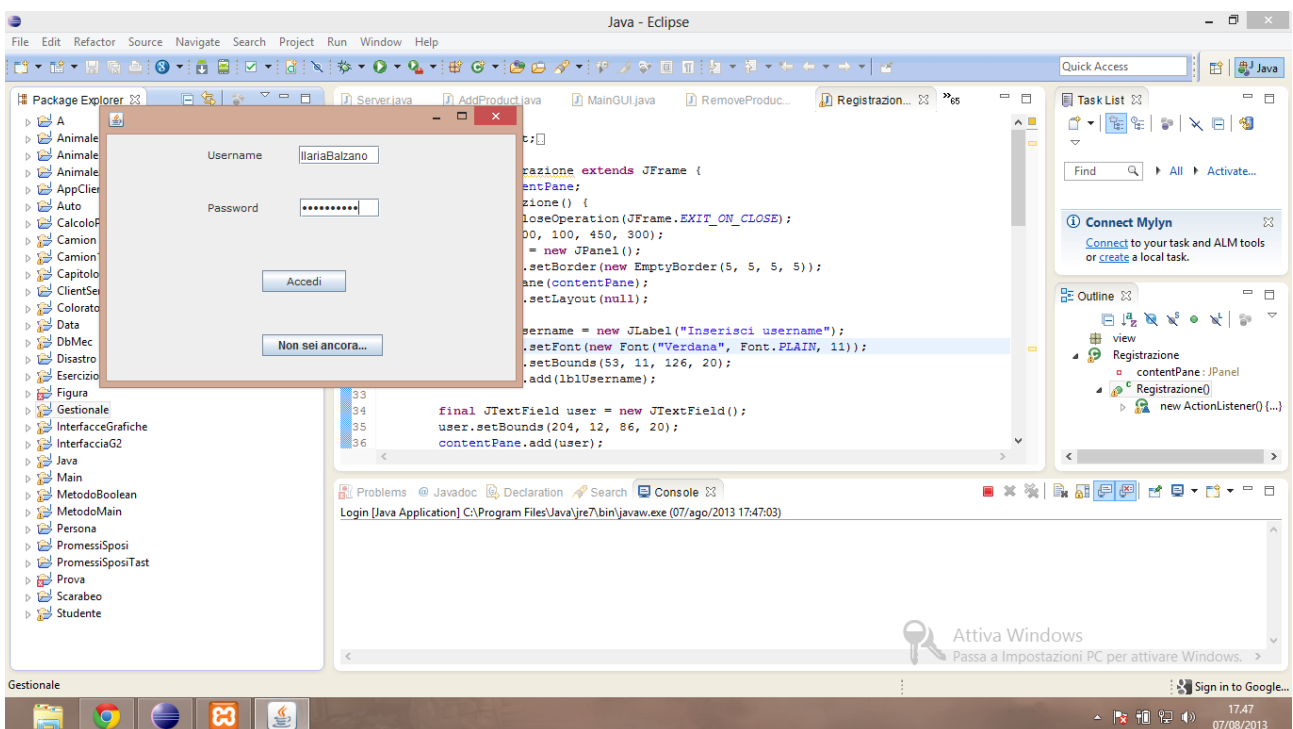
· Visualizzare lo status delle giacenze per i prodotti presenti in magazzino L'applicazione dovrà fare uso di un db mysql in particolare si prevede l'esistenza di due tabelle principali: tabella "utenti" all'interno della quale andranno salvate le seguenti informazioni:

- Nome
- Cognome
- Via
- Indirizzo di posta elettronica
- Username
- Password

Tabella "magazzino" al cui interno saranno memorizzate le seguenti informazioni: · Nome del prodotto

- Descrizione
- Livello attuale delle scorte

Opzionalmente si può prevedere per ogni articolo una soglia al di sotto della quale sarà generato un messaggio a video in cui si invita l'utente ad aumentare le scorte a disposizione



Codice sorgente:

package view;

```
import java.awt.EventQueue;

import java.awt.Font;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

import javax.swing.border.EmptyBorder;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

public class Login extends JFrame {

    private JPanel contentPane;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Login frame = new Login();
```

```

        frame.setVisible(true);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

});

}

/**
 * Create the frame.
 */
public Login() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 300);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    final JLabel lblUsername = new JLabel("Username");
    lblUsername.setFont(new Font("Verdana", Font.PLAIN, 11));
    lblUsername.setBounds(107, 11, 72, 20);
    contentPane.add(lblUsername);

    final JTextField username = new JTextField();
    username.setBounds(204, 12, 86, 20);
    contentPane.add(username);
    username.setColumns(10);

    JLabel lblNewLabel = new JLabel("Password");
    lblNewLabel.setFont(new Font("Verdana", Font.PLAIN, 11));
    lblNewLabel.setBounds(107, 67, 72, 20);

```

```
contentPane.add(lblNewLabel);
```

```
final JPasswordField passwordField = new JPasswordField();
```

```
passwordField.setFont(new Font("Verdana", Font.PLAIN, 11));
```

```
passwordField.setBounds(205, 68, 85, 20);
```

```
contentPane.add(passwordField);
```

```
JButton btnNewButton = new JButton("Accedi");
```

```
btnNewButton.addActionListener(new ActionListener() {
```

```
@Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        Connection conn;
```

```
        Statement ment;
```

```
        try
```

```
        {
```

```
            Class.forName("com.mysql.jdbc.Driver");
```

```
            String URL = "jdbc:mysql://localhost:3306/dbgestionale";
```

```
String utente= "root";
```

```
String pass = "yes";
```

```
        conn=DriverManager.getConnection(URL,utente,pass);
```

```
        ment = conn.createStatement(); //devo creare uno statement
```

```
        System.out.println("Connessione avvenuta con successo");
```

```
        String query = "SELECT * FROM utenti WHERE Username = " + username.getText()+" AND Password ="+passwordField.getText()+"";
```

```
        ResultSet rs = ment.executeQuery(query); // dopo la richiesta devo metterlo sempre
```

```
        rs.last(); //pure questo
```

```
        int numRows = rs.getRow();
```

```
if(numRows>0) { //se è magg di 0 allora va bene e si può passare al Frame successivo
```

```
        MainGUI gui = new MainGUI(); //ossia creo un oggetto istanza della classe MainGUI
```

```
        gui.setLocationRelativeTo(null); //posiziono la finestra al centro dello schermo
```

```

        dispose(); //disturgo la finestra di login

        gui.setVisible(true); //rendo visibile la finestra
    }

    else

        JOptionPane.showMessageDialog(null, "Credenziali non valide!");

    ment.close();

    conn.close();

}

catch(Exception s){

    System.out.println("Errore: " + s.getMessage());

}

}

});

btnNewButton.setFont(new Font("Verdana", Font.PLAIN, 11));

btnNewButton.setBounds(165, 144, 89, 23);

contentPane.add(btnNewButton);

JButton btnRegistrati = new JButton("Non sei ancora registrato? Registrati!");

btnRegistrati.setBounds(53, 214, 344, 23);

contentPane.add(btnRegistrati);

btnRegistrati.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent arg0) {

        dispose();

        Registrazione gui = new Registrazione();

        gui.setLocationRelativeTo(null);

        gui.setVisible(true);

    }

});

```

```

        btnRegistrati.setBounds(165, 212, 128, 23);

        contentPane.add(btnRegistrati);
    }

}

package view;

import java.awt.Font;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;

public class Registrazione extends JFrame {
    private JPanel contentPane;

    public Registrazione() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        contentPane = new JPanel();
    }
}

```

```

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JLabel lblUsername = new JLabel("Inserisci username");
lblUsername.setFont(new Font("Verdana", Font.PLAIN, 11));
lblUsername.setBounds(53, 11, 126, 20);
contentPane.add(lblUsername);

final JTextField user = new JTextField();
user.setBounds(204, 12, 86, 20);
contentPane.add(user);
user.setColumns(10);

JLabel lblNewLabelPass = new JLabel("Inserisci password");
lblNewLabelPass.setFont(new Font("Verdana", Font.PLAIN, 11));
lblNewLabelPass.setBounds(53, 50, 126, 20);
contentPane.add(lblNewLabelPass);

final JPasswordField passwordField = new JPasswordField();
passwordField.setFont(new Font("Verdana", Font.PLAIN, 11));
passwordField.setBounds(204, 50, 85, 20);
contentPane.add(passwordField);

final JButton btnRegistrami = new JButton("Registrami !");
btnRegistrami.addActionListener(new ActionListener() {
@Override
    public void actionPerformed(ActionEvent e) {
        try {
            Connection conn;
            Statement ment;

```

```

Class.forName("com.mysql.jdbc.Driver");

String URL = "jdbc:mysql://localhost:3306/dbgestionale";

String utente= "root";

String pass = "yes";

conn=DriverManager.getConnection(URL,utente,pass);

ment = conn.createStatement();

System.out.println("Connessione avvenuta con successo");

ment.executeUpdate("INSERT INTO
utenti(username,password) VALUES (""+user.getText()+"','"+passwordField.getPassword()+"")");

JOptionPane.showMessageDialog(null,"Operazione
completata con successo.\n Fare click su ok per accedere al sistema");

ment.close();

conn.close();

dispose();

MainGUI gui = new MainGUI();

gui.setLocationRelativeTo(null);

gui.setVisible(true);

}

catch(ClassNotFoundException | SQLException s){

System.out.println("Errore: " + s.getMessage());

}

}

});

btnRegistrami.setBounds(124, 209, 146, 23);

contentPane.add(btnRegistrami);

}

}

package view;

import java.awt.BorderLayout;

```



```
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JMenuBar;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class MainGUI extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public MainGUI() {

        super("Gestionale in Java"); //titolo del Frame

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 300);

        JMenuBar menuBar = new JMenuBar();
        setJMenuBar(menuBar);

        JMenu mnProdotti = new JMenu("Prodotti");
        menuBar.add(mnProdotti);

        JMenuItem mntmAggiungi = new JMenuItem("Aggiungi");
        mntmAggiungi.addActionListener(new ActionListener() {
```

```
        public void actionPerformed(ActionEvent e) {  
            AddProduct gui = new AddProduct();  
            gui.setLocationRelativeTo(null);  
            gui.setVisible(true);  
        }  
    };  
    mnProdotti.add(mntmAggiungi);  
  
    JMenuItem mntmModifica = new JMenuItem("Modifica");  
    mntmModifica.addActionListener(new ActionListener() {  
        public void actionPerformed(ActionEvent e) {  
            ChangeProduct gui = new ChangeProduct();  
            gui.setLocationRelativeTo(null);  
            gui.pack();  
            gui.setVisible(true);  
        }  
    });  
    mnProdotti.add(mntmModifica);  
  
    JMenuItem mntmRimuovi = new JMenuItem("Rimuovi");  
    mntmRimuovi.addActionListener(new ActionListener() {  
        public void actionPerformed(ActionEvent e) {  
            RemoveProduct gui = new RemoveProduct();  
            gui.setLocationRelativeTo(null);  
            gui.pack();  
            gui.setVisible(true);  
        }  
    });  
    mnProdotti.add(mntmRimuovi);
```

```

JMenuItem mntmEsci = new JMenuItem("Esci");
mntmEsci.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(1);
    }
});
mnProdotti.add(mntmEsci);

JMenu mnScorte = new JMenu("Scorte");
menuBar.add(mnScorte);

JMenuItem mntmVisualizzaGiacenze = new JMenuItem("Visualizza giacenze");
mntmVisualizzaGiacenze.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Inventory gui = new Inventory();
        gui.setLocationRelativeTo(null);
        gui.pack();
        gui.setVisible(true);
    }
});
mnScorte.add(mntmVisualizzaGiacenze);
contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
contentPane.setLayout(new BorderLayout(0, 0));
setContentPane(contentPane);
}
}

