| **기능명세서** | 문서번호 |  |
| --- | --- | --- |
| 작성자 | 4조 |
| 작성일자 | 22-04-17 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project** | **프로젝트** |
| **클래스** | **PrintList** | **리스트를 출력해주는 클래스** |
| **메서드** | myDMList() | 메시지 가능한 친구들을 출력해주는 메소드 |
| followingList() | 사용자가 팔로우한 친구들을 출력해주는 메소드 |
| eventList() | 진행중인 이벤트들을 출력해주는 리스트 |
| memberGroupLis() | 유저가 가입한 모임들을 출력해주는 리스트 |
| groupList() | 멤버가 가입하지 않은 모임들을 출력해주는 리스트 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.login** | **프로젝트** |
| **클래스** | **main** | **사용자가 실행했을 때 처음 뜨는 화면을 출력하는 클래스** |
| **메서드** | main() | 메인 메뉴 선택지들을 출력해주는 메소드 |
| **변수** | String input | 사용자로부터 이동하려는 메뉴의 숫자를 입력받은 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.login** | **프로젝트** |
| **클래스** | **Data** | **유저 텍스트 파일과 폴더생성에 관련된 클래스** |
| **메서드** | load() | 유저 텍스트 파일에 있는 내용들을 ArrayList로 옮겨주는 메소드 |
| save() | ArrayList안의 요소들을 다시 user.txt파일로 옮겨주고, 각 user별로 폴더를 생성시켜주는 메소드 |
| **변수** | String line | load메소드에서 파일 한줄한줄 루프문 돌게 만드는 변수 |
| String path | 각각의 유저별로, 유저아이디의 폴더가 생성되는 경로 |
| String path2 | 아이디 폴더 안에 info라는 파일로 유저 정보 저장하는 경로 |
| String path3 | 아이디 폴더 안에 review라는 파일로 리뷰 저장하는 경로 |
| String line | 통합된 user..txt.에 컬렉션에서 저장받은 유저 객체를 저장하는 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package.project.login** | **프로젝트** |
| **클래스** | **User** | **회원들의 정보가 담긴 클래스** |
| **메서드** | User() | 회원들의 정보를 초기화 시켜주는 생성자 |
| Set해당변수() | 사용자로부터 입력 받은 변수를 저장하는 메소드 |
| get해당변수() | 사용자로부터 입력 받은 변수를 가져오는 메소드 |
| **변수** | String id | 유저의 아이디를 저장하는 변수 |
| String password | 유저의 비밀번호를 저장하는 변수 |
| String name | 유저의 이름을 저장하는 변수 |
| String birth | 유저의 생년월일을 저장하는 변수 |
| String gender | 유저의 성별을 저장하는 변수 |
| String tel | 유저의 전화번호를 저장하는 변수 |
| String follow | 유저의 팔로우 여부를 저장하는 변수 |
| String genre | 유저의 장르를 저장하는 변수 |
| String school | 유저의 출신 초등학교를 저장하는 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package.project.login** | **프로젝트** |
| **클래스** | **UserLogin** | **사용자가 로그인 할 때 사용되는 클래스** |
| **메서드** | Login() | 사용자가 로그인 하는 화면을 출력하는 메소드 |
| findUser() | 사용자가 회원으로 로그인 하는 메소드 |
| LoginCategori(String id) | 회원으로 로그인 한 후 메뉴를 출력하는 메소드 |
| whoareu() | 가입되어 있지 않을 경우 출력해주는 메소드 |
| findMyId() | 아이디를 찾아주는 메소드 |
| findMyPassword() | 비밀번호를 찾아주는 메소드 |
| **변수** | String id | 사용자가 입력한 유저의 아이디를 저장하는 변수 |
| String password | 사용자가 입력한 유저의 비밀번호를 저장하는 변수 |
| int userLoginCount | 회원이 로그인 하는 횟수를 저장하는 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package.project.login** | **프로젝트** |
| **클래스** | **SignUp** | **사용자가 회원가입 할 때 사용되는 클래스** |
| **메서드** | addMember() | 회원가입 창을 출력하고 각각의 변수를 입력받는 메소드 |
| IdCheck() | 아이디 유효성 검사 메소드 |
| passwordCheck() | 비밀번호 유효성 검사메소드 |
| tellCheck() | 전화번호 유효성 검사메소드 |
| nameCheck() | 이름 유효성 검사메소드 |
