|  |  |  |
| --- | --- | --- |
| 기능명세서 | 문서번호 | AD-003 |
| 작성자 | 1조 |
| 작성일자 | 19.03.12 |

|  |  |  |
| --- | --- | --- |
| **구분** | **이름** | **역할** |
| **패키지** | **kr.co.sist.cinema.initialscreen** | **첫화면,로그인,회원가입,아이디찾기,비밀번호찾기** |
| **클래스** | **InitialScreen** | **첫화면,로그인,회원가입,아이디찾기,비밀번호찾기** |
| **메소드** | InitialScrren() | 생성자로써 멤버변수 초기화 |
| static String getId() | 회원 아이디 정보 얻어옴 |
| void void initial() | 초기화면 구현 |
| void pwFind() | 비밀번호 찾기 |
| void idFind() | 아이디 찾기 |
| void loginMenu() | 회원 로그인 기능 |
| void unlogin() | 비회원 로그인 |
| void signIn() | 회원 가입 |
| void initialWriter(String fileName) | 데이터 추가하기 |
| getList() | 파일데이터를 불러와 ArrayList에 데이터를 저장 |
| **클래스** | **GeneralMenu** | **회원, 비회원 메뉴** |
| **메소드** | GeneralMenu(String id) | GeneralMenu 클래스 생성자, 접속 아이디를 받음 |
| void generalMenuSelect() | 처음 메뉴화면 구성 |
| **클래스** | **FileRead** | **파일 데이터 경로를 받아 ArrayList에 파일 데이터를 입력** |
| **메소드** | FileRead() | FileRead 생성자, ArrayList와 Scanner 변수 초기화 |
| void initialReader(String fileName) | 매개변수 파일명을 받아와서 해당 파일 데이터를  ArrayList에 입력 |
| void pause() | 화면을 보여줄 때 다음 화면으로 바로 넘어가는 것을 방지 |
| **클래스** | **UImanagerLogin** | **관리자 메뉴** |
| **메소드** | static void begin() | 관리자 메뉴 창 띄움 |
| **클래스** | **UImemberLogin** | **초기 화면 메뉴** |
| **메소드** | static void begin() | 초기 화면 창 띄움 |

|  |  |  |
| --- | --- | --- |
| **구분** | **이름** | **역할** |
| **패키지** | **kr.co.sist.cinema.main** | **해당 소스들을 실행시킬 main 메소드가 들어있는 패키지** |
| **클래스** | **InitialScreenMain** | **초기화면** |
| **메소드** | static void main() | 실행시킴 |

|  |  |  |
| --- | --- | --- |
| **구분** | **이름** | **역할** |
| **패키지** | **kr.co.sist.cinema.manager** | **관리자가 실행하는 기능들의 집합** |
| **클래스** | **CardManage** | **카드 제휴 메뉴** |
| **메소드** | void CardManageStart() | 카드 제휴 메뉴 + 해당 메뉴로 이동 |
| void cardView() | 카드 제휴 정보 읽기 |
| void cardAdd() | 카드 제휴 정보 추가 |
| void cardInfoModify() | 카드 제휴 정보 수정 |
| void pause() | 카드 메뉴로 다시 돌아갈 수 있는 기능 |
| **클래스** | **MemberManage** | **회원 관리** |
| **메소드** | void memberManagestart() | 회원관리 메뉴 |
| void memberDelete() | 회원 삭제 |
| void memberView() | 회원 목록 보기 |
| void memberSearch() | 회원 검색 |
| void numberViews() | 회원 등급 업데이트 |
| **클래스** | **NoticeBoard** | **공지사항, FAQ 관리** |
| **메소드** | NoticeBoard() | 생성자 |
| void NoticeBoardstart() | 공지사항, FAQ 선택 메뉴 |
| void notice() | 공지사항 관리 |
| void questions() | 문의사항 관리 |
| void noticeView() | 공지사항 목록보기 |
| void questionView() | FAQ 목록보기 |
| void noticeBoardWriting(String fileName) | 공지사항 또는 FAQ 글쓰기 |
| void noticeBoardModify(String fileName) | 공지사항 또는 FAQ 글 수정 |
| void noticeBoardDelete(String fileName) | 공지사항 또는 FAQ 글 삭제 |
| **클래스** | **UIcard** | **카드관리 메뉴 꾸밈** |
| 메소드 | static void begin() | 카드관리 메뉴 |
| static void menu() | 지점 선택 |
| **클래스** | **UImember** | **멤버관리 메뉴 꾸밈** |
| 메소드 | static void begin() | 멤버관리 메뉴 |
| static void menu() | 지점 선택 |
| **클래스** | **UInotice** | **공지사항/문의사항 관리 메뉴 꾸밈** |
| 메소드 | static void begin() | 공지사항/문의사항 관리 메뉴 |
| static void menu() | 지점 선택 |