```

```
package view;
```

```
import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.*;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JTextField;

import javax.swing.border.EmptyBorder;

public class insertProduct extends JFrame {

    private String radiobutton;

    private JPanel contentPane;

    public insertProduct(final String radiobutton){

        this.radiobutton = radiobutton;

        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        setBounds(100, 100, 450, 300);

        setLocationRelativeTo(null);

        contentPane = new JPanel();

        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        contentPane.setLayout(null);

        setContentPane(contentPane);

        JLabel lblNewLabel = new JLabel("Caratteristiche nuovo prodotto");

        lblNewLabel.setFont(new Font("Verdana", Font.PLAIN, 11));

        lblNewLabel.setBounds(117, 26, 211, 40);

        contentPane.add(lblNewLabel);

        JLabel lblNewLabel_1 = new JLabel("Nome");
```

```

lblNewLabel_1.setFont(new Font("Verdana", Font.PLAIN, 11));

lblNewLabel_1.setBounds(38, 94, 46, 14);

contentPane.add(lblNewLabel_1);

final JTextField textField_1 = new JTextField();

textField_1.setBounds(127, 92, 86, 20);

contentPane.add(textField_1);

textField_1.setColumns(10);

JLabel lblNewLabel_2 = new JLabel("Descrizione");

lblNewLabel_2.setFont(new Font("Verdana", Font.PLAIN, 11));

lblNewLabel_2.setBounds(38, 159, 77, 20);

contentPane.add(lblNewLabel_2);

final JTextField textField_2 = new JTextField();

textField_2.setBounds(127, 160, 86, 20);

contentPane.add(textField_2);

textField_2.setColumns(10);

JButton btnNewButton = new JButton("Aggiungi");

btnNewButton.addActionListener(new ActionListener() {

@Override

    public void actionPerformed(ActionEvent arg0) {

        Connection conn;

        Statement ment;

        try

        {

            Class.forName("com.mysql.jdbc.Driver");

            String URL = "jdbc:mysql://localhost:3306/dbgestionale";

            String utente= "root";

            String pass = "yes";

            conn=DriverManager.getConnection(URL,utente,pass);

            ment = conn.createStatement();

            ment.executeUpdate("INSERT INTO prodotti(nomeprodotto,descrizione,tipologia) VALUES ('"+

textField_1.getText()+ "','"+textField_2.getText()+ "','"+

radiobutton+"')");

```

```

        JOptionPane.showMessageDialog(null,"Connessione avvenuta con successo");
    }

    catch(ClassNotFoundException | SQLException s){
        System.out.println("Errore: " + s.getMessage());
    }
}
});

btnNewButton.setBounds(164, 228, 89, 23);
contentPane.add(btnNewButton);
}
}

```

age view;

```

import java.awt.BorderLayout;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.border.EmptyBorder;

public class AddProduct extends JFrame {

```

```
private JPanel contentPane;

/**
 * Create the frame.
 */
public AddProduct() {

    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    setBounds(100, 100, 450, 300);

    contentPane = new JPanel();

    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

    contentPane.setLayout(null);

    setContentPane(contentPane);

    JLabel lblNewLabel = new JLabel("Che tipo/i di prodotto vuoi aggiungere?");
    lblNewLabel.setFont(new Font("Verdana", Font.PLAIN, 11));
    lblNewLabel.setBounds(89, 27, 265, 29);
    contentPane.add(lblNewLabel);

    final JRadioButton radio1= new JRadioButton("Alimentari");
    radio1.setFont(new Font("Verdana", Font.PLAIN, 11));
    radio1.setBounds(146, 63, 109, 23);
    contentPane.add(radio1);

    final JRadioButton radio2 = new JRadioButton("Cosmetici");
    radio2.setFont(new Font("Verdana", Font.PLAIN, 11));
    radio2.setBounds(146, 95, 109, 23);
```

```
contentPane.add(radio2);
```

```
final JRadioButton radio3 = new JRadioButton("Vestiti");
```

```
radio3.setFont(new Font("Verdana", Font.PLAIN, 11));
```

```
radio3.setBounds(146, 131, 109, 23);
```

```
contentPane.add(radio3);
```

```
final JRadioButton radio4 = new JRadioButton("Sport");
```

```
radio4.setBounds(146, 167, 109, 23);
```

```
contentPane.add(radio4);
```

```
ButtonGroup g1 = new ButtonGroup();
```

```
g1.add(radio1);
```

```
g1.add(radio2);
```

```
g1.add(radio3);
```

```
g1.add(radio4);
```

```
JButton btnNewButton = new JButton("Aggiungi");
```

```
btnNewButton.addActionListener(new ActionListener() {
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        if(radio1.isSelected())
```

```
            new insertProduct(radio1.getText()).setVisible(true);
```

```
        else if(radio2.isSelected())
```

```
            new insertProduct(radio2.getText()).setVisible(true);
```

```
        else if (radio3.isSelected())
```

```
            new insertProduct(radio3.getText()).setVisible(true);
```

```
        else if(radio4.isSelected())
```

```
            new insertProduct(radio4.getText()).setVisible(true);
```

```
        JOptionPane.showMessageDialog(null,"Hai aggiunto con successo il tuo  
prodotto!");
```



```
        }  
    });  
    btnNewButton.setFont(new Font("Verdana", Font.PLAIN, 11));  
    btnNewButton.setBounds(147, 217, 89, 23);  
    contentPane.add(btnNewButton);  
}  
  
}
```

```
package view;
```

```
import java.awt.Font;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import javax.swing.JButton;  
import javax.swing.JComboBox;  
  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JScrollPane;  
import javax.swing.JTextArea;  
import javax.swing.JTextField;
```

```

import javax.swing.border.EmptyBorder;

public class changeProduct_1 extends JFrame {

private String radioButton;

private JPanel contentPane;

private Connection conn = null;

private Statement ment = null;

static String prodottoSelezionato;

static ResultSet rst;

JTextArea descrizione = new JTextArea(40,30);

JScrollPane scrollPane = new JScrollPane(descrizione);

public changeProduct_1 (String radioButton){

    this.radioButton = radioButton;

initComponents();

}

private void initComponents() {

try {

setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

setBounds(100, 100, 450, 366);

contentPane = new JPanel();

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

setLocationRelativeTo(null);

Class.forName("com.mysql.jdbc.Driver");

String URL = "jdbc:mysql://localhost:3306/dbgestionale";

String utente= "root";

String pass = "yes";

conn=DriverManager.getConnection(URL,utente,pass);

```

```

ment = conn.createStatement();

descrizione.setEditable(true);

scrollPane.setBounds(127, 160, 246, 60);

contentPane.add(scrollPane);

rst = ment.executeQuery("SELECT nomeprodotto FROM prodotti WHERE tipologia ="+radioButton+"");

final JComboBox combo = new JComboBox();

while(rst.next())

combo.addItem(rst.getString("nomeprodotto"));

combo.setBounds(127, 92, 86, 20);

contentPane.add(combo);

combo.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try {

java.lang.String prodottoSelezionato = (String) combo.getSelectedItem();

java.sql.ResultSet rst = ment.executeQuery("SELECT descrizione FROM prodotti WHERE

nomeprodotto="+prodottoSelezionato+"");

rst.next();

descrizione.setText(rst.getString("descrizione"));

} catch (SQLException ex) {

Logger.getLogger(changeProduct_1.class.getName()).log(Level.SEVERE, null, ex);

}

}

});

JLabel lblNewLabel = new JLabel("Caratteristiche prodotto da cambiare");

lblNewLabel.setFont(new Font("Verdana", Font.PLAIN, 11));

lblNewLabel.setBounds(96, 25, 240, 40);

contentPane.add(lblNewLabel);

JLabel lblNewLabel_1 = new JLabel("Nome");

lblNewLabel_1.setFont(new Font("Verdana", Font.PLAIN, 11));

lblNewLabel_1.setBounds(38, 94, 46, 14);

```

```

contentPane.add(lblNewLabel_1);

JLabel lblNewLabel_2 = new JLabel("Descrizione");
lblNewLabel_2.setFont(new Font("Verdana", Font.PLAIN, 11));
lblNewLabel_2.setBounds(38, 159, 77, 20);
contentPane.add(lblNewLabel_2);

JButton btnNewButton = new JButton("Cambia");
btnNewButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent arg0) {

try {

java.lang.String prodottoSelezionato = null;

ment.executeUpdate("UPDATE prodotti SET descrizione='"+ descrizione.getText()+" WHERE
nomeprodotto='"+prodottoSelezionato+"'");

JOptionPane.showMessageDialog(null,"Modifica avvenuta con successo");

} catch (SQLException ex) {

Logger.getLogger(changeProduct_1.class.getName()).log(Level.SEVERE, null, ex);

}

}

});

btnNewButton.setBounds(164, 268, 89, 23);

contentPane.add(btnNewButton);

pack();

} catch (ClassNotFoundException | SQLException ex) {

Logger.getLogger(changeProduct_1.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

package view;

```

```
import java.awt.BorderLayout;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ButtonGroup;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JRadioButton;

import javax.swing.border.EmptyBorder;

public class ChangeProduct extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public ChangeProduct() {

        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        setBounds(100, 100, 450, 300);

        contentPane = new JPanel();

        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
```

```
contentPane.setLayout(null);  
setContentPane(contentPane);
```

```
JLabel lblNewLabel = new JLabel("Che tipo/i di prodotto desideri adesso?");  
lblNewLabel.setFont(new Font("Verdana", Font.PLAIN, 11));  
lblNewLabel.setBounds(89, 27, 265, 29);  
contentPane.add(lblNewLabel);
```

```
final JRadioButton radio1 = new JRadioButton("Alimentari");  
radio1.setFont(new Font("Verdana", Font.PLAIN, 11));  
radio1.setBounds(146, 63, 109, 23);  
contentPane.add(radio1);
```

```
final JRadioButton radio2 = new JRadioButton("Cosmetici");  
radio2.setFont(new Font("Verdana", Font.PLAIN, 11));  
radio2.setBounds(146, 95, 109, 23);  
contentPane.add(radio2);
```

```
final JRadioButton radio3 = new JRadioButton("Vestiti");  
radio3.setFont(new Font("Verdana", Font.PLAIN, 11));  
radio3.setBounds(146, 131, 109, 23);  
contentPane.add(radio3);
```

```
final JRadioButton radio4 = new JRadioButton("Sport");  
radio4.setBounds(146, 167, 109, 23);  
contentPane.add(radio4);
```

```
ButtonGroup g1 = new ButtonGroup();  
g1.add(radio1);
```

```
g1.add(radio2);  
g1.add(radio3);  
g1.add(radio4);
```

```
    JButton btnNewButton = new JButton("Cambia");  
    btnNewButton.addActionListener(new ActionListener() {  
        public void actionPerformed(ActionEvent e) {
```

```
            if(radio1.isSelected())  
                new changeProduct_1(radio1.getText()).setVisible(true);  
  
            else if(radio2.isSelected())  
                new changeProduct_1(radio2.getText()).setVisible(true);  
            else if(radio3.isSelected())  
                new changeProduct_1(radio3.getText()).setVisible(true);  
            else if (radio4.isSelected())  
                new changeProduct_1(radio4.getText()).setVisible(true);
```

```
                JOptionPane.showMessageDialog(null, "Hai modificato il tuo prodotto con  
successo!");
```

```
        }
```

```
    });
```

```
    btnNewButton.setFont(new Font("Verdana", Font.PLAIN, 11));
```

```
    btnNewButton.setBounds(147, 217, 89, 23);
```

```
    contentPane.add(btnNewButton);
```

```
}
```

```
}
```

```
package view;
```

```
import java.awt.Font;
```

```
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.SQLException;
```

```
import java.sql.Statement;
```

```
import javax.swing.JButton;
```

```
import javax.swing.JFrame;
```

```
import javax.swing.JLabel;
```

```
import javax.swing.JOptionPane;
```

```
import javax.swing.JPanel;
```

```
import javax.swing.JTextField;
```

```
import javax.swing.border.EmptyBorder;
```

```
public class removeProduct_1 extends JFrame {
```

```
    private JPanel contentPane = new JPanel();
```

```
    private JTextField textField_3 = new JTextField();
```

```
    private String button;
```

```
    public removeProduct_1 (String button){
```

```
        this.button = button;
```

```
        initComponents();
```

```
    }
```

```
    private void initComponents() {
```

```
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
```

```
        setBounds(100, 100, 450, 300);
```



```

        contentPane = new JPanel();

        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        contentPane.setLayout(null);

        setContentPane(contentPane);

        JLabel lblNewLabel = new JLabel("Prodotto da rimuovere");

        lblNewLabel.setFont(new Font("Verdana", Font.PLAIN, 13));

        lblNewLabel.setBounds(123, 32, 163, 46);

        contentPane.add(lblNewLabel);

        JLabel lblNewLabel_1 = new JLabel("Nome");

        lblNewLabel_1.setFont(new Font("Tahoma", Font.PLAIN, 12));

        lblNewLabel_1.setBounds(123, 89, 74, 26);

        contentPane.add(lblNewLabel_1);

        textField_3.setBounds(185, 93, 86, 20);

        contentPane.add(textField_3);

        textField_3.setColumns(10);

        JButton btnNewButton = new JButton("Rimuovi");

        btnNewButton.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent arg0) {

                Connection conn;

                Statement ment;

                try

                {

                    Class.forName("com.mysql.jdbc.Driver");

                    String URL = "jdbc:mysql://localhost:3306/dbgestionale";

