



"An easy way to organize a multiplex cinema"

Object Oriented Programming project

Students:

- Giacomo Bartoli
- Aldo Junior Simoncini

1). Analysis of the problem:

Cineplus is a program whose aim is to simulate the management of a multiplex cinema.

Using a simple GUI every user can organize the projections, add and remove films, booking reservations visualizing a nice table which represents free and taken seats, and have a small sum up of the cash ordered by date.

The first screen of the program is a quick login. This is necessary to separate the user part from the administration. Administrators can edit the projections, establish prices and see the cash report for each day.

On the other hand, users can only book seats for clients.

So, here there is a small overview of Cineplus main features:

Admin:

- Add, remove or edit a film from projections.
- Setup title, genre, duration and poster for each film.
- Set the number of rooms in the cinema.
- Set number of seats for rows and columns.
- Set ticket price, which differs by type: student, adult, child or people with disabilities.
- Add or remove other users, both admin and normal users.
- View the cash report for each date.

User:

- View daily projections.
- Select film and set number of reservations.
- Print tickets.

It is important to understand that there is a strong contraposition between what user and admin can do. That is because, in this way, administrators can set up all the parameters and then the program works without any other intervention, as it should be in a real multiplex cinema.

At the end, all the data, including projections, cash and user settings, are stored in binary files. To assure data persistence, settings and projections are written on files when the user quits Cineplus.

Coding:

All the program is written in Java.

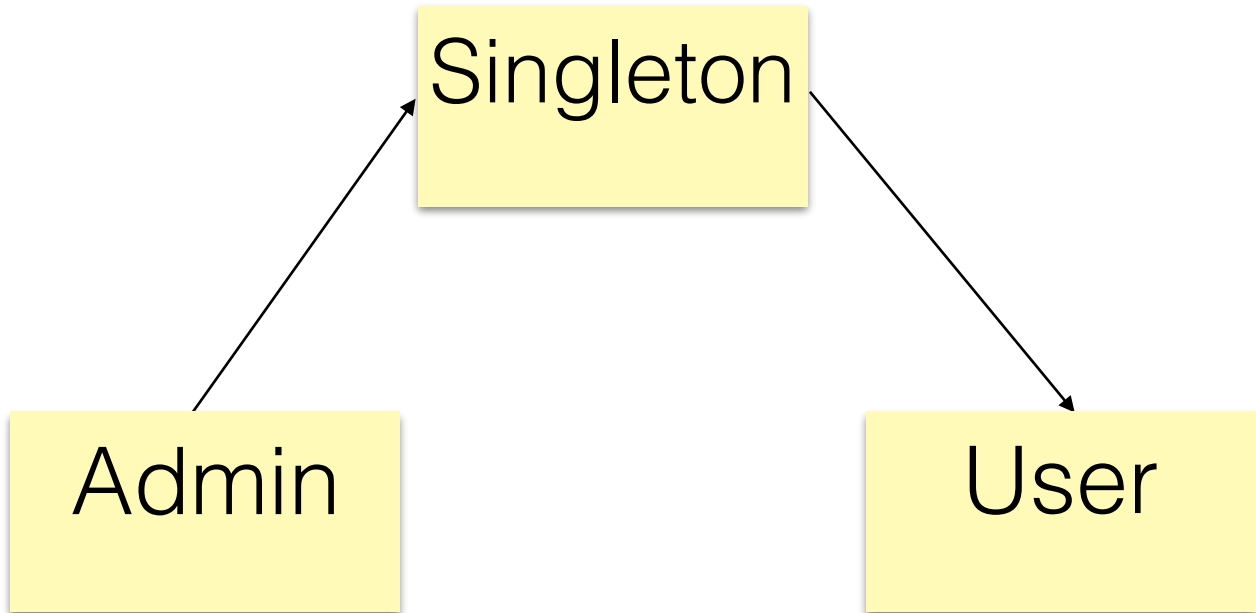
For the graphic user interface it has been used the Swing class.

All the basic rules of the **Java Code Convention** made by Oracle have been observed:

- the classes are written using capital letter at first.
- the name of variables are completely lowercase, except for some complicated names that adopt camel case standard.
- every variable has a specific access control: private, public or protected.
- every class has getters and setters only if it is necessary. Otherwise variables are private.
- all the classes are encapsulated to ensure the independence of the program and avoid bad accesses to the fields.
- every variable has a name which is self-explanatory.
- the code is correctly indented.
- "magic numbers" are put into constants.
- every class and every method has the Javadoc for gather informations.
- the source code of the class has no more than 2000 rows of code.