|  |  |  |
| --- | --- | --- |
| **구분** | **이름** | **역할** |
| **패키지** | **kr.co.sist.cinema.replaycinema** | **영화 다시보기 기능** |
| **클래스** | **ChargedReplay** | **유료 영화 다시보기** |
| **메소드** | ChargedReplay() | 생성자 |
| void chargedReplayCinemaList() | 유료영화 목록 |
| void replayPayment() | 영화선택 및 결제 |
| int dateCalculate(String date) | 날짜에 따른 유료영화 가격 계산 |
| **클래스** | **FreeReplay** | **무료 영화** |
| **메소드** | FreeReplay() | 생성자 |
| void freeReplayCinemaList() | 무료영화 목록 |
| void freeView() | 영화선택 및 시청 |
| **클래스** | **ReplayCinemaMain** | **영화 다시보기 메뉴** |
| **메소드** | void ReplayMenu() | 영화 다시보기 메뉴 |

|  |  |  |
| --- | --- | --- |
| **구분** | **이름** | **역할** |
| **패키지** | **kr.co.sist.cinema.reservation** | **예매 기능** |
| **클래스** | **AreaSelect** | **영**  **화지점 및 영화선택** |
| **메소드** | AreaSelect() | 초기화 생성자 |
| int getAreaNum() | 지점 번호 얻어옴 |
| void areaSelect() | 지점 선택 |
| void movieSelectScreen() | 영화 선택 |
| **클래스** | **CinemaSchedule** | **상영시간** |
| **메소드** | CinemaSchedule(String areaNum, String movieUnique, int movieTime) | 지점번호, 영화번호, 선택시간을 받는 생성자 |
| static int getTimeSelect() | 지정 시간 받아오기 |
| void time() | 예매 영화 정보 등록 |
| **클래스** | **Event** | **이벤트 적용** |
| **메소드** | Event(String cardType, int numberPeople, int pointUse) | 생성자, 카드종류, 사람수, 포인트사용 값을 받아서 초기화 |
| int getPointUse() | 사용포인트 값 받아옴 |
| int getOccurrencePoint() | 생성포인트 값 받아옴 |
| int getPrice() | 가격정보 받아옴 |
| void cardEvent(String areaNum) | 카드이벤트 적용 |
| void applyEvent(String areaNum) | 이벤트 적용 |
| void pointModify() | 해당 회원 포인트 수정 |
| int cultureDayCalculation() | 해당 달의 마지막 수요일 구하기 |
| **클래스** | **ScheduledSeat** | **지정된좌석 표시** |
| **메소드** | ScheduledSeat(String areaNum, String movieUnique, int hour, int minute, ArrayList seatNumberList, Calendar cal) | 지정된 좌석 정보를 넘겨주기 위한 데이터를 초기화 하는 생성자 |
| void scheduledSeatUpdate() | 해당좌석이 지정되있으면 ■ 표시됨 |
| void ticketInformation(Calendar calendar, int occurrencePoint, int pointUse, int price) | 예매한 티켓 정보 데이터에 입력 |
| **클래스** | **Seat** | **좌석 선택** |
| **메소드** | Seat() | 생성자 |
| ArrayList<String> getSeatNumberList() | 선택한 좌석 정보를 받아옴 |
| void SeatInfo(String areaNum, String movieUnique, int hour, int minute, Calendar cal, int numberPeople) | 좌석 정보를 나타냄 |
| **클래스** | **Ticketing** | **예매하기** |
| 메소드 | Ticketing() | 생성자 |
| int getNumberPeople() | 예매티켓 수 정보 받아옴 |
| EVNET getEvent() | 이벤트 정보 받아옴 |
| void paymentInformation() | 결제하기 |
| void cardTypeSelect(String areaNum) | 카드종류 선택 |
| void PointCheck() | 포인트 사용 유무 |
| void check(String areaNum) | 결제 확인 유무 |
| **클래스** | **UIreservation** | **지점선택 메뉴** |
| 메소드 | void menu() | 지점 선택 메뉴창 |