| birthCheck() | 생년월일 유효성 검사 메소드 |
| genderCheck() | 성별유효성 검사 메소드 |
| followerCheck() | 팔로워 여부 유효성 검사 메소드 |
| genreCheck() | 장르 유효성 검사메소드(장르 의외 다른 단어가 들어오지 않도록 해주는 메소드) |
| genreCheck2() | 장르 유효성 검사메소드2(장르가 중복으로 들어오지 않도록 해주는 메소드) |
| **변수** | String id | 사용자가 입력한 유저의 아이디를 저장하는 변수 |
| String password | 사용자가 입력한 유저의 비밀번호를 저장하는 변수 |
| String password | 회원이 로그인 하는 횟수를 저장하는 변수 |
| String name | 유저의 이름을 저장하는 변수 |
| String birth | 유저의 생년월일을 저장하는 변수 |
| String gender | 유저의 성별을 저장하는 변수 |
| String tel | 유저의 전화번호를 저장하는 변수 |
| String follow | 유저의 팔로우 여부를 저장하는 변수 |
| String genre | 유저의 장르를 저장하는 변수 |
| String school | 유저의 학교를 저장하는 변수 |
| User u | 위에서 받은 유저의 정보들을 유저클래스의 유저객체에 저장하는 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package.project.login** | **프로젝트** |
| **클래스** | **Output** | **초기화면의 레이아웃을 담은 클래스** |
| **메서드** | title() | 초기화면에 나오는 그림을 출력하는 메소드 |
| findUser() | 초기화면에 나오는 메뉴를 출력하는 메소드 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package.project.login** | **프로젝트** |
| **클래스** | **nonUserLogin** | **비회원으로 들어올 경우 출력되는 화면을 담은 클래스** |
| **메서드** | nonUserLogin() | 비회원으로 입장하였음을 출력하는 메소드 |
| nonUserMain | 비회원 모드의 메인 메뉴를 출력하는 메소드 |
| nonUserCategory | 메인메뉴에서 카테고리를 입력하면 출력되는 메뉴 메소드 |
| **변수** | nonUserCount | 비회원들의 로그인 횟수를 저장하는 변수 |
| answer | 사용자가 들어고 싶은 메뉴번호를 입력받는 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain** | **회원 메인 메뉴** |
| **클래스** | **Output** |  |
| **메서드** | swearword(String input) | 문자열을 받으면 금지어 필터링을 마친 후 문자열을 반환해주는 메소드 |
| groupComment(String str) | 문자열을 받으면 문자열에 맞는 메뉴를 출력해주는 메소드 |
| groupExitMain() | 그룹 탈퇴 메뉴의 선택지를 출력 해주는 메소드 |
| groupJoinMain() | 그룹 가입 메뉴의 선택지를 출력 해주는 메소드 |
| groupMain() | 그룹 메인 메뉴의 선택지를 출력해주는 메소드 |
| memberMain() | 유저 메인메뉴의 선택지를 출력해주는 메소드 |
| subtitle(String title) | 문자열을 받아 타이틀을 출력해주는 메소드 |
| Subtitle1(String title) | 문자열을 받아 타이틀을 출력해주는 메소드 |
| pause() | 프로그램을 일시 정지 해주는 메소드 |
| line(int num) | num의 개수 만큼 줄을 출력해주는 메소드 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.group** | **회원 그룹** |
| **클래스** | **GroupMain** | **모임 메뉴** |
| **메서드** | groupMain() | 모임 메뉴 선택지를 출력해주는 메소드 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.group** | **회원 모임** |
| **클래스** | **GroupConnect** | **모임 접속** |
| **메서드** | groupConnect() | 가입된 모임 선택지를 출력해주는 메소드 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.group** | **회원 모임** |
| **클래스** | **GroupWork** | **모임 관련 처리 클래스** |
| **메서드** | GroupWork() | GroupWork클래스 생성자 |
| groupSelectMenu() | 모임 게시물의 선택지 출력 메소드 |
| groupExitConfirm() | 모임 탈퇴 확인 메소드 |
| groupJoinconfirm() | 모임 가입 확인 메소드 |
| groupSelect() | 모임 선택 메소드 |
| alterPost(int c) | ‘c’ 게시물을 수정하는 메소드 |
| adminAlterPost(int c) | 관리자 게시물 수정 메소드 |
| deletePost(int c) | ‘c’ 게시물을 삭제하는 메소드 |
| adminDeletePost(int c) | 관리자 게시물 삭제 메소드 |