```

```

        String utente= "root";

        String pass = "yes";

        conn=DriverManager.getConnection(URL,utente,pass);

        ment = conn.createStatement();

        System.out.println("Connessione avvenuta con successo");

        ment.executeUpdate("DELETE FROM prodotti WHERE
nomeprodotto="+textField_3.getText()+"");

        JOptionPane.showMessageDialog(null,"Operazione completata");
    }

    catch(ClassNotFoundException | SQLException s){

        System.out.println("Errore: " + s.getMessage());

    }

}

});

btnNewButton.setBounds(150, 179, 89, 23);

contentPane.add(btnNewButton);

}

}

```

```
package view;
```

```

import java.awt.BorderLayout;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.ButtonGroup;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

```

```
import javax.swing.JPanel;

import javax.swing.JRadioButton;

import javax.swing.border.EmptyBorder;

public class RemoveProduct extends JFrame {

    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public RemoveProduct() {

        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 450, 300);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        contentPane.setLayout(null);
        setContentPane(contentPane);

        JLabel lblNewLabel = new JLabel("Che tipo/i di prodotto desideri rimuovere?");
        lblNewLabel.setFont(new Font("Verdana", Font.PLAIN, 11));
        lblNewLabel.setBounds(89, 27, 265, 29);
        contentPane.add(lblNewLabel);

        final JRadioButton radio1 = new JRadioButton("Alimentari");
        radio1.setFont(new Font("Verdana", Font.PLAIN, 11));
```

```
radio1.setBounds(146, 63, 109, 23);
```

```
contentPane.add(radio1);
```

```
final JRadioButton radio2 = new JRadioButton("Cosmetici");
```

```
radio2.setFont(new Font("Verdana", Font.PLAIN, 11));
```

```
radio2.setBounds(146, 95, 109, 23);
```

```
contentPane.add(radio2);
```

```
final JRadioButton radio3 = new JRadioButton("Vestiti");
```

```
radio3.setFont(new Font("Verdana", Font.PLAIN, 11));
```

```
radio3.setBounds(146, 131, 109, 23);
```

```
contentPane.add(radio3);
```

```
final JRadioButton radio4 = new JRadioButton("Sport");
```

```
radio4.setBounds(146, 167, 109, 23);
```

```
contentPane.add(radio4);
```

```
ButtonGroup g1 = new ButtonGroup();
```

```
g1.add(radio1);
```

```
g1.add(radio2);
```

```
g1.add(radio3);
```

```
g1.add(radio4);
```

```
JButton btnNewButton = new JButton("Rimuovi");
```

```
btnNewButton.addActionListener(new ActionListener() {
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        if(radio1.isSelected())
```

```
            new removeProduct_1(radio1.getText()).setVisible(true);
```

```
        else if(radio2.isSelected())
```

```
            new removeProduct_1(radio2.getText()).setVisible(true);
```

```

        else if (radio3.isSelected())
            new removeProduct_1 (radio3.getText()).setVisible(true);
        else if (radio4.isSelected())
            new removeProduct_1 (radio4.getText()).setVisible(true);

        JOptionPane.showMessageDialog(null, "Hai rimosso il tuo prodotto con
successo!");

    }

});

btnNewButton.setFont(new Font("Verdana", Font.PLAIN, 11));

btnNewButton.setBounds(147, 217, 89, 23);

contentPane.add(btnNewButton);

}

}

package view;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.swing.JFrame;
import javax.swing.JPanel;

```

```
import javax.swing.JScrollPane;

import javax.swing.JTabbedPane;

import javax.swing.JTextArea;

import javax.swing.border.EmptyBorder;

public class Inventory extends JFrame {

    private JPanel contentPane;

    private Connection conn;

    private Statement stmt;

    private JTabbedPane tabbed = new JTabbedPane();

    /**
     * Create the frame.
     */
    public Inventory() {

        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        setBounds(100, 100, 450, 300);

        contentPane = new JPanel();

        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        contentPane.setLayout(null);

        setContentPane(contentPane);

        contentPane.setLayout(null);

        initComponents();

        tabbed.setBounds(10, 10, 430, 280);

        contentPane.add(tabbed);

    }
}
```

```

private void initComponents() {
    try {
        Class.forName("com.mysql.jdbc.Driver");

        String URL = "jdbc:mysql://localhost:3306/dbgestionale";
        String utente= "root";
        String pass = "root";

        conn = DriverManager.getConnection(URL,utente,pass);
        stmt=(Statement)conn.createStatement();

        ResultSet result = stmt.executeQuery("SELECT * FROM prodotti order by tipologia");

        String categoriaCorrente="";

        JTextArea elencoProdotti=null;
        JScrollPane scrollPane;

        while(result.next()) {

            if(!categoriaCorrente.equals(result.getString("tipologia"))) {

                // aggiungo una nuova scheda al tabbedpane
                elencoProdotti = new JTextArea(40,30);
                scrollPane = new JScrollPane(elencoProdotti);

                JPanel panel = new JPanel();
                panel.add(scrollPane);

                categoriaCorrente = result.getString("tipologia");
                tabbed.addTab(categoriaCorrente, panel);

                System.out.println(categoriaCorrente);
            }

            elencoProdotti.setText(elencoProdotti.getText()+result.getString("nomeprodotto")+":"
                +result.getString("descrizione)+"\n");
        }

    } catch ( ClassNotFoundException | SQLException ex) {

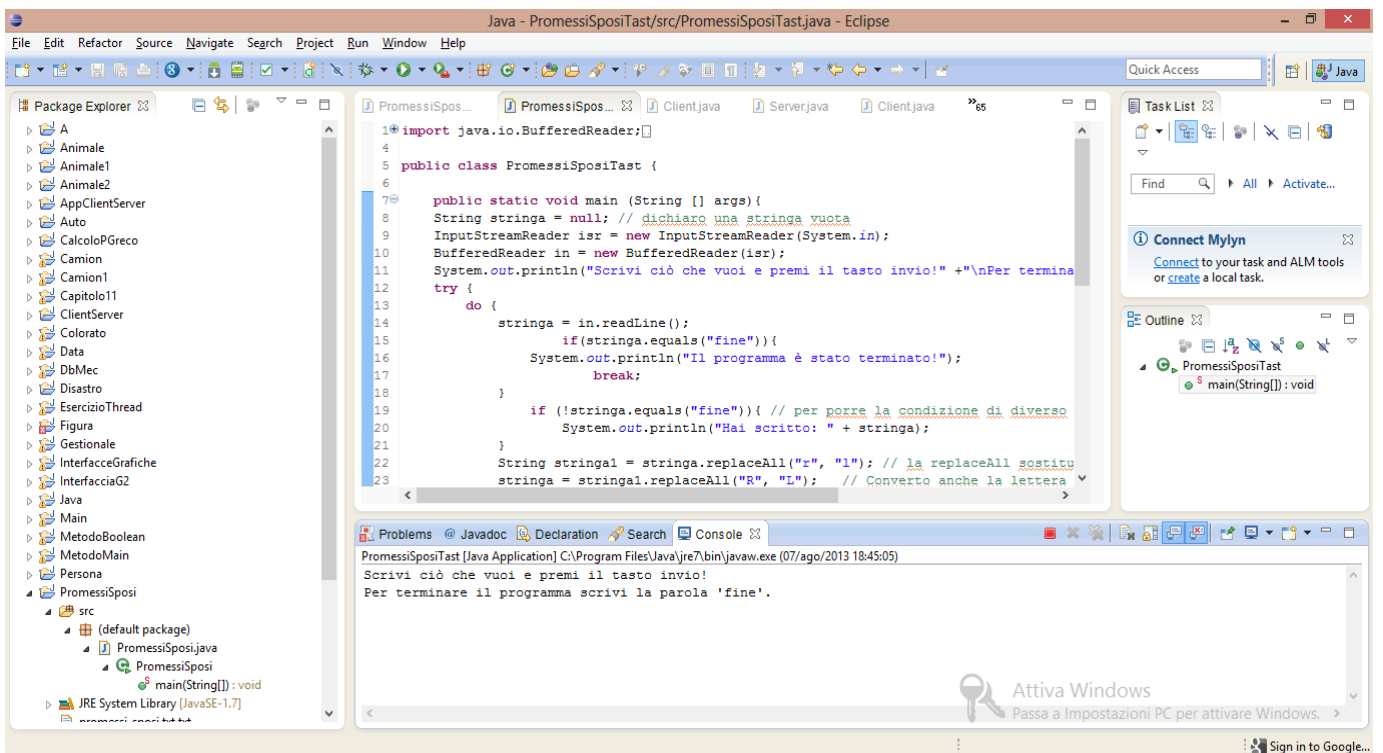
        Logger.getLogger(Inventory.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

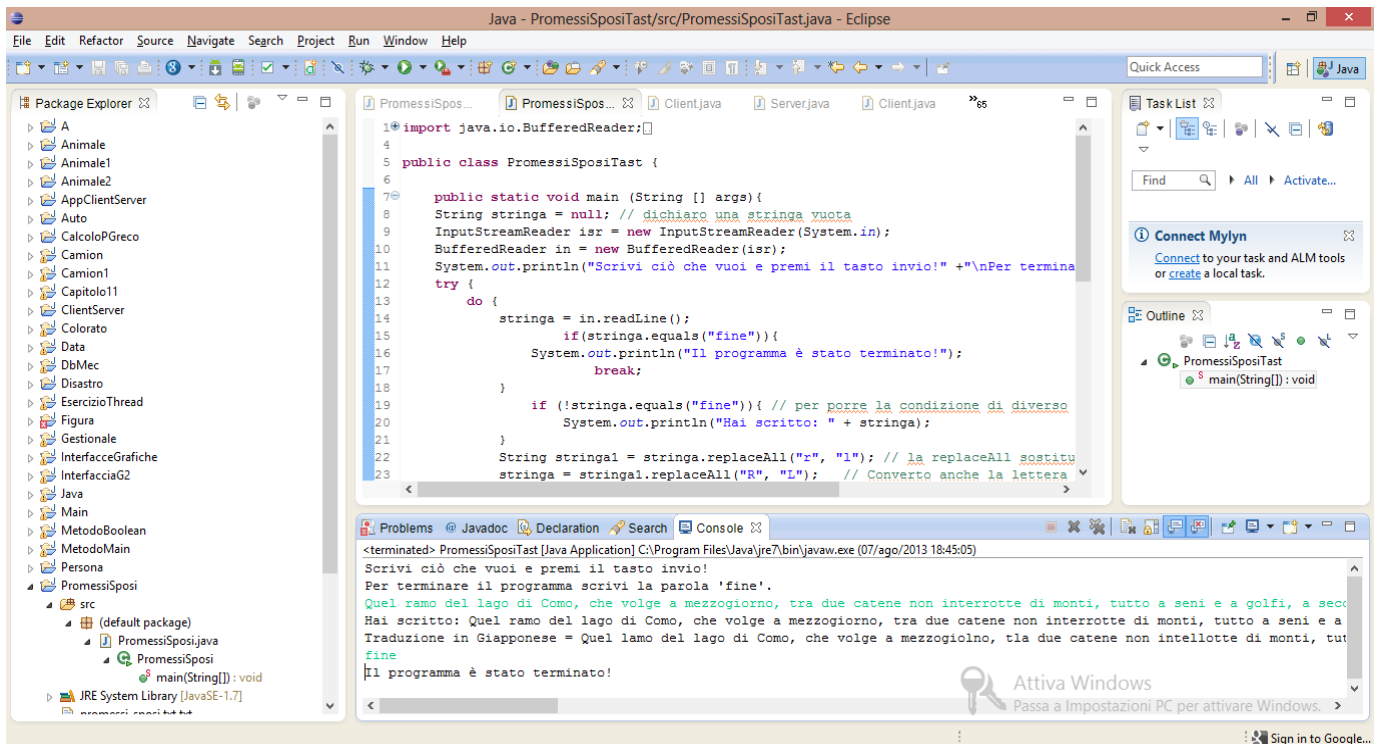
```
}  
  
}
```

## Applicazione 'traduzione' Promessi Sposi Italiano-Giapponese

Scrivere una classe java in grado di "tradurre" un testo in giapponese utilizzando un input da tastiera:







Codice sorgente:

```
import java.io.BufferedReader;
```

```
import java.io.IOException;
```

```
import java.io.InputStreamReader;
```

```
public class PromessiSposiTast {
```

```
    public static void main (String [] args){
```

```
        String stringa = null; // dichiaro una stringa vuota
```

```
        InputStreamReader isr = new InputStreamReader(System.in);
```

```
        BufferedReader in = new BufferedReader(isr);
```

```
        System.out.println("Scrivi ciò che vuoi e premi il tasto invio!" + "\nPer terminare il programma scrivi la parola 'fine'.");
```

```
        try {
```

```
            do {
```

```
                stringa = in.readLine();
```

```
                if (stringa.equals("fine")){
```

```
                    System.out.println("Il programma è stato terminato!");
```

```

        break;
    }

    if (!stringa.equals("fine")){ // per porre la condizione di diverso su una stringa si
usa il ! equals

        System.out.println("Hai scritto: " + stringa);

    }

    String stringa1 = stringa.replaceAll("r", "l"); // la replaceAll sostituisce ogni occorrenza
della r con la l

    stringa = stringa1.replaceAll("R", "L"); // Converto anche la lettera R maiuscola .