|  |  |  |
| --- | --- | --- |
| **구분** | **이름** | **역할** |
| **패키지** | **kr.co.sist.cinema.manager\_movieinfo** | **영화 추가 제거** |
| **클래스** | **MovieInfoManager** | **영화 추가 제거** |
| **메소드** | void menu() | 영화 정보 관리 메뉴 선택 |
| void movieInfoAdd() | 영화 정보 추가 |
| void movieDelete() | 영화 정보 삭제 |
| **클래스** | **Movie** | **영화 정보 저장, 가져오기** |

|  |  |  |
| --- | --- | --- |
| **구분** | **이름** | **역할** |
| **패키지** | **kr.co.sist.cinema.snackstore** | **매점 기능** |
| **클래스** | **SnackStore** | **매점시작, 물품선택, 물품결제** |
| **메소드** | SnackStore() | 생성자 |
| SnackStore(int areaNum) | 지역번호를 받아와서 초기화 하는 생성자 |
| void order() | 매점 주문 메소드 |
| void showProduct(ArrayList<String>  selectProduct) | 주문한 물품 보여주는 메소드 |
| void storePointCheck() | 물품 구입시 포인트 사용 및 결제 |
| void storePayMent(ArrayList<String> result) | 결제 내용을 데이터에 저장 |
| void pointUpdate(int pointUsed) | 사용한 포인트와 누적포인트 데이터에 저장 |
| **클래스** | **UI** | **회원, 비회원 메뉴** |
| 메소드 | UI() | 생성자 |
| void load() | 메뉴판 데이터 읽어옴 |
| void Storelist() | 메뉴판 표시 |
| **클래스** | **StoreMenu** | 메뉴의 각 정보를 읽어오고 저장 |

|  |  |  |
| --- | --- | --- |
| **구분** | **이름** | **역할** |
| **패키지** | **kr.co.sist.cinema.manager\_snackstore** | **매점 관리** |
| **클래스** | **FilePath** | **파일 경로** |
| **클래스** | **Snack** | **상품 정보 저장 및 읽어오기** |
| **클래스** | **SnackManager** |  |
| **메소드** | void load() | 메뉴 데이터 읽어오기 |
| void save() | 메뉴 데이터 추가 |
| void putItem() | 상품 추가 |
| void printItemList() | 상품 목록 출력 |
| void setCategory(ArrayList<Snack> categories, Snack snack) | 상품 추가 정보 입력 |
| void printCategory(ArrayList<Snack> categories) | 데이터에 있는 품목 종류 불러오기 |
| boolean isSole(String category, ArrayList<Snack> categories) | 데이터에 없는 정보인지 확인 |
| void restoreItem() | 품목 삭제 정보 입력 |
| void pullItem() | 품목 데이터에서 품목 삭제 |
| void manageSales() | 판매 현황 보여줌 |
| **클래스** | **Movie** | **영화 정보 저장, 가져오기** |
| 메소드 | void menu() | 매점 관리 메뉴 틀 |
|  | void putItemSubMenu() | 품목 추가 메뉴 틀 |
|  | void title(int n) | 품목 관리에서 실행하고자 하는 기능 제목 |
|  | void pause(int n) | 품목 관리 해당 기능 완료 표시 |
|  | void returnToMenu(int n) |  |

|  |  |  |
| --- | --- | --- |
| **패키지** | **kr.co.sist.cinema.movieinfo** | **영화 정보** |
| **클래스** | **FutureMovieInfoList** | **개봉 예정 영화** |
| **메소드** | FutureMovieInfoList() | 생성자 |
| void load() | 데이터에서 개봉 예정 영화 불러오기 |
| void movieList() | 영화 정보 데이터에서 개봉예정영화 불러오기 |
| void readFile(String listScan) | 개봉예정영화 목록 보여주기 |
| **클래스** | **FutureMovieInfoList** | **개봉 예정 영화** |