| deleteOrAlterPost() | 게시물을 삭제/수정 선택하는 메소드 |
| createPost() | 게시물 생성 메소드 |
| writeComment() | 게시물 댓글 작성 메소드 |
| readPost() | 게시물 읽기 메소드 |
| **변수** | Int num | 선택한 모임을 저장하는 변수 |
| Int input | 선택한 게시물을 저장하는 변수 |
| String path | 모임 파일 경로를 저장하는 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.group** | **회원 모임** |
| **클래스** | **GroupJoin** | **모임 가입 클래스** |
| **메서드** | groupJoin | 회원 모임 가입 진행 메소드 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.group** | **회원 모임** |
| **클래스** | **GroupExit** | **모임 탈퇴 클래스** |
| **메서드** | groupExit | 모임 탈퇴 메소드 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.event** | **이벤트** |
| **클래스** | **EventMain** | **이벤트 메뉴** |
| **메서드** | eventMain() | 이벤트 메뉴 선택지 출력 메소드 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.event** | **이벤트** |
| **클래스** | **EventWork** | **이벤트 관련 처리 클래스** |
| **메서드** | eventSelect() | 선택한 이벤트 내용 출력 메소드 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.follow** | **팔로우** |
| **클래스** | **FollowMain** | **친구 메뉴** |
| **메서드** | followMain(); | 친구 메뉴 선택지 출력 메소드 |
| **변수** | String input | 친구 메뉴 선택지 저장 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.follow** | **팔로우** |
| **클래스** | **ManageFollow** | **팔로우 관리** |
| **메서드** | mangaeFollow() | 친구 관리 메소드 |
| **변수** | String input | 친구 관리 선택지 저장 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.follow** | **팔로우** |
| **클래스** | **DeleteFollow** | **친구 삭제** |
| **메서드** | deleteFollow | 친구 삭제 메소드 |
| **변수** | String user1 | 삭제할 친구 이름 저장 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.follow** | **팔로우** |
| **클래스** | **DMconfirm** | **DM(메세지) 확인** |
| **메서드** | dmConfirm | DM(메시지) 선택 출력하는 메소드 |
| **변수** | String input | DM(메시지) 선택값 저장하는 변수 |
| String intput1 | 대화할 상대(ID) 저장하는 메소드 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.follow** | **팔로우** |
| **클래스** | **userChat** | **DM(메시지) 채팅** |
| **메서드** | userChat(String input) | 대화할 상대 아이디(input)을 받은 후 입력 받은 메시지를 전송하는 메소드 |
| printChat(File[] files, int i) | 대화내역이 있는 파일을 받은 후 대화 내용을 출력하는 메소드 |
| **변수** | String id | 유저 아이디를 저장하는 변수 |
| String path | DM(메시지)의 파일경로를 저장하는 변수 |
| String message | DM(메시지)입력 값을 저장하는 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.managerMode** | **관리자** |
| **클래스** | **Admin\_Main** | **관리자 main 화면** |
| **메서드** | adminSel1() | 관리자 카테고리 관리 중 종류 선택지 출력 메소드 |
| adminSel1\_1(String path) | 관리자 카테고리 관리 중 행동 선택지 출력 메소드 |
| adminSel2() | 관리자 모임 관리 선택지 출력 메소드 |
| adminSel3() | 관리자 이벤트 관리 선택지 출력 메소드 |
| adminSel4() | 관리자 회원계정 관리 출력 메소드 |
| **변수** | String input | 사용자 선택지 저장 변수 |
| String moviePath | 영화 관리자 입력 리스트 txt 주소 변수 |
| String playPath | 연극 관리자 입력 리스트 txt 주소 변수 |
| String musicalPath | 뮤지컬 관리자 입력 리스트 txt 주소 변수 |