2). Architectural design:

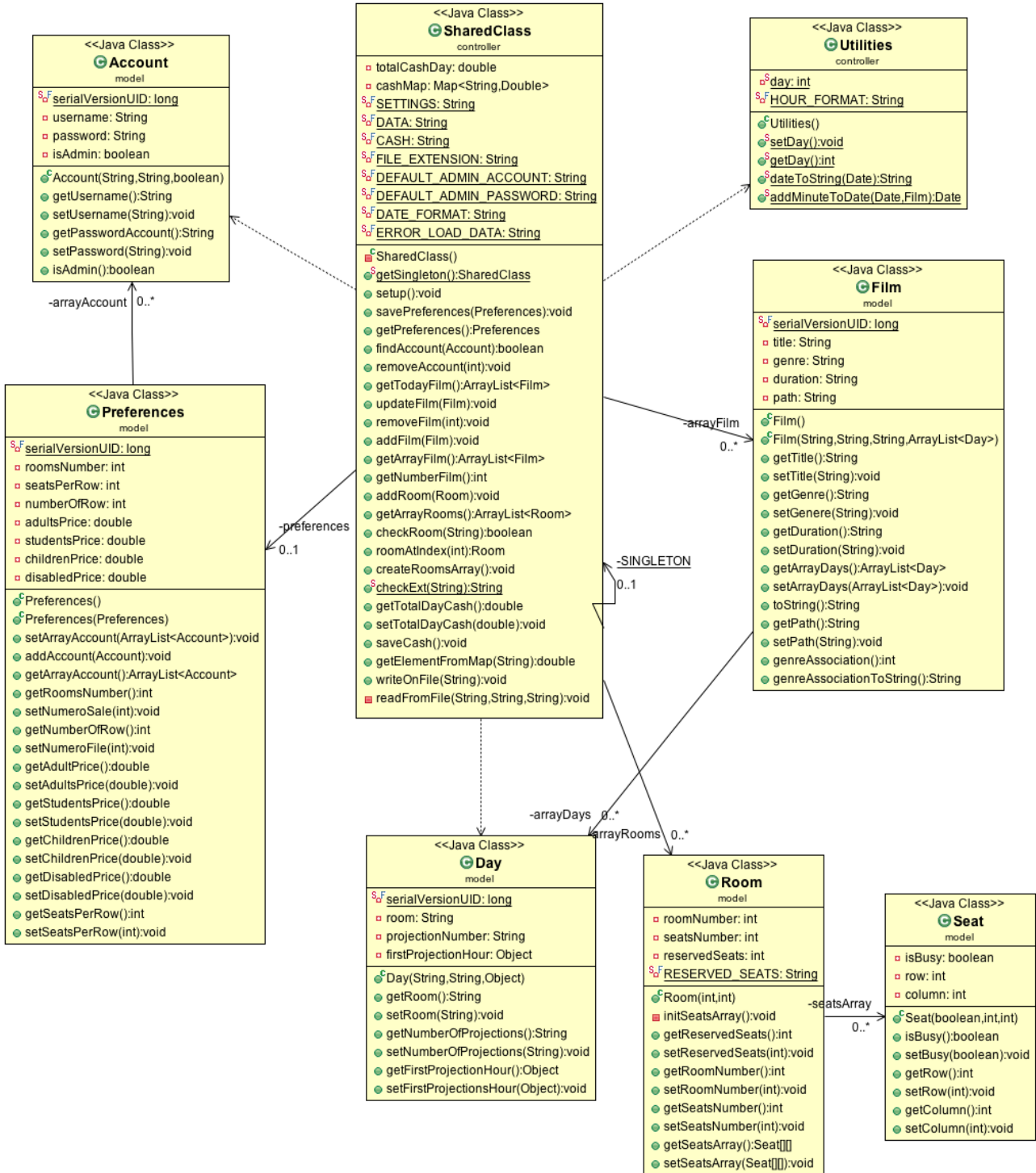
The entire design phase has been guided by the pervasive use of the MVC pattern. In this way it is possible a strong split up between the data representation, visualization and logical construction.



