**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



**ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΟ ΔΙΑΔΙΚΤΥΟ ΚΑΙ ΣΤΟΝ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ - 1η ΕΡΓΑΣΙΑ ΕΤΟΥΣ 2022-2023**

Κοντοπούλου Δέσποινα- Π21066

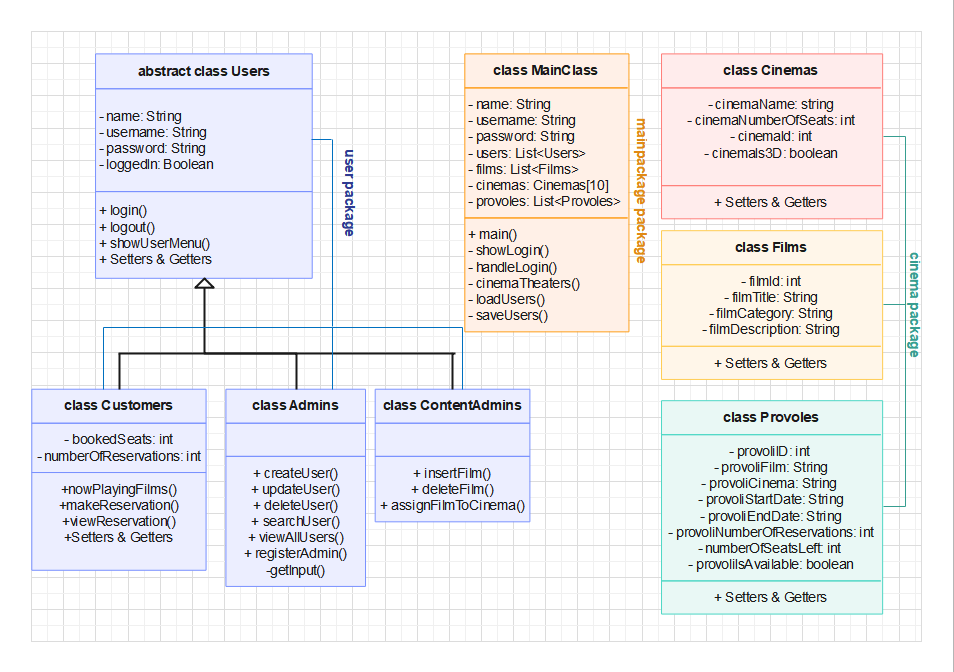
Κοτσοβού Πηνελόπη- Π21069

Ξύδης Τριαντάφυλλος- Π21120

**Ημερομηνία παράδοσης: 10/4/2023**

**Πίνακας Περιεχόμενων**

1. Κλάσεις
   1. Διάγραμμα Κλάσεων
   2. Η κλάση Users
   3. Η κλάση Customers
   4. Η κλάση Admins
   5. Η κλάση ContentAdmins
   6. Η κλάση Cinemas
   7. Η κλάση Films
   8. Η κλάση Provoles
   9. Η κλάση MainClass
2. Δημιουργία, χαρακτηριστικά και λειτουργίες αντικειμένων
   1. Customer
   2. Admin
   3. ContentAdmin
   4. Films
   5. Cinemas
   6. Provoles
3. **Κλάσεις**
   1. **Διάγραμμα Κλάσεων**

Το διάγραμμα UML των κλάσεων της εφαρμογής φαίνεται στην παρακάτω εικόνα. Οι κλάσεις Users, Customers, Admins και ContentAdmins ανήκουν στο πακέτο users. Οι κλάσεις Cinemas, Films και Provoles ανήκουν στο πακέτο cinema, ενώ η MainClass ανήκει στο πακέτο mainpackage.

* 1. **Κλάση Users**

Η κλάση Users είναι μια abstract κλάση την οποία κληρονομούν οι Customers, Admins και ContentAdmins. Πέραν από τις private μεταβλητές **name, username, password, loggedIn**, περιέχει τις «κενές» public μεθόδους *login, logout* και *showUserMenu* για είσοδο/ έξοδο του χρήστη και εμφάνιση του μενού αντίστοιχα, οι οποίες θα υλοποιηθούν ανά παιδί-κλάση. Επιπλέον περιέχονται *setters και getters* για τις private μεταβλητές.

* 1. **Κλάση Customers**

Η κλάση Customers κληρονομεί την κλάση Users, συνεπώς εμπεριέχει τις private μεταβλητές **name, username, password, loggedIn**, σε συνδυασμό με τις δικές της **bookedSeats** και **numberOfReservations**. Κάνει import την Films του package cinema. Όσον αφορά τις μεθόδους, υλοποιούνται οι προσαρμοσμένες για τον πελάτη *login* και *logout*, τύπου void, με ένα μήνυμα στην οθόνη, καθώς και η *showUserMenu,* τύπου void, που δείχνει τις λειτουργίες που μπορεί να εκτελέσει ο πελάτης.

Η Customers περιέχει επιπλέον *setters* και *getters* των bookedSeats, numberOfReservations, τις void μεθόδους *nowPlayingFilms, makeReservation, viewReservation* και 2 *constructors*, τον default και έναν με παραμετροποίηση του αντικειμένου.