| String bookPath | 책 관리자 입력 리스트 txt 주소 변수 |
| String displayPath | 관람 관리자 입력 리스트 txt 주소 변수 |
| String moimPath | 모임 관리 폴더 주소 변수변수 |
| String listPath | 모임 이름 리스트 txt 주소 변수 |
| String eventPath | 이벤트 리스트 txt 주소 변수 |
| memberPath | 회원 리스트 txt 주소 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.managerMode** | **관리자** |
| **클래스** | **AdminCommunity** | **관리자 커뮤니티 관리** |
| **메소드** | read(String path) | 작품 리스트 출력 메소드 |
| add(String path) | 새로운 작품 추가 메소드 |
| delete(String path) | 작품 삭제 메소드 |
| ban(String path) | 비속어가 속한 한줄평 리스트 변환,출력 메소드 |
| **변수** | String titleName | 새로 입력할 작품 제목 저장하는 변수 |
| String dirName | 새로 입력할 작품 감독 이름 저장하는 변수 |
| String actName | 새로 입력할 작품 배우 이름 저장하는 변수 |
| String playTime | 새로 입력할 작품 상영기간 저장하는 변수 |
| String evaluate | 새로 입력할 작품 평점 저장하는 변수 |
| String line | 작품 리스트 한 줄 단위 저장 변수 |
| String temp | 작품 리스트 모든 제목 저장 변수 |
| String input | 사용자 입력값 저장하는 변수 |
| int index | 작품 리스트의 저장된 작품 개수 저장 변수 |
| int sel | 사용자 입력값 저장하는 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.managerMode** | **관리자** |
| **클래스** | **AdminEvent** | **관리자 이벤트 관리** |
| **메소드** | read(String path) | 이벤트 리스트 읽기 메소드 |
| add(String path) | 이벤트 추가 메소드 |
| delete(String path) | 이벤트 삭제 메소드 |
| adminEventSelect() | 이벤트 선택 메소드 |
| adminEventView() | 선택된 이벤트 내용 출력 메소드 |
| adminDeleteEvent(int c) | 이벤트 삭제/수정 메소드 |
| deleteOrAlterEvent() | 삭제/수정 이벤트 선택 받는 메소드 |
| adminAlterEvent(int c) | 이벤트 수정 메소드 |
| createEvent() | 이벤트 추가 메소드 |
| slang(String input) | 금지어 확인 메소드 |
| **변수** | String line | 리스트 한 줄씩 저장하는 변수 |
| int n | 리스트 한 줄씩 번호 표시하는 변수 |
| String titleName | 추가할 이벤트 제목 저장하는 변수 |
| String eventContent | 추가할 이벤트 내용 저장하는 변수 |
| String line | 이벤트 리스트 한 줄 단위 저장 변수 |
| String input | 사용자 입력값 저장하는 변수 |
| int index | 이벤트 리스트의 저장된 작품 개수 저장 변수 |
| int firstNum | 수정할 구간 첫번째 번호 저장 변수 |
| int secondNum | 수정할 구간 두번째 번호 저장 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.managerMode** | **관리자** |
| **클래스** | **AdminMember** | **관리자 회원 관리** |
| **메소드** | findMem() | 회원의 아이디를 검색하여 정보를 출력하는 메소드 |
| blackList() | 블랙리스트를 출력하는 메소드 |
| addBlackList() | 블랙리스트에 유저를 추가하는 메소드 |
| logGraph() | 로그인과 비 로그인의 로그인 횟수 통계를 출력하는 메소드 |
| **변수** | String input | 사용자의 입력을 받는 변수 |
| User result | 일치하는 아이디의 객체를 저장하는 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.managerMode** | **관리자** |
| **클래스** | **BlackList** | **블랙리스트 유저의 정보를 담은 클래스** |
| **메소드** | BlackList() | 블랙리스트의 생성자 메소드 |
| set해당변수() | 사용자로부터 입력받은 매개변수를 블랙리스트 유저에 저장하는 메소드 |
| get해당변수() | 블랙리스트의 정보를 받아오는 메소드 |
| **변수** | String bid | 블랙리스트의 아이디를 저장하는 변수 |
| String bpassword | 블랙리스트의 비밀번호 를 저장하는 변수 |
| String bbirth | 블랙리스트의 생년월일을 저장하는 변수 |
| String bgender | 블랙리스트의 성별을 저장하는 변수 |
| String btel | 블랙리스트의 전화번호를 저장하는 변수 |
