

ABSTRACT CLASSES:

```
public abstract class Movie {
    protected String title;
    protected String gender;
    protected String ageRestriction;
    protected int duration;
    protected String namePoster;

    public Movie(String title, String gender, String ageRestriction, int
duration, String namePoster) {
        this.title = title;
        this.gender = gender;
        this.ageRestriction = ageRestriction;
        this.duration = duration;
        this.namePoster = namePoster;
    }

    public static ArrayList<Movie> consultMovies(String fileName) throws IOException {
        Gson gson = new GsonBuilder().setDateFormat("MMM d, yyyy HH:mm:ss a").
            setPrettyPrinting().create();
        ArrayList<Movie> movies = new ArrayList<>();
        FileManager.createFile(fileName);
        String moviesJson = new String(Files.readAllBytes(Paths.get(fileName)));
        if (gson.fromJson(moviesJson, ArrayList.class) != null) {
            java.lang.reflect.Type typeMovies = new TypeToken<ArrayList<MovieBillboard>>() {
            }.getType();
            movies = gson.fromJson(moviesJson, typeMovies);
            return movies;
        }
        return movies;
    }
}

public abstract class Person {
    protected String id;
    protected String name;

    public Person(String id, String name) {
        this.name = name;
        this.id = id;
    }

    public String getId() {
        return id;
    }

    public String getName() {
        return name;
    }
}
```

```

int month = Integer.parseInt((String) cmbxMonth.getSelectedItem());
String securityCode = txtSecurityCode.getText();

Gson gson = new GsonBuilder().setPrettyPrinting().create();

if (fullName.trim().isEmpty()) {
    JOptionPane.showMessageDialog(
        null, "Full name is required", "Error", JOptionPane.ERROR_MESSAGE);
    return;
} else if (!fullName.matches("[a-zA-Z ]{2,24}+$")) {
    JOptionPane.showMessageDialog(
        null, "Full name can have only letters and its length must be between 3 and 16",
        "Error", JOptionPane.ERROR_MESSAGE);
    return;
}

if (identifyCard.trim().isEmpty()) {
    JOptionPane.showMessageDialog(
        null, "identify card is required", "Error", JOptionPane.ERROR_MESSAGE);
    return;
} else if (!InputDataValidation.validateCI(identifyCard)) {
    JOptionPane.showMessageDialog(
        null, "identify card was wrong", "Error", JOptionPane.ERROR_MESSAGE);
    return;
}

if (creditCard.trim().isEmpty()) {
    JOptionPane.showMessageDialog(
        null, "credit card was wrong", "Error", JOptionPane.ERROR_MESSAGE);
    return;
}

```