|  |  |  |
| --- | --- | --- |
| **패키지** | **package kr.co.sist.cinema.mypage** | **회원 관리** |
| **클래스** | **Member** | **회원 정보 데이터에서 각 데이터를 분류하여 저장하고 읽기** |
| **클래스** | **MyMembership** | **해당 회원의 멤버십 정보 열람** |
| **메소드** | MyMembership(String id) | 아이디를 초기화하는 생성자 |
| void info() | 간략한 멤버십 정보를 출력 |
| void menu() | [멤버십 정보] 페이지에서 선택가능한 메뉴를 보여주는 메소드 |
| void selectMenu() | 선택한 [멤버십 정보] 메뉴를 불러오는 메소드 |
| void selectRegisterCardMenu() | '카드가 존재(o) > [멤버십 카드 해지하기], 카드가 존재(x) > [멤버십 카드 등록하기]' 메뉴를 불러옴 |
| void registerCard() | 멤버십 카드 등록 |
| void registration() | 주어진 멤버십 카드 번호를 이용하여 정보 등록하기 |
| void completeRegistration(String input) | 멤버십 카드 등록 완료되었는지 보여줌 |
| boolean isValid(String input) | 유효한 멤버십 카드 번호인지 확인 |
| void cancelCard() | 삭제할 멤버십 카드 정보와 삭제 여부 |
| void completeDelete() | 멤버십 카드 삭제 진행 완료 여부 |
| String myCard() | 멤버십 카드 정보 보기 |
| void pointStatusMenu() | 포인트 현황 메뉴 UI |
| void selectPointStatusMenu() | 포인트 현황 메뉴에서 선택한 번호에 대한 기능 실행 |
| void pointDetails() | 포인트 내역 조회하기 |
| void pointDetailsMenu() | 포인트 현황 기능 선택 메뉴 UI |
| void selectPointDetailsMenu() | 포인트 현황 기능 메뉴에서 선택한 기능 실행 |
| void reservedPoint() | 적립된 포인트 내역 조회 |
| void UsedPoint() | 사용한 포인트 내역 조회 |
| void extinctPoint() | 소멸된 포인트 내역 조회 |
| void rsDescSort(ArrayList<Reservation> rs) | 포인트 내역을 내림차순으로 정렬 |
| String thisYear() | 해당 년도 값 읽기 |
| String point() | 해당 아이디의 포인트 값 읽기 |
| String dDay() | 12월 31일에 대한 틱값 읽기 |
| boolean extinct(int i) | 포인트가 생성된지 1년이 됬으면 true 아니면 false |
| String occuredPoint(int i) | 소멸되지 않은 포인트만 값으로 넘김 |
| String usedPoint(int i) | 사용한 포인트 정보 넘김 |
| String sortation(int i) | 포인트 현황 메뉴에서 선택한 메뉴에 대해 나타냄 |
| void expectedToPerish() | 포인트 소멸 예정일에 대한 정보를 설명 |
| void loadMember() | 회원정보 데이터 읽어옴 |
| void loadReservation() | 예약정보 데이터 읽어옴 |
| **클래스** | **MyPage** | **마이페이지** |
| **메소드** | MyPage(String id) | 아이디를 초기화하는 생성자 |
| void info() | 회원 정보를 출력하는 메소드 |
| void menu() | mypage에서 선택가능한 메인 메뉴를 보여주는 메소드 |
| void selectMenu() | 선택한 메인 메뉴를 불러오는 메소드 |
| void loadMember() | 회원정보 데이터에서 값 불러와서 저장 |
| **클래스** | **MyPerchaseHistory** | **물품 구매목록** |
| **메소드** | void perchaseLoad() | 물품결제 데이터에서 데이터를 읽어옴 |
| void perchaseResult() | 해당 회원의 물품구매 내역 보여줌 |
| **클래스** | **MyReservation** | **예매 내역** |
| **메소드** | void confirmAndCancel() | 해당 회원의 예매내역에 관한 정보 및 기능 선택 |
| void selectMenu() | 예매내역 메뉴에서 기능 선택 |
| void confirmDetails() | 예매내역 보여줌 |
| void cancel() | 예매 취소 |
| boolean isOneDayAgo(int i) | 예매한 티켓이 상영하기까지 하루가 안남았으면 false 하루이상 남았으면 true |
| void pastDetails() | 지난 예매 내역 확인 |
| void cancelDetails() | 취소 예매 내역 확인 |
| void notice() | 유의사항 메뉴 |
| void noticeTicketExchange() | 티켓 교환방법 유의사항 |
| noticeCancelAndRefund() | 티켓 취소 유의사항 |
| void noticeNote() | 관람 유의사항 |
| boolean isPast(int i) | 상영일시가 현재시간을 넘었는지 확인 |
| String sortation(int i) | 지난예매인지 취소예매인지 현재예매인지 구분 |
| void pause() | 다음 출력문으로 바로 이동하지 않고 출력문을 보여주기위함 |
| void loadReservation() | 예약정보 데이터 읽어옴 |
| **클래스** | **PerchaseData** | **구입물품 데이터에서 각 분류별로 데이터 저장 및 읽기** |
| **클래스** | **Reservation** | **예약 정보 데이터에서 각 데이터를 분류하여 저장하고 읽기** |