| String bfollow | 블랙리스트의 팔로우를 저장하는 변수 |
| String bgenre | 블랙리스트의 장르를 저장하는 변수 |
| String bschool | 블랙리스트의 학교를 저장하는 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.managerMode** | **관리자** |
| **클래스** | **BlackListData** | **블랙리스트 파일을 컬렉션에 옮겨주고 돌려주는 클래스** |
| **메소드** | loadBlacklist() | 블랙리스트 텍스트 파일을 ArrayList로 옮겨주는 메소드 |
| saveBlacklist | ArrayList의 요소들을 블랙리트스 텍스트파일로 옮겨주는 메소드 |
| **변수** | String line | load메소드에서 파일의 한줄한줄을 읽어주는 루프문을 돌려준다 |
| String str | save메소드에서 사용자로부터 입력받은 블랙리스트의 정보를 저장해준다 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.managerMode** | **관리자** |
| **클래스** | **AdminMoim** | **관리자 모임 관리** |
| **메소드** | readMoim() | 모임 리스트 읽어주는 메소드 |
| addDelManu() | 모임 2번째 페이지 출력 메소드 |
| addMoim(String listPath, String path) | 모임 창설 메소드 |
| deleteMoim(String listPath, String path) | 모임 삭제 메소드 |
| manage(String listPath, String path) | 모임 게시물/댓글 수정 메소드 |
| **변수** | String line | 파일 읽기에 사용하는 변수 |
| int index | 모임 출력 시 인덱스 번호 저장하는 변수 |
| String input | 사용자 입력 저장하는 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.myPage** | **마이페이지** |
| **클래스** | **MyPageMain** | **마이페이지 메인화면** |
| **메서드** | myPageMain() | 마이페이지 메뉴 출력하는 메서드 |
| **변수** | String userId | 유저 아이디를 저장하는 변수 |
| Calendar c | 현재 날짜를 입력받기 위한 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.myPage** | **마이페이지** |
| **클래스** | **MyInfomation** | **내 정보보기 메인화면** |
| **메서드** | MyInfo() | 내 정보 출력하는 메서드 |
| choiceMenu() | 정보보기 내의 메뉴를 선택하는 메서드 |
| newInfo() | 선택한 정보를 변경하는 메서드 |
| saveMessage() | 정보 변경, 저장 후 확인 메세지 출력 메서드 |
| chkGender() | 성별 유효성검사 |
| chkPhone() | 핸드폰번호 유효성검사 |
| **변수** | String userId | 유저 아이디를 저장하는 변수 |
| Calendar c | 현재 날짜를 입력받기 위한 변수 |
| String[] temp | 유저 정보 배열을 받아오는 메서드 |
| String input | 사용자 입력값 저장 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.myPage** | **마이페이지** |
| **클래스** | **MakeList** | **각종 txt파일을 배열로 변환하는 메서드** |
| **메서드** | String[] infoList() | info.txt 파일을 배열로 변환하는 메서드 |
| saveList(String[] temp) | infoList()메서드를 통해 변환한 배열을 info파일에 덮어쓰는 메서드 |
| makeReviewArray(ArrayList<UserReview> list) | UserReview클래스 배열에 요소를 추가하는 메서드 |
| makeUserList(ArrayList<User> list) | User 클래스 배열에 요소를 추가하는 메서드 |
| changeUserList(String changeInfo) | 수정된 user정보를 받아 info파일과 save파일을 수정하는 메서드 |
| **변수** | String userId | 유저 아이디를 저장하는 변수 |
| ArrayList<User> userList | user정보를 입력받는 배열을 저장하는 변수 |
| String path | 파일 경로를 저장하는 메서드 |
| String info | 사용자의 정보 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.myPage** | **마이페이지** |
| **클래스** | **UserReview** | **회원별 한줄평 정보가 담긴 클래스** |
| **메서드** | UserReview() | 리뷰 정보 초기화하는 변수 |
| get 해당변수() | 사용자로부터 입력 받은 변수를 저장하는 메소드 |
| set 해당변수() | 사용자로부터 입력 받은 변수를 가져오는 메소드 |
| **변수** | int yer | 리뷰한 작품을 감상한 연도 |