The main idea of Cineplus is very easy. Administrator set data to let Cineplus work, user reads these data.

This is possible because of Singleton. This object coordinates actions across the system and it is the core of the application. It is intermediary between admin and user, its task is to takes data from administration part and give them to user part. Another important function of singleton is to save all application data on file.

Model:



Description of the main aspects:

SharedClass: this is the **singleton pattern**. It is fundamental for passing data between different controllers. Obviously, there is only one instance of this class in all the application. All the methods can be called from any part of the code. This class follows the **hiding information** principle: the access is not direct but you must use the suitable getter to invoke `SharedClass`, `SharedClass.getSingleton()`;

Utilities: in this class we put every method that we consider useful. This is to avoid repetition of code.

Preferences: this class represents the settings of the application. Settings can be modified only by administrators. It includes prices, the size of the room and the possibility to add and remove other Cineplus users.

Day: it is the daily films projections. The constructor get only the beginning hour of a specific film and the number of projections. Then, Cineplus calculates automatically start-end time considering a break of 30 minutes between a projection and another one.

For example, if you set up a film starting at 6.00pm with a duration of 90 minutes and 3 projections, Cineplus does the following operations:

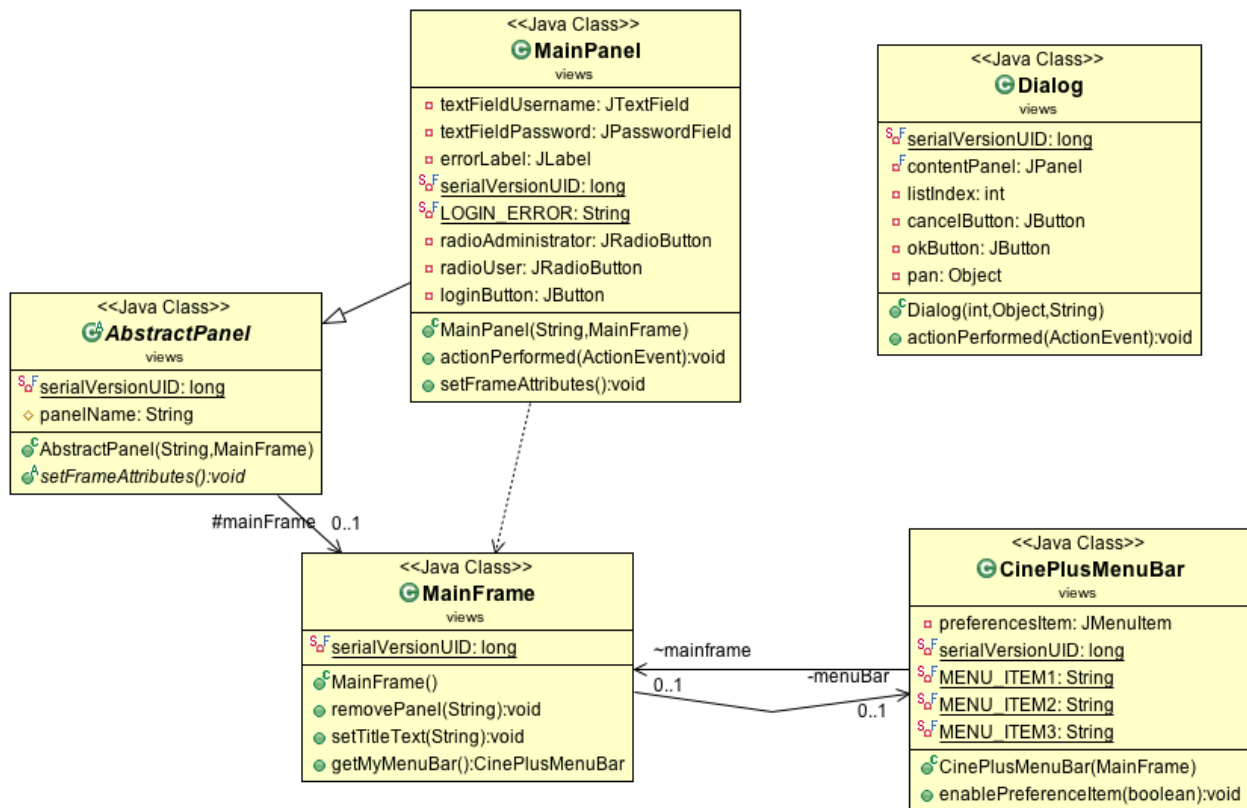
$(6\text{pm} + 90\text{ minutes} + 30\text{ minutes of break}) \times 3$.