* Void nowPlayingFilms(List<Films> films): Μέθοδος για την εμφανιση των ταινιών που προβάλλονται τώρα. Εάν η λίστα films είναι άδεια, εμφανίζεται στην οθόνη το αντίστοιχο μήνυμα. Ειδάλλως, για κάθε ταινία στην λίστα εκτυπώνεται ο τίτλος με την βοήθεια της string getFilmTitle() της κλάσης film.
* Void makeReservation(List<Provoles> provoles): Μέθοδος για την κράτηση θέσης. Με την βοήθεια της κλάσης Provoles και των μεθόδων της, εμφανίζονται οι ταινίες και οι κινηματογράφοι στις οποίες παίζονται.
* Void viewReservation(): Μέθοδος για την προβολή της κράτησης του χρήστη.
  1. **Κλάση Admins**

Η κλάση Admins κληρονομεί την κλάση Users, συνεπώς εμπεριέχει τις private μεταβλητές **name, username, password, loggedIn**. Όσον αφορά τις μεθόδους, υλοποιούνται οι προσαρμοσμένες για τον διαχειριστή *login* και *logout*, τύπου void, με ένα μήνυμα στην οθόνη, καθώς και η *showUserMenu,* τύπου void, που δείχνει τις λειτουργίες που μπορεί να εκτελέσει ο διαχειριστής.

Η Admins περιέχει επιπλέον τις μεθόδους *createUser, registerAdmin, getInput* τύπου Users, και τις μεθόδους *updateUser, deleteUser, searchUser, viewAllUsers* τύπου void.

* Users createUser(): Μέθοδος για την δημιουργία πελάτη. Με την χρήση των κλάσεων customers, bufferedReader και της μεθόδου getInput, ο χρήστης εισάγει τα απαραίτητα στοιχεία ώστε να δημιουργηθεί και έπειτα να επιστραφεί ο user.
* Users registerAdmin(): Μέθοδος για την δημιουργία contentAdmin ή admin. Αφού ο χρήστης διαλέξει τον τύπο χρήστη που θέλει να δημιουργήσει και γίνει έλεγχος για σφάλματα, δημιουργώ και επιστρέφω τον χρήστη ομοίως με την createUser.
* Users getInput(BufferedReader reader,Users user): Μέθοδος για την εισαγωγή δεδομένων από τον χρήστη. Με την βοήθεια ενός BufferedReader αντικειμένου κάνουμε set των μεταβλητών ενός χρήστη user, τον οποίο επιστρέφουμε .
* Void updateUser(List<Users> users): Μέθοδος για την ενημέρωση στοιχείων ενός χρήστη. Αναζητώ τον χρήστη στην λίστα users με βάση το username που εισάγεται, ζητώ και έπειτα κάνω set τα νέα στοιχεία του χρήστη με έλεγχο σφαλμάτων. Εάν δεν βρέθηκε χρήστης με το username που δόθηκε, εκτυπώνω το αντίστοιχο μήνυμα.
* Void deleteUser(List<Users> users, Users userInControl): Μέθοδος για την διαγραφή χρήστη. Ομοίως με την update, αναζητώ τον χρήστη για διαγραφή. Εάν ο χρήστης που θέλω να διαγράψω ταυτίζεται με τον χρήστη που κάνει την διαγραφή ή δεν υπάρχει, επιστρέφω μήνυμα λάθους. Ειδάλλως, αφαιρώ τον χρήστη από την λίστα.
* Void searchUser(List<Users> users): Μέθοδος για αναζήτηση ενός χρήστη. Η αναζήτηση γίνεται όπως και στις προηγούμενες μεθόδους. Έπειτα εκτυπώνεται το όνομα, το username και ο ρόλος του χρήστη. Εάν ο χρήστης δεν βρεθεί, εκτυπώνεται μήνυμα λάθους.
* Void viewAllUsers(List<Users> users): Μέθοδος για την προβολή στοιχείων όλων των χρηστών. Για κάθε user στην λίστα users εκτυπώνεται το όνομα, το username και ο ρόλος.
  1. **Κλάση ContentAdmins**

Η κλάση ContentAdmins κληρονομεί την κλάση Users, συνεπώς εμπεριέχει τις private μεταβλητές **name, username, password, loggedIn**. Επιπλέον κάνει import την κλάση Cinemas του package cinema. Όσον αφορά τις μεθόδους, υλοποιούνται οι προσαρμοσμένες για τον διαχειριστή περιεχομένου *login* και *logout*, τύπου void, με ένα μήνυμα στην οθόνη, καθώς και η *showUserMenu,* τύπου void, που δείχνει τις λειτουργίες που μπορεί να εκτελέσει ο διαχειριστής περιεχομένου.