| int month | 리뷰한 작품을 감상한 월 |
| int date | 리뷰한 작품을 감상한 일 |
| String type | 감상한 작품의 종류(영화, 뮤지컬, 도서 등) |
| String title | 감상한 작품의 제목 |
| String review | 한줄평 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.myPage** | **마이페이지** |
| **클래스** | **Title** | **각종 고정화면을 출력해주는 클래스** |
| **메서드** | myPageTitle() | 마이페이지 메인선택화면 출력 |
| genreTitle() | 관심장르 목록 고정화면 출력 |
| reviewTitle() | 전체 글 목록보기 고정화면 출력 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.myPage** | **마이페이지** |
| **클래스** | **MyGenre** | **관심장르를 보고 수정하는 클래스** |
| **메서드** | favGenre() | 선택지 출력 메서드 |
| getNewGenre() | 수정할 장르를 입력받는 메서드 |
| genreString(String[] newGenre) | newGenre배열을 info텍스트 파일에 들어갈 형식에 알맞은 문자열로 변환 |
| **변수** | String path | 유저 정보가 담긴 txt파일의 경로를 저장하는 변수 |
| String input | 사용자 입력값 저장 |
| String[] newGenre | 새롭게 지정한 장르를 입력받는 배열 |
| String[] genres | 선택할 장르 목록이 들어있는 배열 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.myPage** | **마이페이지** |
| **클래스** | **MyGenre** | **관심장르를 보고 수정하는 클래스** |
| **메서드** | favGenre() | 선택지 출력 메서드 |
| getNewGenre() | 수정할 장르를 입력받는 메서드 |
| genreString(String[] newGenre) | newGenre배열을 info텍스트 파일에 들어갈 형식에 알맞은 문자열로 변환 |
| **변수** | String path | 유저 정보가 담긴 txt파일의 경로를 저장하는 변수 |
| String input | 사용자 입력값 저장 |
| String[] newGenre | 새롭게 지정한 장르를 입력받는 배열 |
| String[] genres | 선택할 장르 목록이 들어있는 배열 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.myPage** | **마이페이지** |
| **클래스** | **MyReviewCalendar** | **리뷰 기록이 달력과 연동되어 출력되도록 하는 클래스** |
| **메서드** | monthlyCalendar() | 리뷰 목록을 표시한 월간 달력 배열을 저장하는 메서드 |
| createCalendar(String[][]monthly, Calendar thisMonth) | monthlyCalendar() 가 저장한 배열을 달력 형식에 맞추어 출력하는 메서드 |
| lastDate(int month) | 매 월의 마지막 날을 반환하는 메서드 |
| startDate(int dayNum) | 매 월의 첫 요일이 배열에서 1로 입력되도록 설정하는 메서드 |
| calendarMenu() | 내 리뷰 보기 페이지의 선택지를 출력하고 입력받는 메서드 |
| **변수** | String path | 유저 정보가 담긴 txt파일의 경로를 저장하는 변수 |
| String input | 사용자 입력값 저장 |
| Calendar thisMonth | 입력받은 연, 월의 1일로 초기화하는 변수 |
| ArrayList<Integer>reviewDate | 입력받은 연, 월과 일치하는 UserReview의 날짜를 입력받는 배열 |
| int year | 캘린더 초기화하기 위한 연도 지정변수 |
| int month | 캘린더 초기화하기 위한 월 지정 변수 |
| String[] newGenre | 새롭게 지정한 장르를 입력받는 배열 |
| String[] genres | 선택할 장르 목록이 들어있는 배열 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.myPage** | **마이페이지** |
| **클래스** | **ReviewList** | **전체 리뷰 목록을 볼 수 있도록 하는 클래스** |
| **메서드** | allReviewList() | 전체 작성 리뷰 목록을 출력 |
| call() | 메뉴 선택 후 선택 후 선택지 불러오는 매서드 |
| **변수** | File reviews | review파일에 리뷰가 존재하는지 확인하기 위한 변수 |
| String userId | 해당 아이디의 reviews.txt파일 경로를 지정하기 위한 변수 |
| int year | 해당 연월의 myReviewCalenda 클래스를 재호출하기 위한 변수 |
| int month |
| ArrayList<UserReview>list | 사용자의 리뷰목록의 정보를 받아 배열로 저장하기 위한 변수 |
| String input | 사용자 입력값 저장 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.myPage** | **마이페이지** |