To do this, many methods from Utilities class are invoked.

Film: this class contains the typical fields that a film needs: title, genre, duration, the room where is shown, etc.. . Maybe it could be expanded inserting more details as film cast, the director and rating. Moreover, film class includes an ArrayList of Days (`List<Day>`) which stands for the set days when the film is projected.

Room: this class is instanced every time a user add a film. There is only one limit: one room for film daily. Room class contains also an array of Seats, which is a small class made of 3 fields: row/column number and a boolean variable to indicate if the seat is free or reserved.

View:



This view asserts the **Template Method**.

These classes focus on the login panel, at the beginning of the application.

AbstractPanel, as the name suggests, is an abstract class which declares setFrameAttributes() method. This one is implemented in the MainPanel class, which extends AbstractPanel.

In this way the panel contains the MainFrame, and the MainPanel has all the swing components to built the login form.

Dialog class is used to prevent data input error by the user. When the functions that control data, typed by user, intercept an error, a dialog class is instanced and it displays an error message. This message is set passing a string in the constructor.

At the end, CineplusMenuBar is the JMenuItem placed under the panel title. This bar is enabled only when the administrator submit correctly its data, because through CineplusMenuBar you can access to preferences.

When the login credentials are submitted into the JTextField and JPasswordField, Cineplus creates a new Account (model package) and compare data with the array of Accounts in the SharedClass. If data matches, user can access to the application.

3). Packages organization:

- **src.main**

Here is the main of the program. Inside the main method there are three methods that are called. The first one is *Utilities.setDay()*, which set the proper date every time Cineplus starts. This is necessary otherwise the projections table can not find what films display in the user view.

setGUI() method uses an external library (JTattoo) to load the user interface theme. The last function of the main package is the *saveCash()* method.

As the name suggest, this action load the cash of the past events.

- **src.model**

The model part of the application. In this package are collected all the classes that grab data from the view. All the details were given in the previous section.

- **src.view**

Inside this package we put all the views that do not belongs nor to the user part neither to the admin part. Fundamentally, there are some classes which represents the initial login and an *AbstractPanel* whose *setFrameAttributes()* method is implemented inside the classes that extend this panel. The *Dialog* class has been used for showing to the user all the error data input.

- **src.viewadmin**

This package leads all the administration view of Cineplus. *AdminSplitPane* class is the main view of this section, but the most complex class is the *AddFilmPanel*. The final version has been refactored many times, since we decided to instanced the graphic components inside cycles. There are also many private methods that are used only inside the class. The remaining classes are simpler and do not deserve to be described, main aspects were listed previously.