    System.out.println("Traduzione in Giapponese = " +stringa); //dopo la modifica
}while(true);
}

/*do {
    stringa = in.readLine();

    if(stringa.equals("fine")){

        System.out.println("Il programma è stato terminato!");

        break;
    }

    if (!stringa.equals("fine")){ // per porre la condizione di diverso su una stringa si
usa il ! equals

        System.out.println("Hai scritto: " + stringa);

    }

    String stringa1 = stringa.replaceAll("R", "L"); // la replaceAll sostituisce ogni occorrenza
della R con la L

    System.out.println(stringa1); //dopo la modifica
}while(true);
}

```

```
*/ catch(IOException e){  
  
}  
  
}  
  
}
```

### Applicazione Client/Server

Si vuole creare un'applicazione **client/server** in grado di calcolare la somma di una serie di numeri inviati in input.

Il programma sarà composto da due classi:

**Client:** invia al server una serie di numeri interi. Questi numeri verranno inviati come righe di testo ed ognuna di esse conterrà un numero intero. L'ultima riga inviata al server (per indicare la fine della sequenza) dovrà essere lo 0;

**Server:** effettua la somma dei numeri ricevuti e la ritrasmette al client utilizzando una riga contenente il valore 0 per indicare la fine della sequenza.

Java - AppClientServer/src/Server.java - Eclipse

File Edit Refactor Source Navigate Search Project Run Window Help

Quick Access

Package Explorer

- A
- Animale
- Animale1
- Animale2
- AppClientServer
  - src
    - (default package)
      - Client.java
      - Server.java
  - JRE System Library [JavaSE-1.7]
  - Auto
  - CalcoloPGreco
  - Camion
  - Camion1
  - Capitolo11
  - ClientServer
  - Colorato
  - Data
  - DbMec
  - Disastro
  - EsercizioThread
  - Figura
  - Gestionale
  - InterfacceGrafiche
  - InterfacciaG2
  - Java
  - Main
  - MetodoBoolean
  - MetodoMain
  - Persona
  - PromessiSposi
  - PromessiSposiTast

```
23 }
24 catch (IOException ex)
25 {
26     ex.printStackTrace();
27     return false;
28 }
29 System.out.println("Server creato con successo!");
30 return true;
31 }
32
33 public void runServer()
34 {
35     while (true)
36     {
37         try
38         {
39             // Il server resta in attesa di una richiesta
40             System.out.println("Server in attesa di richieste..");
41             Socket s1 = server.accept();
42             System.out.println("Client connesso.");
43         }
44     }
45 }
```

Task List

Find

Connect Mylyn

Outline

- Server
  - port: int
  - server: ServerSocket
  - Server(int)
  - startServer(): boolean
  - runServer(): void
  - main(String[]): void

Problems Javadoc Declaration Search Console

Server (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (07/ago/2013 17:52:26)

Server creato con successo!  
Server in attesa di richieste..

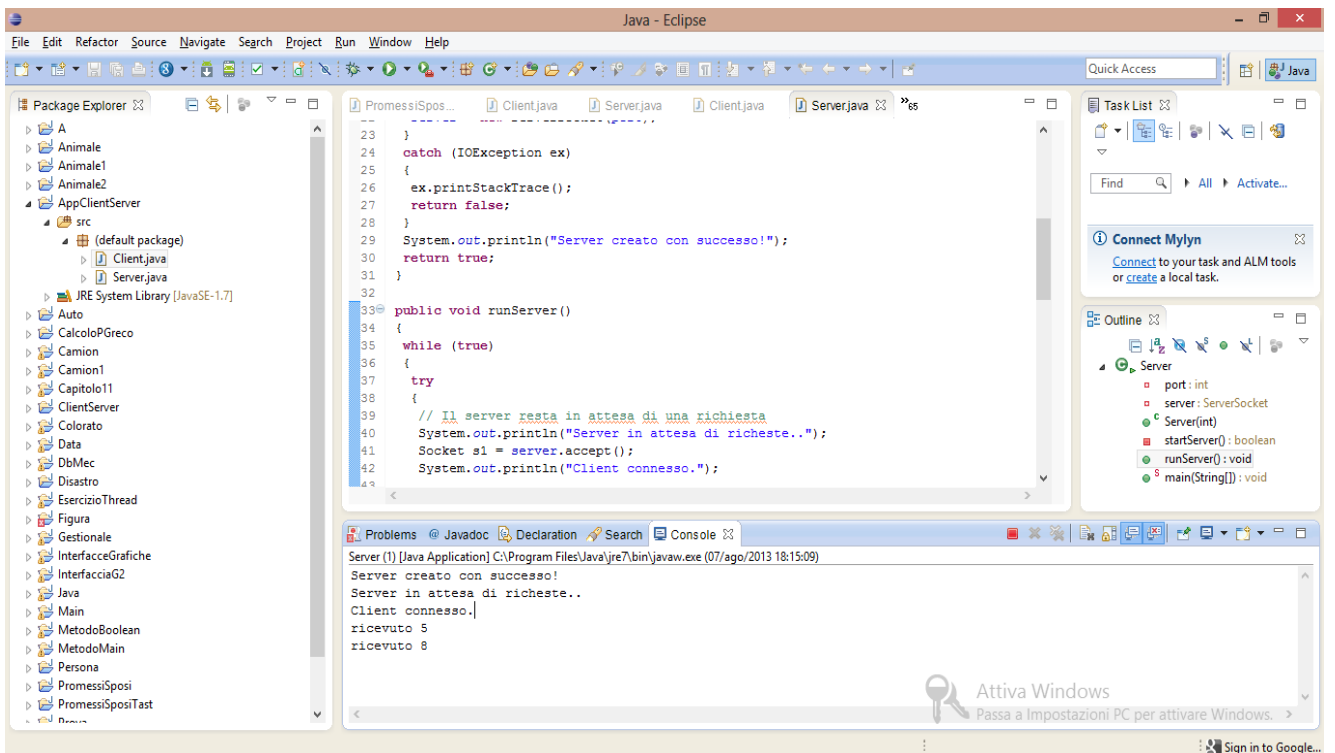
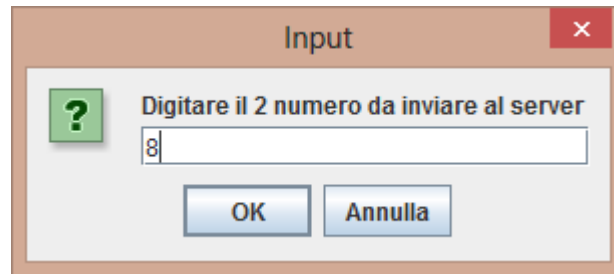
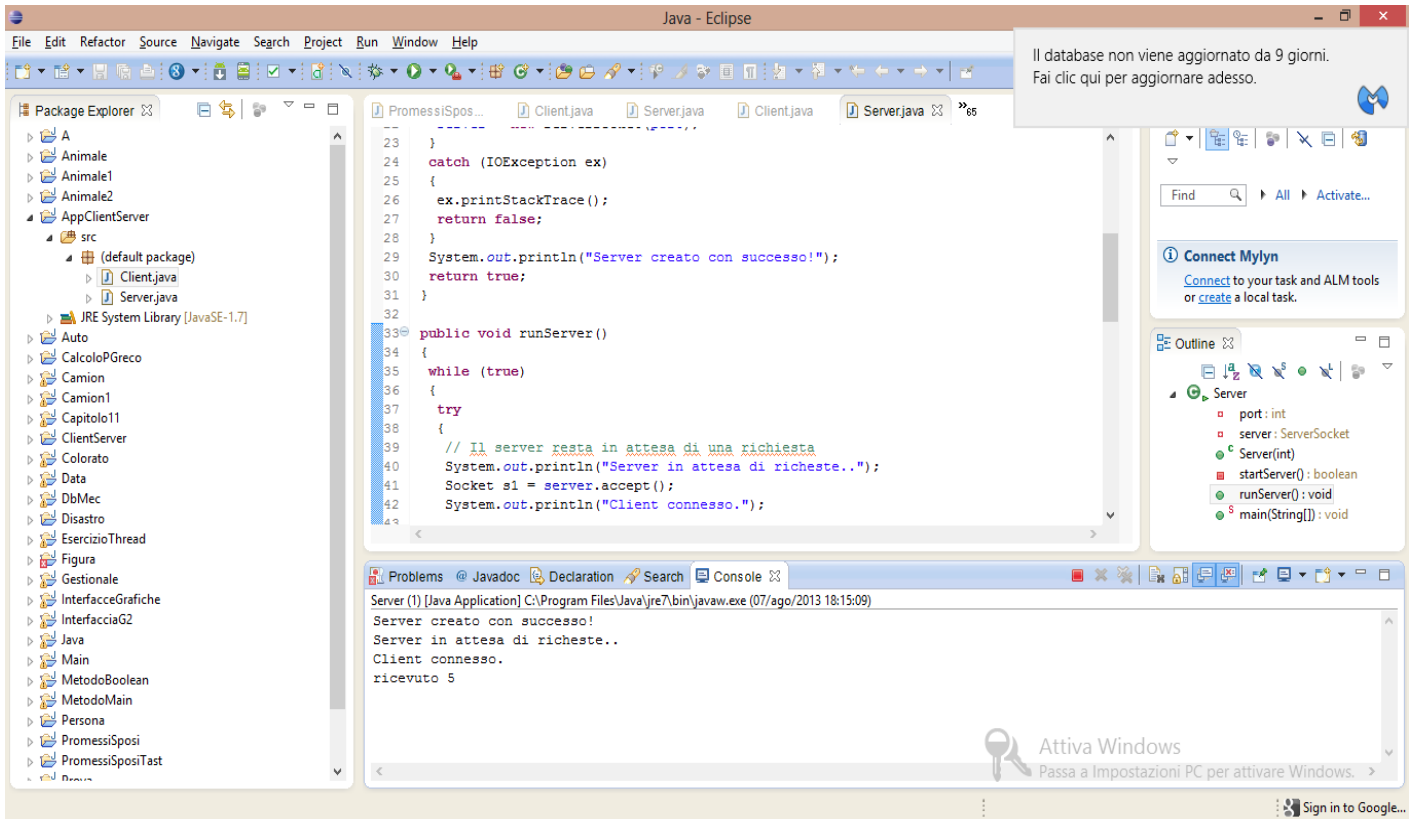
Attiva Windows  
Passa a Impostazioni PC per attivare Windows.

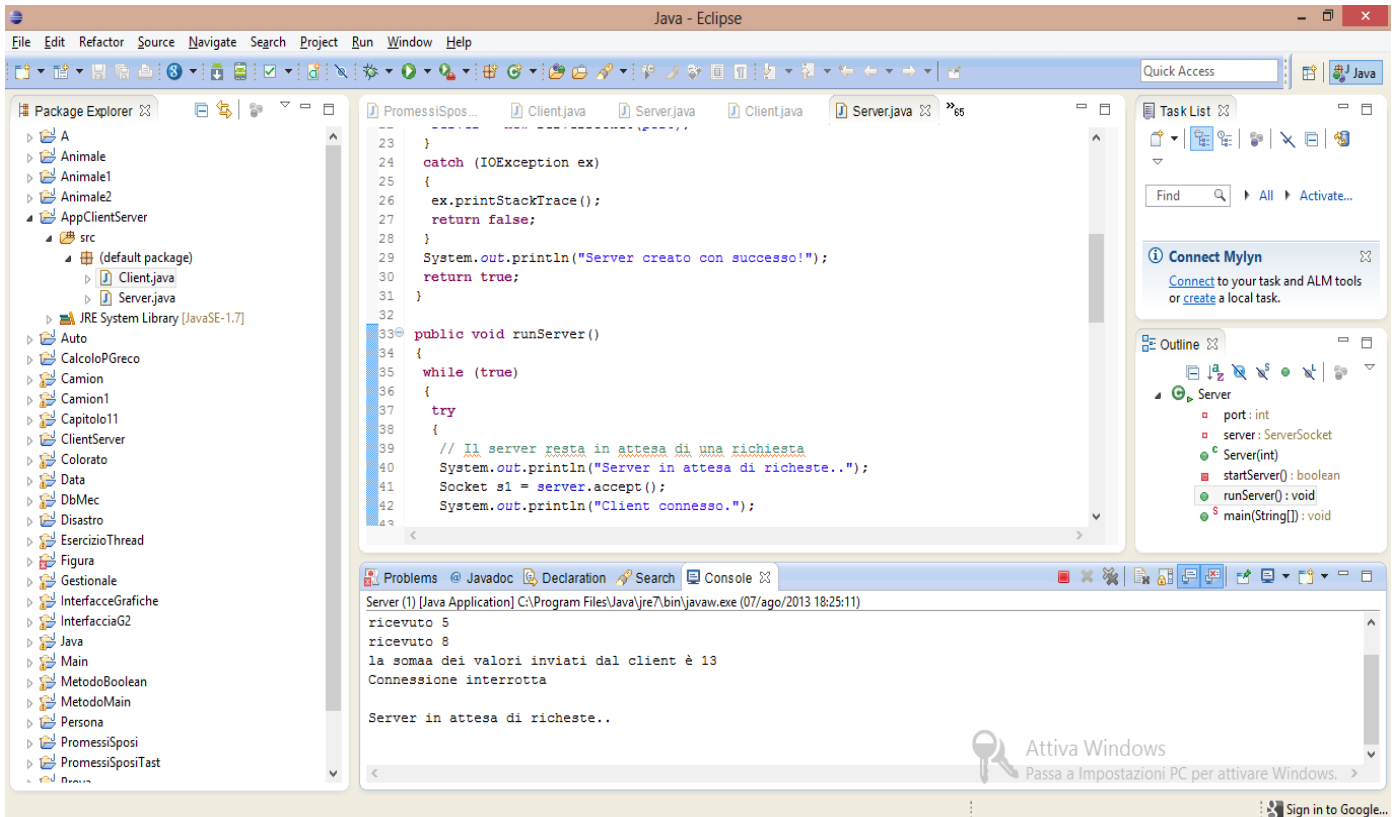
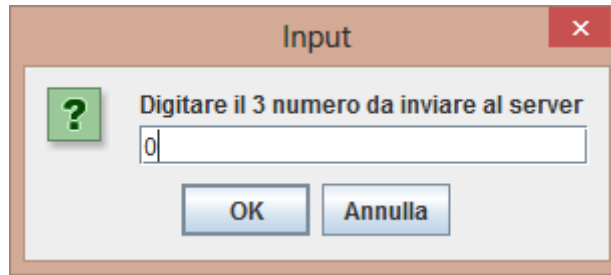
Writable Smart Insert 64:12 Sign in to Google...

Input

Digitare il 1 numero da inviare al server

OK Annulla





Codice sorgente:

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

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