| **클래스** | **FindeReview** | **특정 날짜를 입력받으면 해당 날짜의 리뷰를 출력** |
| **메서드** | find() | 리뷰 추가 |
| boolean check(String) | 입력받은 날짜의 유효성검사. |
| noReviews() | review파일에 작성된 리뷰가 없을경우 안내문 반환 매서드 |
| **변수** | String input | 사용자 입력값(검색날짜)저장 |
| String[]ymd | 입력받은 날짜를 연, 월, 일 분리해서 배열에 저장 |
| int index | 입력받은 날짜에 기록이 있는지 확인하는 변수 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.myPage** | **마이페이지** |
| **클래스** | **DeleteID** | **회원 탈퇴하기** |
| **메서드** | quit() | 탈퇴 여부 입력받은 메서드 |
| delete() | 탈퇴위해 비밀번호와 본인확인질문 확인하는 메서드 |
| deleteUserFile() | user폴더와, user.txt파일의 회원 정보를 삭제하는 메서드 |
| **변수** | String input | 사용자 입력값 저장 |
| String[]ymd | 입력받은 날짜를 연, 월, 일 분리해서 배열에 저장 |
| ArrayList<User> userList | user.txt파일의 user정보를 배열로 가지고옴. |
| File infoFile | 회원탈퇴시 삭제해야할 회원 개인정보 데이터 |
| File reviewFile | 회원탈퇴시 삭제해야 할 작품 리뷰 데이터 |
| File userFolder | 회원탈퇴시 삭제해야 할 회원 폴더 |
| StringBuilder users | 변경할 user.txt 파일의 내용이 저장됨 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.community** | **커뮤니티** |
| **클래스** | **CommunityMain** | **커뮤니티 메뉴창** |
| **메서드** | commuMain() | 커뮤니티 메뉴창을 출력 및 카테고리 선택 |
| **변수** | String input | 사용자 입력값 저장 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.community** | **커뮤니티** |
| **클래스** | **CommunityUI** | **커뮤니티 UI** |
| **메서드** | communityMenu | 커뮤니티 메뉴창 선택지 출력 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.community** | **커뮤니티** |
| **클래스** | **CategorySetClass** | **카테고리의 기본값을 기입하는 클래스** |
| **변수** | String moviePath | 영화 데이터가 있는 경로 |
| String movieReview | 영화 리뷰 데이터가 있는 경로 |
| String playPath | 연극 데이터가 있는 경로 |
| String playReview | 연극 리뷰 데이터가 있는 경로 |
| String musicalPath | 뮤지컬 데이터가 있는 경로 |
| String musicalReview | 뮤지컬 리뷰 데이터가 있는 경로 |
| String bookPath | 도서 데이터가 있는 경로 |
| String bookReview | 도서 리뷰 데이터가 있는 경로 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.community** | **커뮤니티** |
| **클래스** | **CommunityMain** | **커뮤니티 메뉴창** |
| **메서드** | commuMain() | 커뮤니티 메뉴창을 출력 및 카테고리 선택 |
| **변수** | String input | 사용자 입력값 저장 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.community** | **커뮤니티** |
| **클래스** | **Community** | **카테고리 선택 시 해당 데이터를 불러오는 클래스** |
| **메서드** | movie(String input) | 영화의 경로 지정 및 Select 클래스로 이동 |
| play(String input) | 연극의 경로 지정 및 Select 클래스로 이동 |
| musical(String input) | 뮤지컬의 경로 지정 및 Select 클래스로 이동 |
| book(String input) | 도서의 경로 지정 및 Select 클래스로 이동 |
| categorieOutput(String category) | 상위 메소드에서 전달받은 값을 통해 해당 카테고리 경로를 지정받은 후 해당 카테고리의 메뉴창을 출력 및 선택 값을 return  회원용 비회원용 둘다 해당 메소드에서 진행 |
| fileRead(String path) | 상위 메소드에서 전달받은 카테고리 경로를 통해 각 값들을 CategorySetCalss에게 선언해주고 평점을 계산하여 오름차순 정렬한 값을 return |
| **변수** | String input | 카테고리 선택 값을 의미(번호) |