- **src.viewuser**

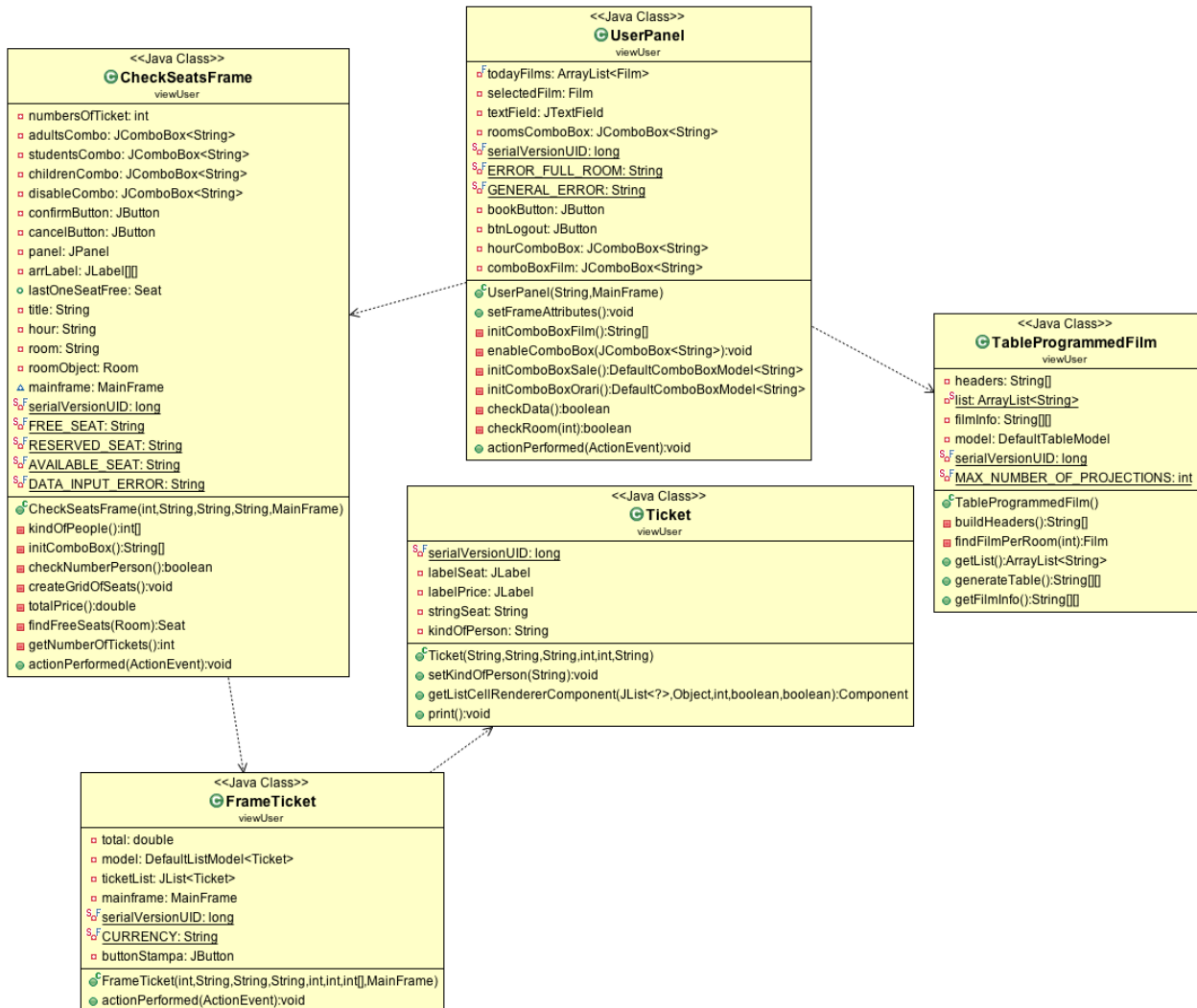
The graphic user interface for the user part of the application. *UserPanel* class is a *JPanel* which englobes the *TableProgrammedFilm* class. In the *CheckSeatsFrame* there is a small sum up of the free or reserved seats inside the selected room. The *FrameTicket* class contains all the ticket booked.

- **src.controller**

This is supposed to be the most interesting part of the program, even if it is composed by only two classes. The first one, *SharedClass*, is a Singleton. All the data are stored in arrays that are saved into this class. In every part of the code they can be called using the *SharedClass.getSingleton()* method. The other class, *Utilities*, is a set of common methods that had been collected in this class to avoid codes repetition.

4). Workflow development:

- Giacomo Bartoli (user side):



UserPanel is the first class instanced when an user access to Cineplus.

Inside this class there is an HTML table which shows the current projections. At the bottom there are three combo box, where the user can select the interested film and booking a pre-established number of tickets.

Then, a CheckSeatFrame is instanced. In this panel, user can see all the seats inside the room. Seats differs from color:

- green spot = free.
- yellow spot = available.
- red spot = reserved.

Cineplus automatically detects the favorite free seats inside the room and paints them in yellow. If user confirms, yellow seats become reserved and the application repaint them in

red. The algorithm of searching free seats take as input the number of tickets and scrolls the grid of seats looking for free seats closed to each other.

Every time a free seat is discovered, a variable is incremented. If a red seat is found, the variable is re-assigned to zero. When the variable reach the number of tickets required, the application has find the sequence of seats available for booking.

Computational costs of this method is $O(2n)$, which is expansive, but we have to consider that the grid cannot be bigger than 20-25 rows/columns in a standard cinema room. This means that array size should be at most 50 cells maximum.

