SelezionaAreaFrame

1. Command Pattern

- **Dove si applica:** Gestione delle azioni del pulsante "Seleziona"
- **Motivazione:** l'azione da eseguire quando l'utente clicca sul pulsante "Seleziona" è incapsulata all'interno di un ActionListener. Questo è un esempio del Command Pattern, dove l'azione (qui rappresentata dal metodo saveSelection) è separata dall'oggetto che la invoca, permettendo una gestione più flessibile delle azioni
- **Codice rilevante:**

```
// ActionListener per il pulsante Seleziona
selectButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String selectedArea = (String) areaComboBox.getSelectedItem();
        if (selectedArea != null) {
            saveSelection(selectedArea);
            JOptionPane.showMessageDialog(SelezionaAreaFrame.this, "Hai selezionato: " + selectedArea);
            dispose(); // Chiude la finestra
        } else {
            JOptionPane.showMessageDialog(SelezionaAreaFrame.this, message:"Seleziona un'area.", title:"Attenzione", JOptionPane.WARNING_MES
        }
    }
});
```

2. Strategy Pattern

- **Dove si applica:** Lettura del file CSV e salvataggio della selezione nel database
- **Motivazione:** codice per la lettura dei dati dal file CSV e il salvataggio nel database si può refactorizzare utilizzando strategie diverse per la lettura dei dati (da CSV, da un database, ecc.) e per il salvataggio
- **Codice rilevante**:

```
private void saveSelection(String selectedArea) {
    String query = "UPDATE \"OperatoriRegistrati\" SET \"centro_monitoraggio\" = ? WHERE \"username\" = ?";
    try (Connection connection = databaseManager.getConnection(); // Ottieni connessione dal DatabaseManager
         PreparedStatement preparedStatement = connection.prepareStatement(query)) {

        preparedStatement.setString(parameterIndex:1, selectedArea);
        preparedStatement.setString(parameterIndex:2, username);

        int rowsAffected = preparedStatement.executeUpdate();
        if (rowsAffected > 0) {
            System.out.println("Centro di monitoraggio selezionato salvato per l'utente: " + username);
        } else {
            System.out.println(x:"Nessun record aggiornato. Verifica che l'username sia corretto.");
        }
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, message:"Errore durante il salvataggio della selezione.", title:"Errore", JOptio
        ex.printStackTrace();
    }
}
```

```
private void loadAreas() {
    try (BufferedReader br = new BufferedReader(new FileReader(csvFilePath))) {
        String line;
        // Ignora la prima riga se è l'intestazione
        br.readLine();

        while ((line = br.readLine()) != null) {
            String[] data = line.split(regex:";"); // Usa ';' come separatore
            if (data.length > 1) {
                String areaName = data[1].trim();
                // Aggiungi solo aree non duplicate
                if (loadedAreas.add(areaName)) {
                    areaComboBox.addItem(areaName);
                }
            }
        }
    } catch (IOException ex) {
        JOptionPane.showMessageDialog(this, message:"Errore durante la lettura del file CSV.",
        ex.printStackTrace();
    }
}
```

3. Observer Pattern

- **Dove si applica:** Aggiornamento dell'interfaccia utente in seguito alla selezione di un'area
- **Motivazione:** Quando un'area viene selezionata, il resto dell'applicazione potrebbe dover essere aggiornato in base a questa selezione.
- **Codice rilevante**:

  JOptionPane.showMessageDialog(SelezionaAreaFrame.this, "Hai selezionato: " + selectedArea);

4. Facade Pattern

- **Dove si applica:** Interazione con il database
- **Motivazione:** il metodo saveSelection funge da Facade per la logica di aggiornamento del database, fornendo un'interfaccia semplificata per eseguire questa operazione complessa. L'uso del DatabaseManager per gestire la connessione al database è un esempio di come il Facade Pattern può ridurre la complessità delle interazioni con il database, nascondendo i dettagli della connessione e della preparazione delle query
- **Codice rilevante:**

```java
private void saveSelection(String selectedArea) {
    String query = "UPDATE \"OperatoriRegistrati\" SET \"centro_monitoraggio\" = ? WHERE \"username\" = ?";
    try (Connection connection = databaseManager.getConnection(); // Ottieni connessione dal DatabaseManager
         PreparedStatement preparedStatement = connection.prepareStatement(query)) {

        preparedStatement.setString(parameterIndex:1, selectedArea);
        preparedStatement.setString(parameterIndex:2, username);

        int rowsAffected = preparedStatement.executeUpdate();
        if (rowsAffected > 0) {
            System.out.println("Centro di monitoraggio selezionato salvato per l'utente: " + username);
        } else {
            System.out.println(x:"Nessun record aggiornato. Verifica che l'username sia corretto.");
        }
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, message:"Errore durante il salvataggio della selezione.", title:"Errore", JOpti
        ex.printStackTrace();
    }
}
```