Η ContentAdmins περιέχει επιπλέον τις μεθόδους *insertFilm, deleteFilm, assignFilmToCinema* τύπου void*.*

* Void insertFilm(List<Films> films): Μέθοδος για την προσθήκη νέας ταινίας. Δημιουργώ το αντικείμενο film της films και με την χρήση του αντικειμένου reader της BufferedReader διαβάζω τα δεδομένα που εισάγει ο χρήστης και κάνω set τις μεταβλητές της film, εφόσον δεν προκύψει κάποιο σφάλμα. Έπειτα εισάγω το αντικείμενο που δημιούργησα στην λίστα films.
* Void deleteFilm(List<Films> films):Μέθοδος διαγραφής μίας ταινίας. Ζητείται από τον χρήστη να εισάγει το ID της ταινίας προς διαγραφή. Με την χρήση του αντικειμένου reader της BufferedReader διαβάζω το ID. Εφόσον δεν υπάρχουν σφάλματα, αναζητώ μέσα στην λίστα films το αντικείμενο με το δοθέν ID. Εάν δεν υπάρχει, εκτυπώνω μήνυμα λάθους, αλλιώς διαγράφω την ταινία από την λίστα.
* Void assignFilmToCinema(List<Films> films, Cinemas cinemas[], List<Provoles> provoles): Εκτυπώνω στον χρήστη τα διαθέσιμα σινεμά και του ζητάω να εισάγει το id εκείνου που θέλει. Εφόσον δεν υπάρξουν σφάλματα και βρω το σινεμά, εκτυπώνω και τις διαθέσιμες ταινίες και του ζητάω να εισάγει το id εκείνης που θέλει. Εφόσον δεν προκύψει σφάλμα και βρω την ταινία, δημιουργώ το αντικείμενο provoli της κλάσης Provoles από την ταινία και τον κινηματογράφο που βρήκα και το προσθέτω στην λίστα.
  1. **Κλάση Cinemas**

Η κλάση Cinemas ανήκει στο package cinema μαζί με τις Films, Provoles. Περιέχει τις private μεταβλητές **cinemaName, cinemaNumberOfSeats, cinemaId, cinemaIs3d**. Όσον αφορά τις μεθόδους, περιέχει τα *public setters και getters*, καθώς και 2 *constructor*, ένα default και ένα με παραμέτρους.

* 1. **Κλάση Films**

Η κλάση Films ανήκει στο package cinema μαζί με τις Cinemas, Provoles. Περιέχει τις private μεταβλητές **filmId, filmTitle, filmCategory, filmDescription**. Όσον αφορά τις μεθόδους, περιέχει τα *public setters και getters*, καθώς και 2 *constructor*, ένα default και ένα με παραμέτρους.

* 1. **Κλάση Provoles**

Η κλάση Provoles ανήκει στο package cinema μαζί με τις Cinemas, Films. Περιέχει τις private μεταβλητές **provoliId, provoliNumberOfReservations, numberOfSeatsLeft, provoliFilm, provoliCinema, provoliStartDate, provoliEndDate, provoliIsAvailable**. Όσον αφορά τις μεθόδους, περιέχει τα *public setters και getters*, καθώς και *1 constructor,* το οποίο δέχεται ένα αντικείμενο cinema και ένα αντικείμενο film των αντίστοιχων κλάσεων και θέτει τις αντίστοιχες μεταβλητές.

* 1. **Κλάση MainClass**

Η κλάση MainClass είναι η κύρια κλάση του project. Περιέχει τις private static String μεταβλητές name, username, password καθώς και τις λίστες users, films, cinemas, provoles των αντίστοιχων κλάσεων. Περιέχει τις μεθόδους showLogin, cinemaTheaters , loadUsers, saveUsers τύπου void και handleLogin τύπου Users.

* private static showLogin(BufferedReader reader): Μέθοδος για login στην εφαρμογή. Ζητά από τον χρήστη να εισάγει τα name, username και password του. Ελέγχει για σφάλματα στην εισαγωγή των δεδομένων και εάν όλα τα πεδία πληρούν τα κατάλληλα κριτήρια κάνει set τις μεταβλητές name, username, password.
* Private static Users handleLogin(String name, String username, String password): Εάν ο κωδικός είναι “ImContentAdmin123” ή “ImAdmin123” τότε δημιουργείται χρήστης και θα έχει τον ρόλο content admin ή admin αντίστοιχα. Εάν το password δεν είναι κάποιο από αυτά, ο χρήστης που θα δημιουργηθεί θα έχει τον ρόλο του πελάτη. (Το password των admins μπορεί να αλλάξει μέσω της update users).
* Private static void cinemaTheaters(): Μέθοδος για την δημιουργία αντικειμένων της Cinemas που θα μπουν στην λίστα cinemas.
* Private void loadUsers():
* Private void saveUsers():
* Public static void main(String[] args):

Πρώτα καλείται η συνάρτηση cinemaTheaters, δημιουργείται ένα reader και καλείται η showLogin. Σε συνδυασμό με την handleLogin δημιουργείται ο χρήστης- χειριστής της εφαρμογής.

Έπειτα ανάλογα με τον ρόλο του χρήστη εκτυπωνεται το μενού