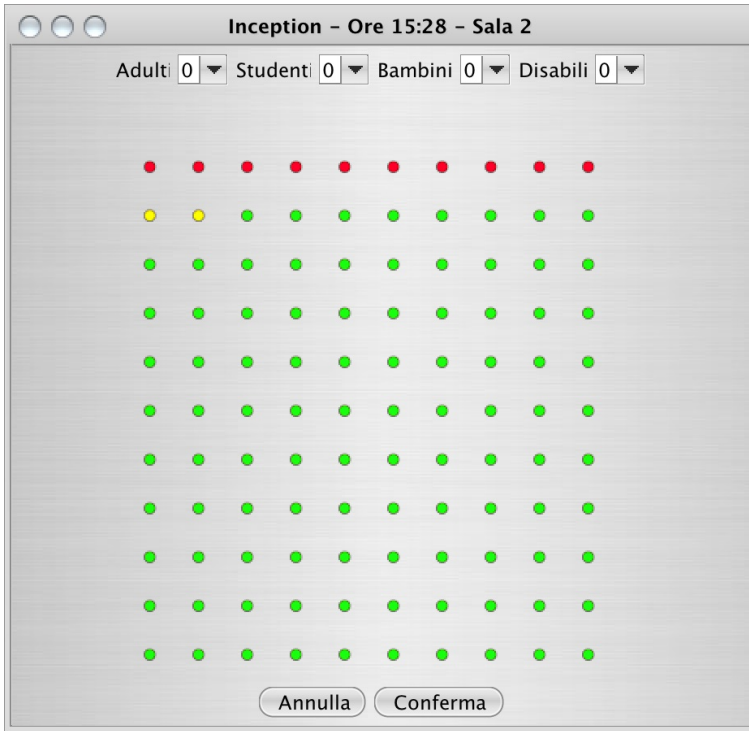
The next step is to show to user tickets (Ticket class) were all the details are recapped. At the end there is the total amount. The print function starts automatically.

Cineplus			
SALA 1	SALA 2	SALA 3	SALA 4
Shutter Island 15:29	Inception 15:28	The social network 16:52	La grande bellezza 15:30
	Inception 17:28	The social network 18:32	La grande bellezza 17:30
		The social network 20:12	La grande bellezza 19:30
			La grande bellezza 21:30
			La grande bellezza 23:30

Film: Orario: Sala: Numero biglietti:

A small preview of the first panel: films are ordered by room. The user selects the film from the first combo box and then put the number of tickets. Clicking on "Prenota" it switches to CheckSeatsFrame panel.

TableProgrammedFilm is the class that draws this table. Here, there is a sort of parser which takes the number of projections ,for each film of the current date, and converts all these data in HTML standard. See the *generateTable()* method inside the class.



