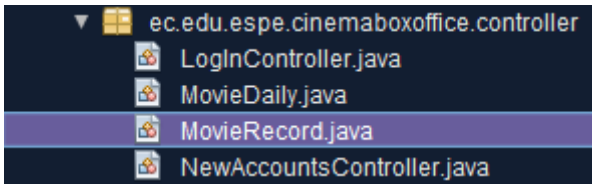
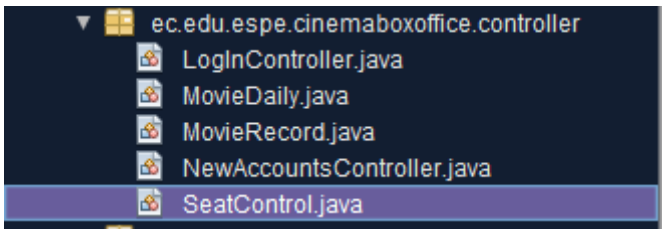
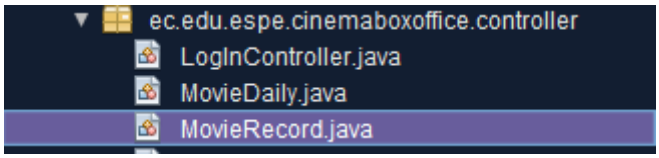
Single responsibility in Classes




```
42
43      public String defineRoom(int numberRoom) {
44          if (numberRoom > 0 & numberRoom < 7) {
45              if (numberRoom > 0 & numberRoom < 4) {
46                  return "2D";
47              } else {
48                  return "3D";
49              }
50          }
51          return "-";
52      }
```




```
12  public class SeatControl {
13
14      public String defineRoom(int numberRoom) {
15          if (numberRoom > 0 & numberRoom < 7) {
16              if (numberRoom > 0 & numberRoom < 4) {
17                  return "2D";
18              } else {
19                  return "3D";
20              }
21          }
22          return "-";
23      }
24  }
25
```



```java
public void controlSeats(Movie movie, int x, int y, boolean seatAvailability) throws IOException {
    Gson gson = new GsonBuilder().setDateFormat("MMM d, yyyy HH:mm:ss a").setPrettyPrinting().create();
    ArrayList<Movie> movies = Movie.consultMovies("Billboard.json");
    for (int z = 0; z < movies.size(); z++) {
        if (movie.getTitle().equals(movies.get(z).getTitle())) {
            int cont = 0;
            for (int i = 0; i < 15; i++) {
                for (int j = 0; j < 10; j++) {
                    if (i == x & j == y) {
                        movies.get(z).getRoom().getSeats()[cont].setSeatAvailability(seatAvailability);
                    }
                    cont++;
                }
            }
            FileManager.deleteFile("Billboard.json");
            FileManager.writeFile("Billboard.json", gson.toJson(movies));
        }
    }
}
```

```java
public class SeatControl {

    public void controlSeats(Movie movie, int x, int y, boolean seatAvailability) throws IOException {
        Gson gson = new GsonBuilder().setDateFormat("MMM d, yyyy HH:mm:ss a").setPrettyPrinting().create();
        ArrayList<Movie> movies = Movie.consultMovies("Billboard.json");
        for (int z = 0; z < movies.size(); z++) {
            if (movie.getTitle().equals(movies.get(z).getTitle())) {
                int cont = 0;
                for (int i = 0; i < 15; i++) {
                    for (int j = 0; j < 10; j++) {
                        if (i == x & j == y) {
                            movies.get(z).getRoom().getSeats()[cont].setSeatAvailability(seatAvailability);
                        }
                        cont++;
                    }
                }
                FileManager.deleteFile("Billboard.json");
                FileManager.writeFile("Billboard.json", gson.toJson(movies));
            }
        }
    }
}
```

The solution is to create a new class called "SeatControl" where the two methods comply with the OOP rules, since they were previously implemented in the "MovieRecord" class, breaking with the sole responsibility at the class level.

Single responsibility at the method level

```java
private void btnChoosePosterFileActionPerformed(java.awt.event.ActionEvent evt) {
```

```java
private void btnBillboardSaveActionPerformed(java.awt.event.ActionEvent evt) {
```

These methods are found in the package view class "FrmMovieBillboard" that control the saving of movies and the control of data entry in the textboxes, the solution is to modularize the method and the new methods refactor them to the InputValidation class found in the package utils.