| String category | 카테고리 명 |
| String communityInput | 장르 선택에 있어 영향을 줄 수 있는 커뮤니티 입력값 |
| ArrayList<CategorySetClass> list | 선택한 카테고리의 데이터가 들어가있는 배열 |
| String Path | 선택한 카테고리의 데이터가 있는 경로 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.community** | **커뮤니티** |
| **클래스** | **Select** | **커뮤니티 총괄 메소드를 가지고있는 클래스** |
| **메서드** | genreSelect(String category, String communityInput, String input) | 전달받은 category, communityInput, input을 통하여 각 카테고리 별로 분류 후 하기 메소드로 이동하기 위한 메소드 |
| categoryCheck(String category, communityInput, String input, String methodName) | 회원, 비회원 카테고리 메뉴에서 선택지를 고르는 기능 |
| workSeach(String nameInput,, String category, String input) | 상위 메소드에서 전달받은 nameInput을 이용하여 데이터에서 해당 값이 들어가있는 작품 데이터 가져오기 기능 |
| tasteSelect(String input, ArrayList<String> userTasteList, String category) | 현재 접속한 유저의 관심장르 체크 및 장르 선택/새로운 장르 추천으로 넘어가기 위한 메소드 기능 |
| nowCategoryListCheck(Calendar now, ArrayList<String> nowCategoryList, String category) | 현재 상영중인 작품리스트를 출력하기 위한 메소드 |
| genreView(String input, String methodName, String genreSelectInput, String category) | 장르별 보기 기능을 구현하기 위한 메소드로 해당 값을 입력 후 장르의 TOP5, 작품 검색기능으로 넘어가기위한 메소드 |
| workSelect(String input, ArrayList<String> userTasteList, String methodName, String category) | 작품 검색 및 상위 메소드를 확인하여 되돌아가는 기능을 가진 메소드 |
| review(String name, String input, ArrayList<String> userTasteList, String methodName, String category) | 리뷰 경로를 확인하여 해당 작품의 리뷰를 확인하기 위한 메소드 |
| **변수** | String slangPath | 비속어 변환을 위한 데이터 경로 |
| String methodName | 상위 메소드로 돌아가기 위한 변수 |
| ArrayList<String> userTasteList | 접속한 유저의 관심 장르를 담는 배열 변수 |
| String category | 상위에서 받아온 카테고리 값 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.follow** | **팔로우** |
| **클래스** | **FollowAdd** | **Follow 추가 클래스** |
| **메서드** | followAdd() | Follow 추가를 위한 메뉴창 출력 및 값 입력 메소드 |
| followAddMethod(Scanner scan) | 상위 메소드에서 값을 받고 Follow할 아이디를 입력 및 예외처리 진행 메소드 |
| following(Scanner scan, String inputId, String userPath) | (Y/N)를 입력받은 값을 통해 본인과 상대 follow.txt파일에 각자의 아이디를 삽입하는 메소드 |
| openUser(String inputId) | 상대 user.txt파일을 열어 Follow 공개 여부를 체크한 후 return 해주는 메소드 |
| **변수** | String inputId | Follow할 상대 유저의 아이디 |
| userLogin.id | 본인 아이디 |
| String userPath | 본인 follow 경로 |

| **구분** | **이름** | **역할** |
| --- | --- | --- |
| **패키지** | **package project.memberMain.follow** | **팔로우** |
| **클래스** | **FollowCal** | **Follow한 유저의 캘린더 공유하는 클래스** |
| **메서드** | followCalMethod(Scanner scan) | 캘린더를 확인할 상대 유저 아이디 받는 메소드 |
| follower(String inputId) | 본인 following.txt파일을 확인하기 위한 메소드 |
| followCalMenu(String inputId) | 상대 캘린더를 확인 후 메뉴창 출력 및 이동을 위한 선택지 메소드 |
| call(String inputId) | 상위 메소드로 돌아가기 위한 메소드 |
| **변수** | String followerId | 본인 following.txt 경로 |
| String inputId | 상대 아이디 |