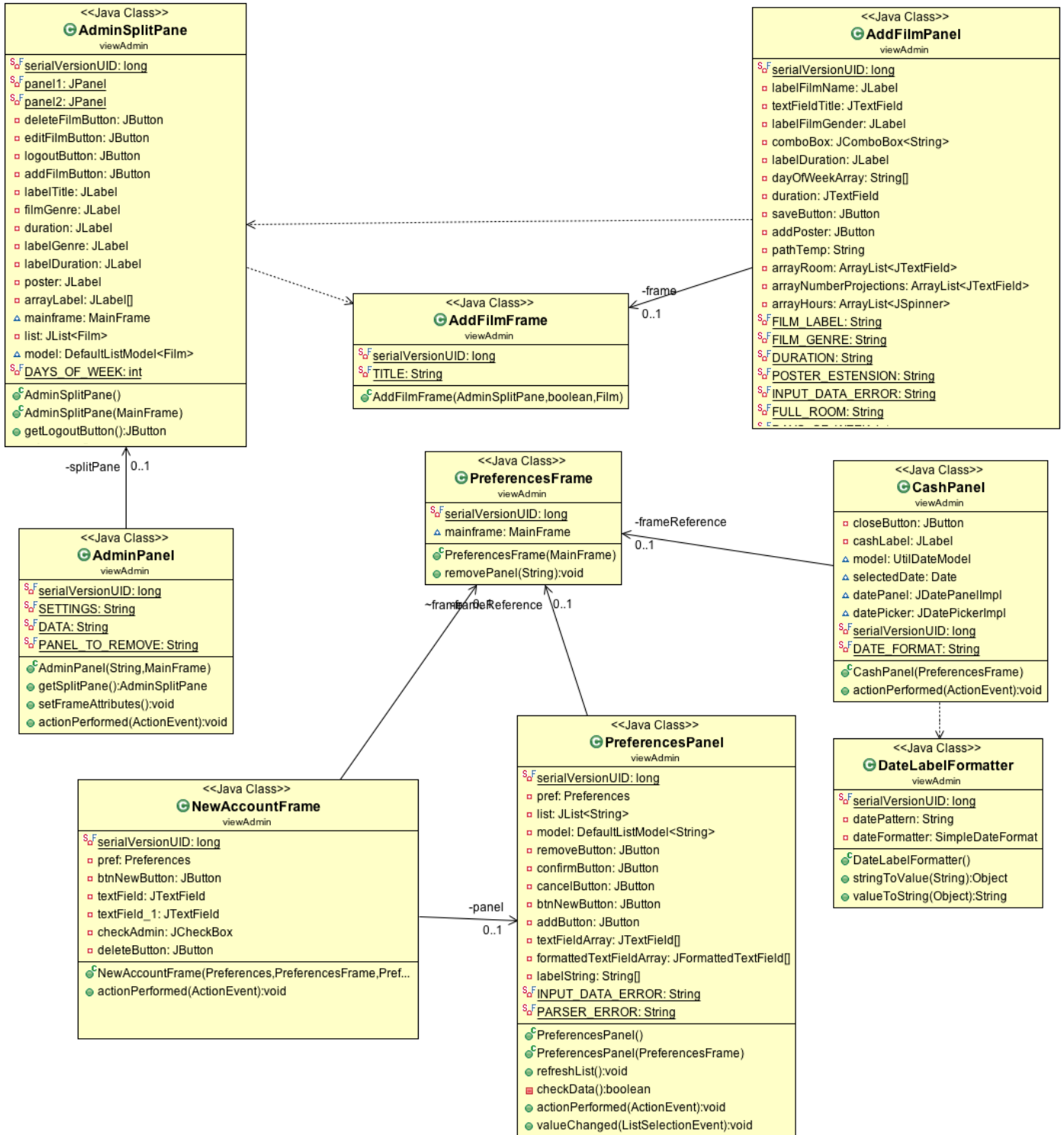
CheckSeatsFrame: in this panel is represented the room of the selected film. There are four checkbox at top that let the user selects how many adults, children, students or people with disabilities there are.

Ticket: after have confirmed seats, Cineplus opens this window were there are all the tickets sold. Every one of them has the necessary details: film title, room, hour and seat.

At the bottom of the windows there is the total amount, which is saved automatically into the file "cash.dat".



• Aldo Junior Simoncini (admin side):



This is the UML scheme for the admin part of Cineplus.

Default login credential for Cineplus are USERNAME: admin PASSWORD: admin. After the first login, they can be modified.

At first, the AdminPanel is loaded.

AdminPanel is a class that extends from AbstractPanel.

Inside the AdminPanel there is the AdminSplitPane class instanced. This view shows, on the left side, all the films that had been added to the projections and, on the right side, the details about the selected film. This is possible because inside the AdminSplitPane there are two methods: the first one, refreshList(), scrolls the array of films into the singleton and put the data into the split panel. The other one, cleanScreen(), put all the strings empty before call refreshList(). Into the listeners of this class, there is also the necessary code to load an eventually poster of the film. Just click on the "scegli locandina" button to instance a JFileChooser, which incorporates a FileFilter to ensure that only .jpg files are uploaded. To change the file extension accepted it is sufficient to edit the POSTER_EXTENSION constant inside the same class. Cineplus does not copy the file inside its bundle, but it saves only the file path, that is a String field inside the Film class.

On the whole, AddFilmPanel class is called when the admin wants to add a film to the projections. This panel shows all the parameters to add a film: it is possible to specific a room, number of projections, start time, genre and duration for every day. When the user press "save", the application add a new film to the array of films into the SharedClass (singleton).

Through the JMenuBar on the top, user can access to the preferences of Cineplus.

This view is based on the PreferencesPanel class and the Preferences class in the model.