|  |  |  |
| --- | --- | --- |
| **구분** | **이름** | **역할** |
| **패키지** | **kr.co.sist.cinema.post.manager** | **포스트 관리** |
| **클래스** | **BannedWord** | **금지어 데이터에서 각 데이터를 분류하여 저장하고 읽기** |
| **클래스** | **BannedWordManager** | **금지어 관리** |
| **메소드** | void load() | 금지어 데이터 읽어옴 |
| void AddBannedWord() | 금지어 추가 |
| void save() | 금지어 데이터에 저장 |
| void print() | 금지어 목록 |
| **클래스** | **Post** | **포스트 데이터에서 각 데이터를 분류하여 저장하고 읽기** |
| **클래스** | **PostManager** | **포스트 관리** |
| **메소드** | void PostManagerStart() | 포스트 관리 기능 선택하여 실행 |
| void postView() | 포스트 목록 보기 |
| void postDelete() | 포스트 삭제 |
| **클래스** | **UIpost** | **포스트관리 메뉴 UI** |
| **메소드** | static void begin()throws IOException | 포스트 관리 메뉴 UI |

|  |  |  |
| --- | --- | --- |
| **구분** | **이름** | **역할** |
| **패키지** | **kr.co.sist.cinema.post.member** | **영화 포스트** |
| **클래스** | **UI** | **포스트 메뉴 UI** |
| **메소드** | void begin() | 무비포스트 시작 메세지 |
| void end() | 무비포스트 종료 메세지 |
| void menu() | 무비포스트 메뉴 |
| void myPostMenu() | 내 무비포스트 메뉴 |
| void title(int n) | 선택한 메뉴에 대한 이름 |
| void pause(int n) | 각 기능별로 다음출력문으로 바로 넘어가지 않게 잠시 멈춤 |
| **클래스** | **PostMain** | **영화 포스트 기능** |
| **메소드** | void PostWrite() | 포스트 기능 선택하기 |
| void load() | 포스트 데이터 읽어오기 |
| void deletePost() | 접속한 회원의 포스트 삭제 |
| void writePost() | 작성할 포스트 영화가 있는지 비교 |
| void postUpdate(String movieSerialNum, String rating) | 작성한 포스트 데이터에 입력 |
| setMovie(Post post) | 포스트를 쓸 영화 검색 |
| void setOtherThanMovie(Post post) | 포스트 작성하기 |
| void setHead(Post post) | 포스트 제목 작성 |
| void collectInput() | 포스트 내용 작성 |
| boolean isLineClean(String line, ArrayList<String> bannedWords) | 금지어가 있는지 비교 |
| boolean isAllClean(ArrayList<String> inputBody) | 금지어가 없으면 내용 누적 |
| void editPost() | 나의 포스트 내용 수정하기 |
| void printMyPost() | 내가 쓴 포스트 목록 |
| void save() | 포스트 데이터에 저장하기 |