```
public class Server
```

```
{
```

```
    private int port;
```

```
    private ServerSocket server;
```

```
    public Server(int port)
```

```
{
```

```
this.port = port;

if(!startServer()){

    System.err.println("Errore durante la creazione del Server");

}

}
```

```
private boolean startServer()

{

try

{

    server = new ServerSocket(port);

}

catch (IOException ex)

{

    ex.printStackTrace();

    return false;

}

System.out.println("Server creato con successo!");

return true;

}
```

```
public void runServer()

{

while (true)

{

try

{

    // Il server resta in attesa di una richiesta

    System.out.println("Server in attesa di richieste..");

    Socket s1 = server.accept();

}
```

```

System.out.println("Client connesso.");

// Ricava lo stream di output associate al socket
// e definisce una classe wrapper di tipo
// BufferedWriter per semplificare le operazioni
// di scrittura
OutputStream s1out = s1.getOutputStream();
BufferedWriter bw = new BufferedWriter(
    new OutputStreamWriter(s1out));

// Il server invia la risposta al client
bw.write("Benvenuto sul server\n");
BufferedReader br;

    br = new BufferedReader(new InputStreamReader(s1.getInputStream()));

    Integer v;
    int somma = 0;
    while((v=Integer.parseInt(br.readLine()))!=0) {
        System.out.println("ricevuto "+v);
        somma+=v;
    }

System.out.println("la somma dei valori inviati dal client è "+somma);

// Chiude lo stream di output e la connessione
bw.close();
s1.close();
System.out.println("Connessione interrotta\n");
}
catch (IOException ex)
{
    ex.printStackTrace();
}

```



```
}  
}  
}
```

```
public static void main (String args[])  
{  
    // Crea un oggetto di tipo Server in ascolto  
    // sulla porta 5555  
    Server ss = new Server(5555);  
    ss.runServer();  
}  
}
```

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

```
import javax.swing.JOptionPane;
```

```
public class Client  
{  
    public static void main(String args[])  
    {  
        try  
        {  
            // Apre una connessione verso un server in ascolto  
            // sulla porta 5555. In questo caso utilizzo localhost  
            // che corrisponde all'indirizzo IP 127.0.0.1  
            System.out.println("Apertura connessione in corso..");  
            Socket s1 = new Socket ("localhost", 5555);  
            PrintStream bw = new PrintStream(s1.getOutputStream(), true);
```

```

int i = 1;

// Ricava lo stream di input dal socket s1
while(true){

String msg = JOptionPane.showInputDialog("Digitare il " + i++ + " numero da inviare al server");

int msgInt = Integer.parseInt(msg);

bw.println(msg);

    if (msgInt==0)

        break;

// Legge l'input e lo visualizza sullo schermo

// System.out.println("Messaggio n " + i++ + ":" + msg/*+ dis.readLine()*!);

// Al termine, chiude lo stream di comunicazione e il socket.

}

// prima di chiudere la socket se si ha bisogno di leggere dati dal server è opportuno che si ottenga uno
stream di input

// e si legga quanto desiderato

s1.close(); // nella versione originale questa close si trovava all'interno del while in cui si leggono i valori e si
spedivano al server solo che chiudendo la socket dopo il primo invio alla seconda iterazione si genera
un'eccezione

// per risolvere questo problema in modo alternativo a questa soluzione si dovrebbe ottenere un
nuovo oggetto socket ad ogni iterazione del ciclo while dopo di che sempre all'interno dello stesso while si
dovrebbe

// ottenere un nuovo stream di output, questo approccio non è però percorribile a casua dell'alto
costo sia computazionale sia in termini di dispendio delle risorse

System.out.println("Chiusura connessione effettuata");

}

catch (ConnectException connExc)

{

System.err.println("Errore nella connessione."+connExc.getLocalizedMessage());

}

catch (IOException ex)

{

ex.printStackTrace();

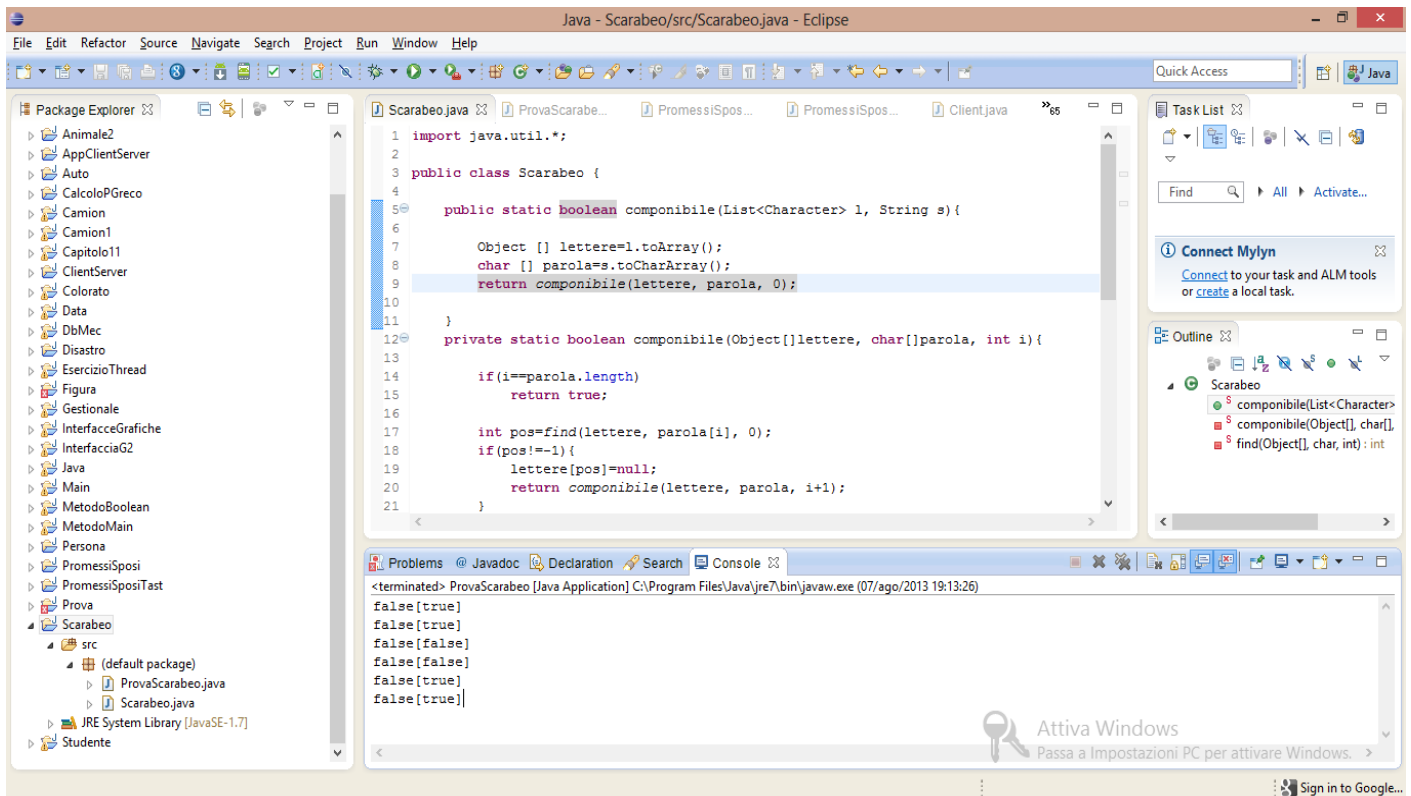
```

```
}  
}  
}
```

### **Applicazione gioco 'Scarabeo'**

Scrivere un metodo statico booleano componibile(List<Character> l, String s) che, data una lista di caratteri alfabetici (che contempla anche il carattere jolly '\*') ed una stringa, restituisce true se la stringa può essere composta usando i caratteri della lista, e false altrimenti. Scrivere la soluzione in modo da poter compilare ed eseguire il seguente programma di prova senza doverlo modificare:

```
import java.util.*;  
class ProvaScarabeo {  
public static void main(String[] args) {  
List<Character> l1 = new ArrayList<Character>();  
l1.add('B');  
l1.add('L');  
l1.add('O');  
l1.add('B');  
l1.add('*');  
l1.add('A');  
l1.add('M'); // B L O B * A M  
List<Character> l2 = new ArrayList<Character>();  
l2.add('*');  
l2.add('F');  
l2.add('*');  
l2.add('A');  
l2.add('A');  
l2.add('L');  
l2.add('F');  
l2.add('L'); // * F * A A L F L  
System.out.println(Scarabeo.componibile(l1,"bambo la") + "[true]");  
System.out.println(Scarabeo.componibile(l1,"bolla ") + "[true]");  
System.out.println(Scarabeo.componibile(l1,"ambra ") + "[false]");  
System.out.println(Scarabeo.componibile(l2,"affan no") + "[false]");  
System.out.println(Scarabeo.componibile(l2,"farfa lla") + "[true]");  
System.out.println(Scarabeo.componibile(l2,"affar e") + "[true]");  
}  
}
```



Codice sorgente:

```

import java.util.*;

public class Scarabeo {

    public static boolean componibile(List<Character> l, String s){

        Object [] lettere=l.toArray();
        char [] parola=s.toCharArray();
        return componibile (lettere, parola, 0);

    }

    private static boolean componibile(Object[]lettere, char[]parola, int i){

        if(i==parola.length)
            return true;

        int pos=find(lettere, parola[i], 0);
        if(pos!=-1){
            lettere[pos]=null;
            return componibile (lettere, parola, i+1);
        }
        else {
            pos=find(lettere, '*', 0);
            if(pos!=-1){
                lettere[pos]=null;
                return componibile (lettere, parola, i+1);
            }
            else return false;
        }
    }

}

```

```

private static int find(Object[] lettere, char c, int i){
    if(i==lettere.length)
        return -i-1;

    if( (new Character(c)).equals((Character)lettere[i] ))
        return 0;

    else return 1+find(lettere, c, i+1);
}
}

```

```

import java.util.ArrayList;
import java.util.List;
class ProvaScarabeo {
    public static void main(String[] args)
    {
        List<Character> l1 = new ArrayList<Character>();
        l1.add('B');
        l1.add('L');
        l1.add('O');
        l1.add('B');
        l1.add('*');
        l1.add('A');
        l1.add('M');
        // B L O B * A M
        List<Character> l2 = new ArrayList<Character>();
        l2.add('*');
        l2.add('F');
        l2.add('*');
        l2.add('A');
        l2.add('A');
        l2.add('L');
        l2.add('F');
        l2.add('L');
        // * F * A A L F L
        System.out.println(Scarabeo.componibile(l1,"bambo la") + "[true]");
        System.out.println(Scarabeo.componibile(l1,"bolla ") + "[true]");
        System.out.println(Scarabeo.componibile(l1,"ambra ") + "[false]");
        System.out.println(Scarabeo.componibile(l2,"affan no") + "[false]");
        System.out.println(Scarabeo.componibile(l2,"farfa lla") + "[true]");
        System.out.println(Scarabeo.componibile(l2,"affar e") + "[true]");
    }
}

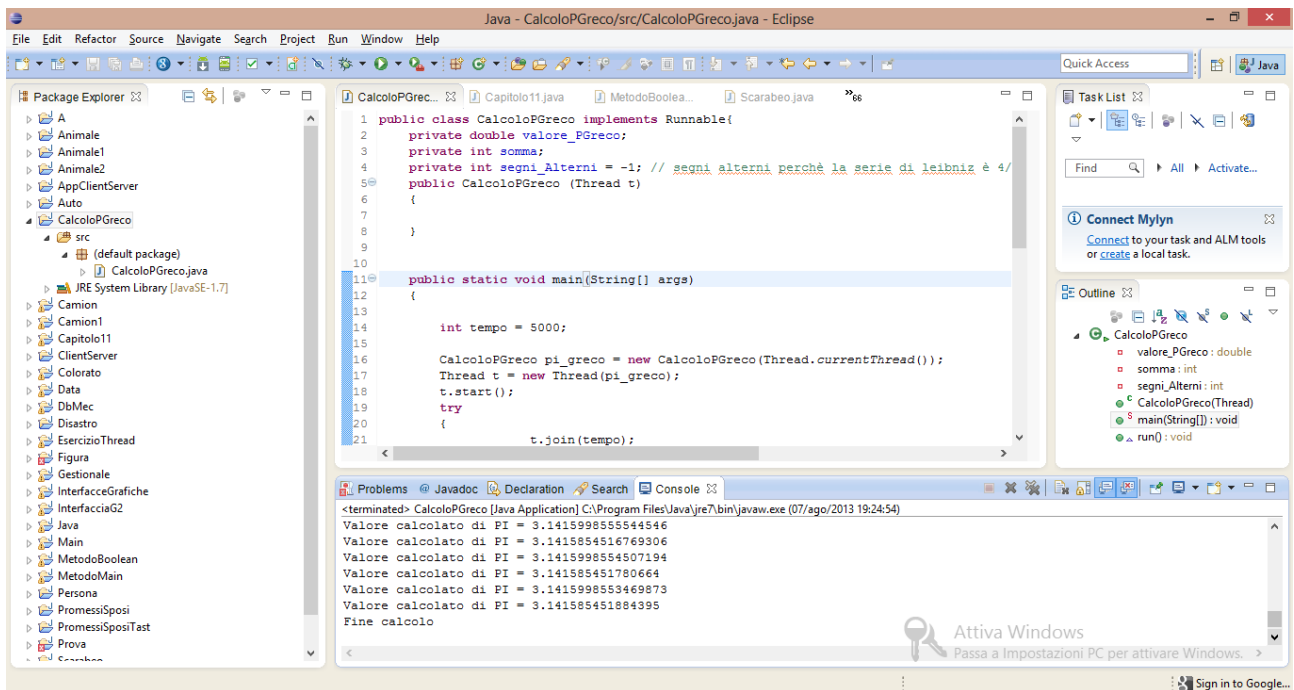
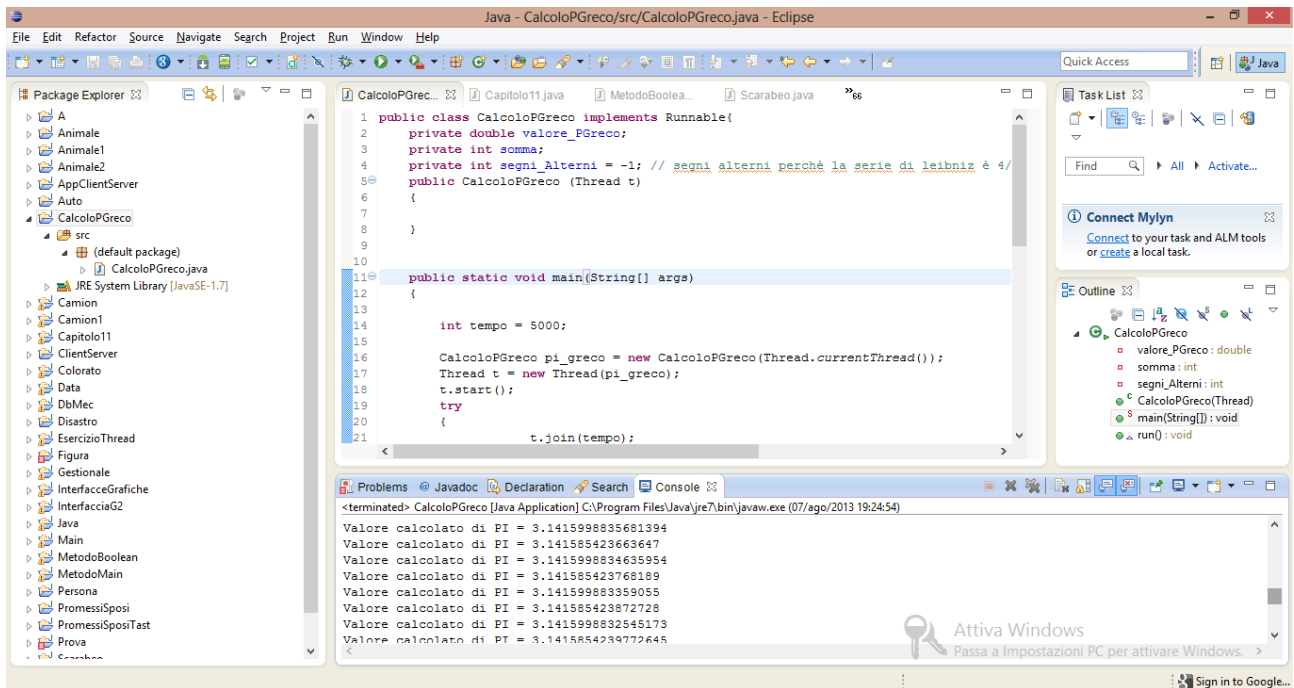
```

## Applicazione Calcolo Pgreco

Scrivere un programma che attiva un thread T che effettua il calcolo approssimato di pigreco. Il programma principale riceve in input un unico argomento che rappresenta il tempo massimo di attesa dopo cui il programma principale interrompe il thread T.

Il thread T effettua un ciclo infinito per il calcolo di pigreco usando la serie di Gregory-Leibniz (pigreco =  $4/1 - 4/3 + 4/5 - 4/7 + 4/9 - 4/11 \dots$ ).

Il thread esce dal ciclo dopo x millisecondi (tempo passato come parametro).



Codice sorgente:

```
public class CalcoloPGreco implements Runnable{
    private double valore_PGreco;
    private int somma;
    private int segni_Alterni = -1; // segni alterni perchè la serie di leibniz è 4/1 -4/3 + 4/5 - 4/7 ecc ecc
    (una volta +, una volta -)
    public CalcoloPGreco (Thread t)
    {
```

```

    }

    public static void main(String[] args)
    {

        int tempo = 5000;