PreferencesPanel class contains a Preferences field, so the view takes all the data from there. The button "visualizza incassi" is the direct access to open a CashPanel, where there is a sort of calendar made of DateLabelFormatter. Just select a date to see the cash report.

All the data, including settings, prices, projections and cash reports, are stored into .dat files in the same directory of Cineplus.

On the right part of the preferences panel there is a small menu where user can organize accounts. There is not much to say about this function. It is very basic: insert the name and a password to create a new user. Tick the box if you want the user be administrator of Cineplus. This view refers to the Account (model package) and has only three fields with setters and getters. When the administrator click on the logout button all the preferences are saved.

Preferenze

Numero sale:

Numero posti per fila:

Numero file:

Prezzo adulti:

Prezzo studenti:

Prezzo bambini:

Prezzo disabili:

Account Admin e Utenti:

admin
user

Visualizza incassi

Aggiungi Rimuovi

Annulla Conferma

Here the admin can set all the parameters of its multiplex cinema. There is a strong control for the input: strings and no sense characters are not accepted.

Preferenze

april 2014

dom	lun	mar	mer	gio	ven	sab
30	31	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	1	2	3
4	5	6	7	8	9	10

Today: 23-apr-2014 Clear

Incaso al 23 04 2014 € 77.00

Indietro

In this windows the admin can see the cash report daily. This view uses the Cash model and a DateLabelFormatter.

Admin

Cineplus

- Shutter Island
- La grande bellezza
- The social network
- Inception

Inception

Genere: Azione

Durata: 90

Lun: 19:50 21:50

Mar: 19:50 21:50 23:50

Mer: 19:50 21:50

Gio: 19:50 21:50

Ven: 19:50 21:50

Sab: 19:50 21:50


Dom: 19:50 21:50

Aggiungi film

Modifica film

Cancella film

Logout



This is the first view of the administration. It is a JSplitPane where on the left side is filled using a JList<Film>.

5). Considerations:

To end this paper, we should consider some aspects of the development process.

We started from scratch with a small scheme on what Cineplus would have to be.

Later, we designed the graphic interface.

The next step was the model part of the application. This was the most complicated phase.

What keeps us busy in testing and re-testing was the control on the input user, because of the complexity of our algorithms.

The last one difficulty was to optimize the code. We refactored many methods avoiding repetition of code and we used suggested tools (PMD, FindBugs, CheckStyle) to improve performance.

6). Testing:

We tested Cineplus for almost a week and we have to say that there are not critical bugs.

To be sure everything works fine, please stick to the following rules:

- Every new film must have title, genre and duration.
- Fill correctly all the fields concerning the number of projections and the rooms.
- Rooms must be numbers and not letters.
- If you do not want to associate a room, leave the field empty.
- If a film has no projections do not fill the textfield using zero, just leave the field empty.
- Every film cannot have a number of projection greater than 6. To edit this, please refer to the *MAX_NUMBER_OF_PROJECTIONS* constant in the *TableProgrammedFilm* class.
- The default values for the administrator are
 - Username: admin
 - Password: admin
- In one room there is only a film for a day. Films can have different rooms only for different days.
- When you switch from the admin part of Cineplus to the user part, be careful of clicking the logout button. In this way Cineplus automatically saves all the data, including preferences, accounts and cash.
- When the *UserPanel* shows all the film at the current date, it has been considered a break between one projection and another one of 30 minutes. This value can be modified, look at the *BREAK* constant inside *src.controller.Utilities* class.

7). Notes:

We are proud about this application because it does everything we estimated from the beginning. A real multiplex cinema could use Cineplus as management software.

In the first version of the project we planned a features to show projections on a web server. We tried to do this, using JApplet technology and it was worked. However, there were many problems with certificates, so we decided to discard this option.

Another possible way to fix this issue was to export the content of the TableProgrammedFilm, which is already HTML code, and loading through FTP on a web server.

For the GUI has been used an external library called JTattoo, all the details at the following link: <http://www.jtattoo.net/index.html>

For the calendar and the date picker, it has been used the JDatePicker library, available at this link: <http://sourceforge.net/projects/jdatepicker/>

All the Cineplus project can be found at the BitBucket repository:

<https://bitbucket.org/bgiacomo/oop1314-cineplus>

Official website:

<http://Cineplusproject.altervista.org>