        CalcoloPGreco pi_greco = new CalcoloPGreco(Thread.currentThread());
        Thread t = new Thread(pi_greco);
        t.start();
        try
        {
            t.join(tempo);
            if(t.isAlive()) t.interrupt();
        }
        catch (InterruptedException x)
        {
            System.out.println("Calcolo interrotto");
        }
        System.out.println("Fine calcolo");
    }
    public void run()
    {
        while(!Thread.currentThread().isInterrupted())
        {
            somma++;
            segni_Alterni = -segni_Alterni;
            valore_PGreco += segni_Alterni * 4.0 / ((2 * somma) - 1);
            System.out.println("Valore calcolato di PI = " + valore_PGreco);
        }
    }
}

```

## Applicazioni prenotazione viaggio

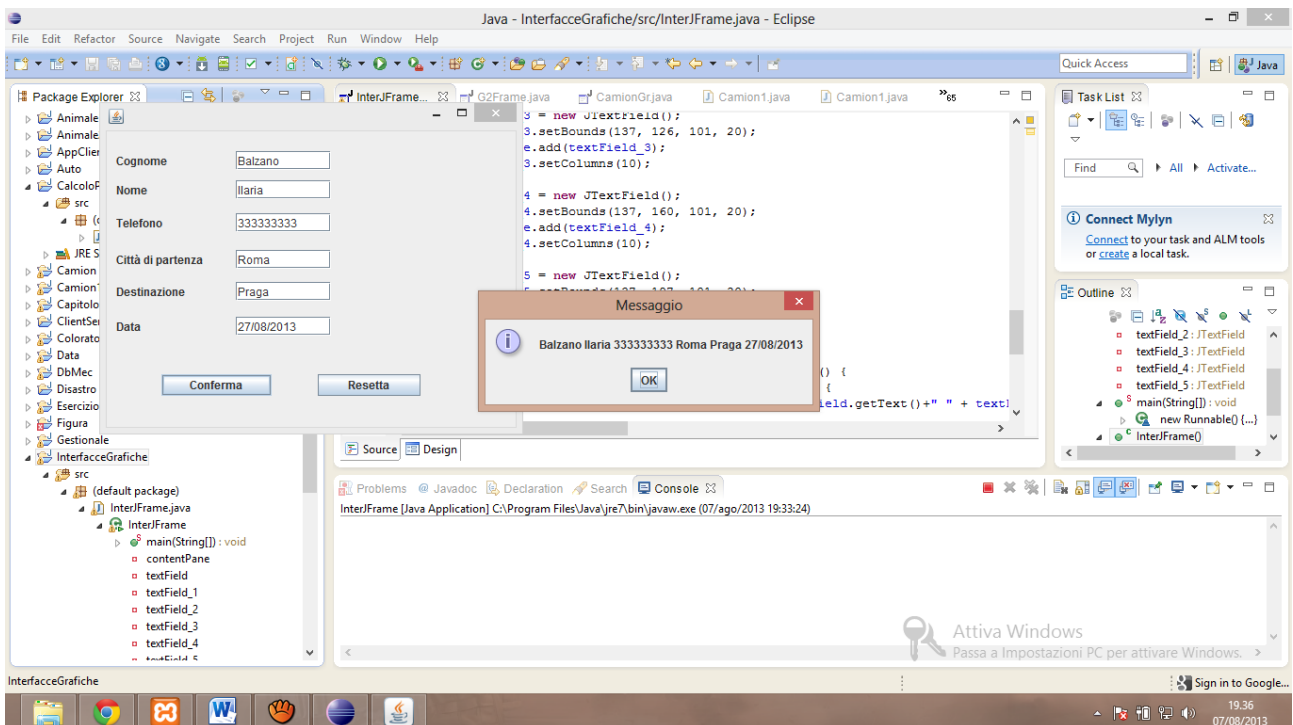
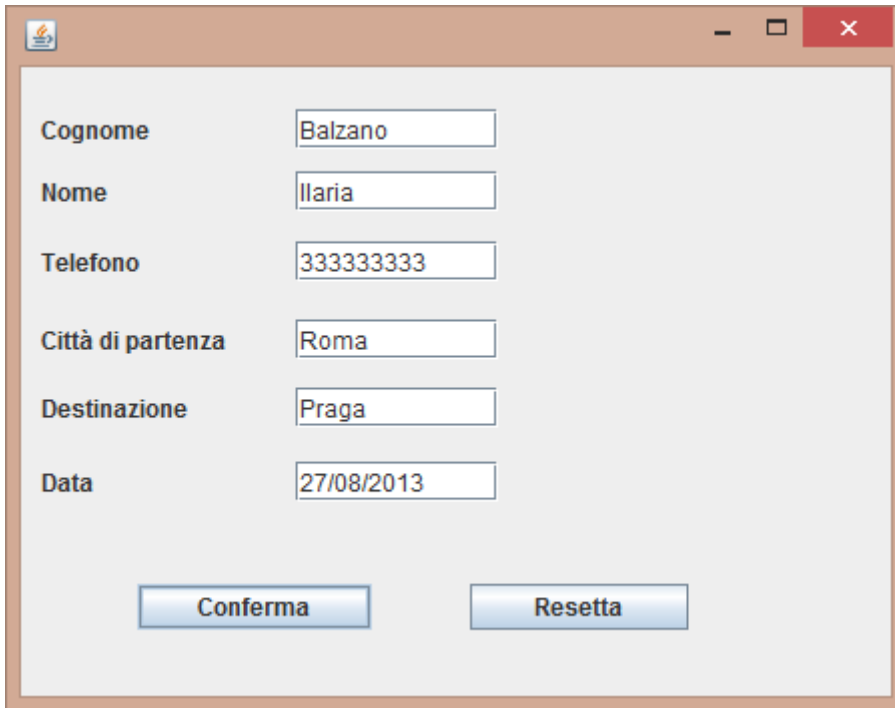
- 1) Creare un'applicazione che consenta ad un utente di prenotare il biglietto di sola andata per una data destinazione, utilizzando un' interfaccia grafica.

L'utente deve inserire le seguenti informazioni:

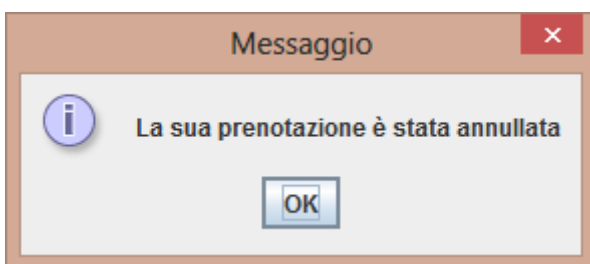
- cognome;
- nome;
- telefono;
- città di partenza;
- città di destinazione;
- data;

Dopo l'inserimento dei dati, l'utente può premere il pulsante "invia", per confermare la prenotazione o il pulsante "Resetta", per non confermarla.

- La pressione sul pulsante "Invia" ha come effetto la memorizzazione della prenotazione in un oggetto e la comunicazione all'utente stesso dell'avvenuta registrazione, mediante visualizzazione di una finestra di dialogo, verranno poi visualizzati i dati inseriti in una textarea.
- La pressione del pulsante "Resetta" ha come effetto la cancellazione dei dati inseriti dalla GUI e la comunicazione dell'avvenuto annullamento attraverso una finestra di dialogo



Premendo il tasto 'Resetta', invece, si ottiene la seguente finestra:





Codice sorgente:

```
import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JOptionPane;

public class InterJFrame extends JFrame {

    private JPanel contentPane;
    private JTextField textField;
    private JTextField textField_1;
    private JTextField textField_2;
    private JTextField textField_3;
    private JTextField textField_4;
    private JTextField textField_5;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
```

```

        try {
            InterJFrame frame = new InterJFrame();
            frame.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
};
}

```

```
/**
```

```
* Create the frame.
```

```
*/
```

```

public InterJFrame() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 353);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("Cognome");
    lblNewLabel.setBounds(10, 24, 75, 14);
    contentPane.add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("Nome");
    lblNewLabel_1.setBounds(10, 55, 75, 14);
    contentPane.add(lblNewLabel_1);

    JLabel lblTelefono = new JLabel("Telefono");

```

```
lblTelefono.setBounds(10, 90, 75, 14);  
contentPane.add(lblTelefono);
```

```
JLabel lblCittDiPartenza = new JLabel("Citt\u00E0 di partenza");  
lblCittDiPartenza.setBounds(10, 129, 117, 14);  
contentPane.add(lblCittDiPartenza);
```

```
JLabel lblNewLabel_2 = new JLabel("Destinazione");  
lblNewLabel_2.setBounds(10, 163, 101, 14);  
contentPane.add(lblNewLabel_2);
```

```
JLabel lblNewLabel_3 = new JLabel("Data");  
lblNewLabel_3.setBounds(10, 200, 62, 14);  
contentPane.add(lblNewLabel_3);
```

```
textField = new JTextField();  
textField.setBounds(137, 21, 101, 20);  
contentPane.add(textField);  
textField.setColumns(10);
```

```
textField_1 = new JTextField();  
textField_1.setBounds(137, 52, 101, 20);  
contentPane.add(textField_1);  
textField_1.setColumns(10);
```

```
textField_2 = new JTextField();  
textField_2.setBounds(137, 87, 101, 20);  
contentPane.add(textField_2);  
textField_2.setColumns(10);
```

```
textField_3 = new JTextField();  
textField_3.setBounds(137, 126, 101, 20);  
contentPane.add(textField_3);  
textField_3.setColumns(10);
```

```
textField_4 = new JTextField();  
textField_4.setBounds(137, 160, 101, 20);  
contentPane.add(textField_4);  
textField_4.setColumns(10);
```

```
textField_5 = new JTextField();  
textField_5.setBounds(137, 197, 101, 20);  
contentPane.add(textField_5);  
textField_5.setColumns(10);
```

```
JButton btnNewButton = new JButton("Conferma");  
btnNewButton.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent arg0) {  
        JOptionPane.showMessageDialog(null, textField.getText()+" "+  
textField_1.getText()+" "+ textField_2.getText()+" "+ textField_3.getText()+" "+ textField_4.getText()+"  
" + textField_5.getText() );  
    }  
});  
btnNewButton.setBounds(58, 258, 117, 23);  
contentPane.add(btnNewButton);
```

```
JButton btnNewButton_1 = new JButton("Resetta");  
btnNewButton_1.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {  
        textField.setText("");  
    }  
});
```

```

        textField_1.setText("");
        textField_2.setText("");
        textField_3.setText("");
        textField_4.setText("");
        textField_5.setText("");

        JOptionPane.showMessageDialog(null, "La sua prenotazione è stata
annullata");

    }

});

btnNewButton_1.setBounds(224, 258, 109, 23);

contentPane.add(btnNewButton_1);

}

}

```

- 2)** Scrivere un'interfaccia grafica in grado di visualizzare le preferenze degli utenti. Dovranno essere presenti:

Due textfield Nome e Cognome;

Un campo data di nascita con la scelta per il giorno e per il mese con una combobox e l'anno inseribile con una textfield. Dovranno essere controllate solo le corrispondenze giorno mese (inserire 31 Aprile non deve essere possibile) senza controlli sull'anno bisestile (Febbraio massimo 28 giorni). Eventuali errori dovranno essere comunicati con una finestra di dialogo;

Un gruppo di bottoni scelta "Vacanze preferite" con almeno due valori;

Una scelta multipla "Interessi" con almeno tre valori;

Due pulsanti "Invia" che scrive i dati inseriti in una textarea e "Resetta" che cancella tutti i campi inseriti.

Nome

Cognome

Data di nascita **Giorno**  **Mese**  **Anno**

**Vacanze Preferite**

Mare

Montagna

Campagna

**Interessi**

Informatica  Motori

Musica  Shopping

Dopo aver premuto il pulsante 'Invia':

The screenshot shows a Java Swing window with a light gray background and a brown border. At the top, there are standard window controls (minimize, maximize, close). The form contains the following elements:

- Nome:** Text field containing "Ilaria".
- Cognome:** Text field containing "Balzano".
- Data di nascita:** Three fields: "Giorno" (dropdown with "2"), "Mese" (dropdown with "Dicembre"), and "Anno" (text field with "1993").
- Vacanze Preferite:** Three radio buttons: "Mare" (selected), "Montagna", and "Campagna" (selected).
- Interessi:** Four checkboxes: "Informatica" (checked), "Musica" (checked), "Motori" (unchecked), and "Shopping" (checked).
- Buttons:** "Invia" and "Resetta" buttons at the bottom.
- Preview:** A white box on the right side displays the entered data: "Nome: Ilaria", "Cognome: Balzano", "Data di nascita: 2 2 1993", "Vacanze preferite: Mare, Campagn", and "2-Dicembre-1993".

Codice sorgente:

```
import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import javax.swing.JRadioButton;
import javax.swing.JCheckBox;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

import javax.swing.DefaultComboBoxModel;

public class G2Frame extends JFrame {
```

```

private JPanel contentPane;
private JTextField textField;
private JTextField textField_1;
private JTextField textField_2;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                G2Frame frame = new G2Frame();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the frame.
 */
public G2Frame() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 518, 514);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("Nome");
    lblNewLabel.setBounds(22, 22, 68, 14);
    contentPane.add(lblNewLabel);

    JLabel lblCognome = new JLabel("Cognome");
    lblCognome.setBounds(22, 58, 82, 14);
    contentPane.add(lblCognome);

    textField = new JTextField();
    textField.setBounds(114, 19, 115, 20);
    contentPane.add(textField);
    textField.setColumns(10);

    textField_1 = new JTextField();
    textField_1.setBounds(114, 55, 115, 20);
    contentPane.add(textField_1);
    textField_1.setColumns(10);

    JLabel lblDataDiNascita = new JLabel("Data di nascita");
    lblDataDiNascita.setBounds(22, 105, 97, 14);
    contentPane.add(lblDataDiNascita);

    JLabel lblGiorno = new JLabel("Giorno");
    lblGiorno.setBounds(114, 105, 46, 14);
    contentPane.add(lblGiorno);

    final JComboBox comboBox = new JComboBox();
    comboBox.setModel(new DefaultComboBoxModel(new String[] {"", "1",
"2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15",
"16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28",
"29", "30", "31"}));
}

```



```

comboBox.setBounds(153, 102, 44, 20);
contentPane.add(comboBox);

JLabel lblMese = new JLabel("Mese");
lblMese.setBounds(207, 105, 46, 14);
contentPane.add(lblMese);

final JComboBox comboBox_1 = new JComboBox();
comboBox_1.setModel(new DefaultComboBoxModel(new String[] {"",
"Gennaio", "Febbraio", "Marzo", "Aprile", "Maggio", "Giugno", "Luglio",
"Agosto", "Settembre", "Ottobre", "Novembre", "Dicembre"}));
comboBox_1.setBounds(240, 102, 97, 20);
contentPane.add(comboBox_1);

JLabel lblAnno = new JLabel("Anno");
lblAnno.setBounds(370, 105, 46, 14);
contentPane.add(lblAnno);

textField_2 = new JTextField();
textField_2.setBounds(413, 102, 68, 20);
contentPane.add(textField_2);
textField_2.setColumns(10);

JLabel lblVacanzePreferite = new JLabel("Vacanze Preferite");
lblVacanzePreferite.setBounds(22, 151, 108, 14);
contentPane.add(lblVacanzePreferite);

final JRadioButton rdbtnMare = new JRadioButton("Mare");
rdbtnMare.setBounds(21, 176, 109, 23);
contentPane.add(rdbtnMare);

final JRadioButton rdbtnMontagna = new JRadioButton("Montagna");
rdbtnMontagna.setBounds(22, 209, 109, 23);
contentPane.add(rdbtnMontagna);

final JRadioButton rdbtnCampagna = new JRadioButton("Campagna");
rdbtnCampagna.setBounds(22, 248, 109, 23);
contentPane.add(rdbtnCampagna);

JLabel lblInteressi = new JLabel("Interessi");
lblInteressi.setBounds(22, 289, 82, 14);
contentPane.add(lblInteressi);

final JCheckBox chckbxInformatica = new JCheckBox("Informatica");
chckbxInformatica.setBounds(22, 310, 97, 23);
contentPane.add(chckbxInformatica);

final JCheckBox chckbxMusica = new JCheckBox("Musica");
chckbxMusica.setBounds(22, 345, 97, 23);
contentPane.add(chckbxMusica);

final JCheckBox chckbxNewCheckBox = new JCheckBox("Motori");
chckbxNewCheckBox.setBounds(132, 310, 97, 23);
contentPane.add(chckbxNewCheckBox);

final JCheckBox chckbxNewCheckBox_1 = new JCheckBox("Shopping");
chckbxNewCheckBox_1.setBounds(132, 345, 97, 23);
contentPane.add(chckbxNewCheckBox_1);

JButton btnNewButton = new JButton("Invia");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        JTextArea textArea = new JTextArea();

```

```

        JTextArea textArea_1 = new JTextArea();
        textArea.setBounds(232, 145, 190, 100);

        String vacanzePreferite=""; // in base a ciò che uno
sceglie.

        if(rdbtnMontagna.isSelected())
            vacanzePreferite="Montagna";
        else if(rdbtnCampagna.isSelected())
            vacanzePreferite = "Campagna";
        else if(rdbtnMare.isSelected())
            vacanzePreferite="Mare";
        textArea.setText("Nome: " + textField.getText()+ "\n
Cognome: "+ textField_1.getText()+"\n Data di nascita: "+
comboBox.getSelectedItemAt()+ " "+ comboBox.getSelectedItemAt()+ " "+
textField_2.getText()+ "\n Vacanze preferite: " + vacanzePreferite );

String startDateString = comboBox.getSelectedItemAt()+"-
"+comboBox_1.getSelectedItemAt()+"-"+textField_2.getText(); ; // per
controllare le date per i comboBox
        DateFormat df = new SimpleDateFormat("dd-MMMM-yyyy");
        Date startDate=null;
        String newDateString = null;
        try
        {
            startDate = df.parse(startDateString);
            newDateString = df.format(startDate);
            textArea_1.setBounds(232, 250, 190, 23);
            textArea_1.setText(startDateString);
        } catch (ParseException e)
        {
            e.printStackTrace();
        }
        getContentPane().add(textArea);
        getContentPane().add(textArea_1);
        revalidate();
        repaint();
    }
});
btnNewButton.setBounds(88, 417, 115, 23);
contentPane.add(btnNewButton);

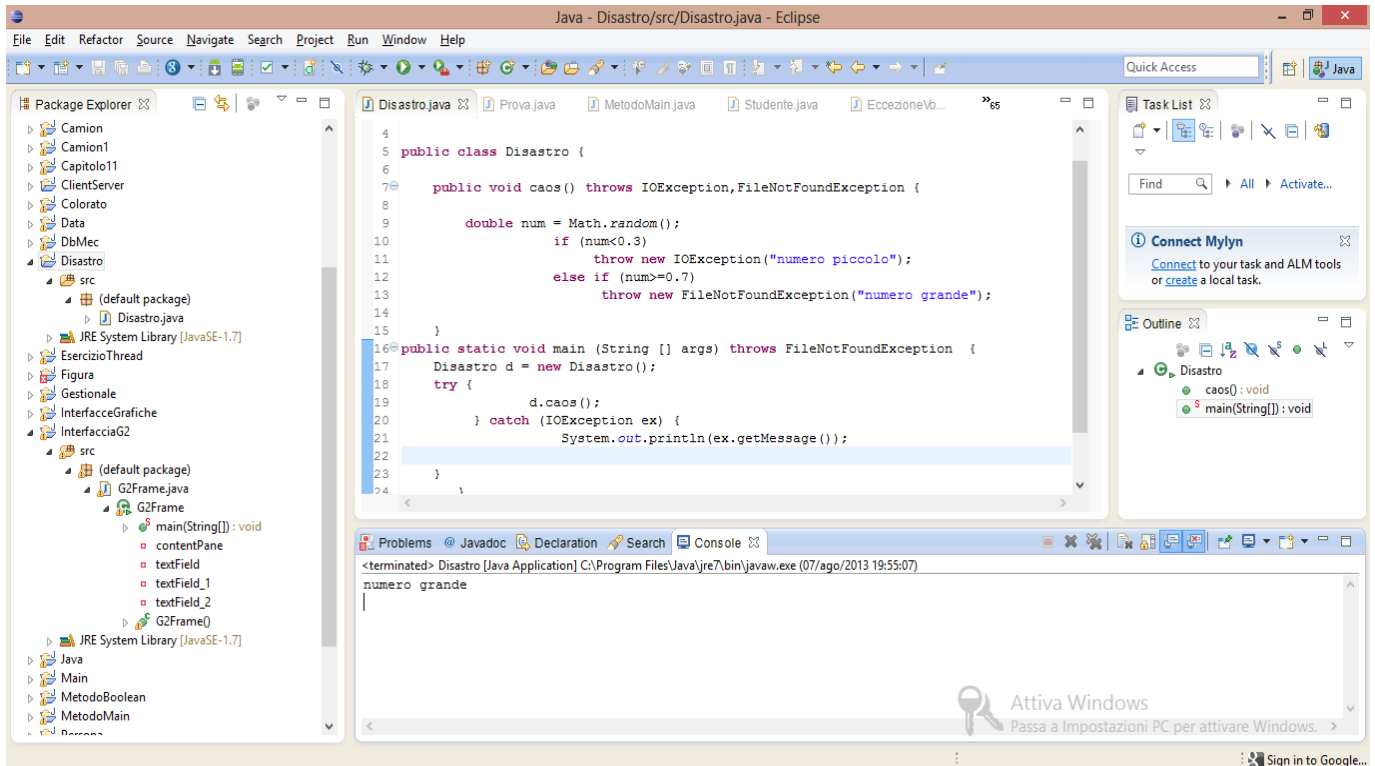
JButton btnNewButton_1 = new JButton("Resetta");
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText("");
        textField_1.setText("");
        textField_2.setText("");
        comboBox.setSelectedItemAt(""); // per resettare
        comboBox_1.setSelectedItemAt("");
        rdbtnMare.setSelected(false); // per resettare
        rdbtnMontagna.setSelected(false);
        rdbtnCampagna.setSelected(false);
        chckbxInformatica.setSelected(false);
        chckbxMusica.setSelected(false);
        chckbxNewCheckBox.setSelected(false);
        chckbxNewCheckBox_1.setSelected(false);
    }
});
btnNewButton_1.setBounds(244, 417, 115, 23);
contentPane.add(btnNewButton_1);
}

```

}

## Applicazione lancio eccezioni

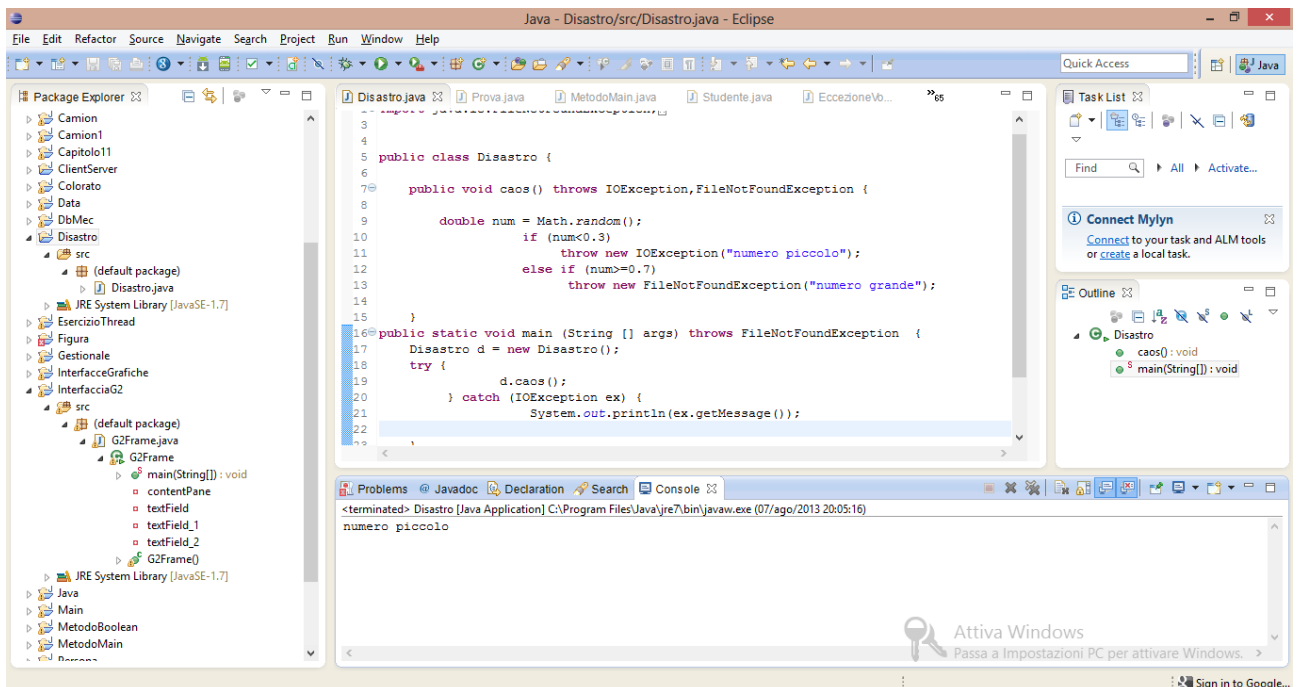
Scrivere una classe chiamata Disastro con un metodo caos() che random lancia un'eccezione di tipo IOException o FileNotFoundException:



The screenshot shows the Eclipse IDE with the following code in Disastro.java:

```
4
5 public class Disastro {
6
7     public void caos() throws IOException,FileNotFoundException {
8
9         double num = Math.random();
10        if (num<0.3)
11            throw new IOException("numero piccolo");
12        else if (num>=0.7)
13            throw new FileNotFoundException("numero grande");
14
15    }
16    public static void main (String [] args) throws FileNotFoundExcep... {
17        Disastro d = new Disastro();
18        try {
19            d.caos();
20        } catch (IOException ex) {
21            System.out.println(ex.getMessage());
22        }
23    }
24 }
```

The console output shows: `numero grande`



The screenshot shows the Eclipse IDE with the following code in Disastro.java:

```
3
4
5 public class Disastro {
6
7     public void caos() throws IOException,FileNotFoundException {
8
9         double num = Math.random();
10        if (num<0.3)
11            throw new IOException("numero piccolo");
12        else if (num>=0.7)
13            throw new FileNotFoundException("numero grande");
14
15    }
16    public static void main (String [] args) throws FileNotFoundExcep... {
17        Disastro d = new Disastro();
18        try {
19            d.caos();
20        } catch (IOException ex) {
21            System.out.println(ex.getMessage());
22        }
23    }
24 }
```

The console output shows: `numero piccolo`

Codice sorgente:

```
import java.io.FileNotFoundException;
```

```
import java.io.IOException;
```

```
public class Disastro {
```

```
    public void caos() throws IOException,FileNotFoundException {
```

```
        double num = Math.random();
```

```
        if (num<0.3)
```

```
            throw new IOException("numero piccolo");
```

```
        else if (num>=0.7)
```

```
            throw new FileNotFoundException("numero grande");
```

```
    }
```

```
    public static void main (String [] args) throws FileNotFoundException {
```

```
        Disastro d = new Disastro();
```

```
        try {
```

```
            d.caos();
```

```
        } catch (IOException ex) {
```

```
            System.out.println(ex.getMessage());
```

```
        }
```

```
    }
```

```
}
```