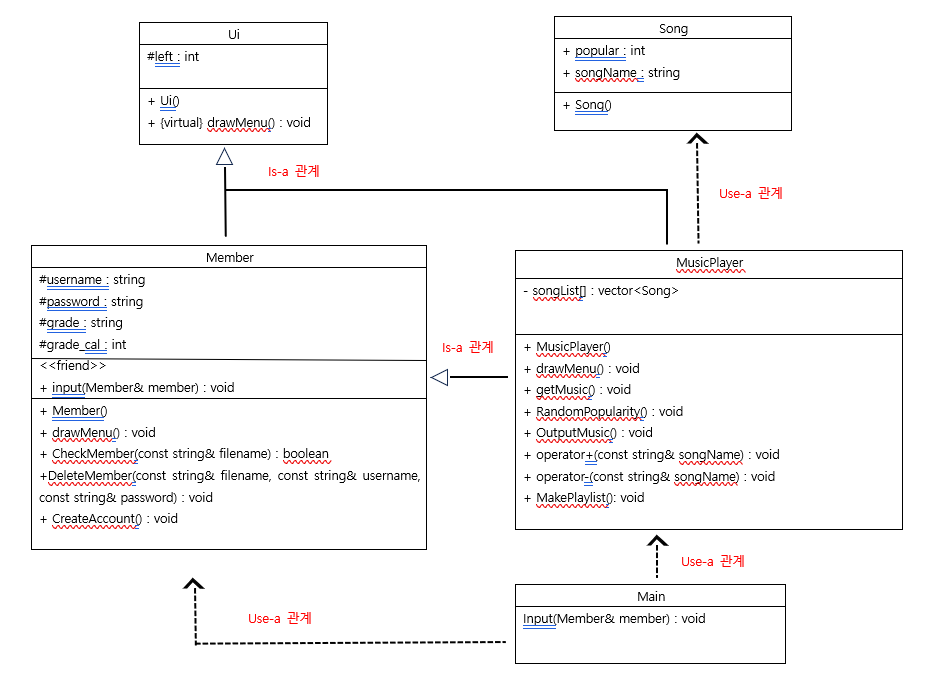
**객체지향 텀 프로젝트 중간보고서**

조원: 유동균, 유지원, 권태열, 이충훈

* 문제 명세

저희 팀의 프로젝트 주제는 음악 플레이어입니다. 음악 플레이어에 들어가는 기능으로는 플레이어를 이용하는 회원등록, 음악보기, 나만의 플레이리스트 만들기입니다.

회원등록 기능을 만들어 이용자가 로그인을 하도록 만듭니다. 또한 이용자에 등급을 매겨 이후 기능 사용에 차별점을 넣습니다. 음악보기 기능은 음악을 출력해 볼 수 있으며 없는 노래의 추가를 가능하도록 만듭니다. 나의 플레이리스트 기능은 플레이리스트를 만들어 노래를 추가하고 플레이리스트의 노래 모음을 조회 가능하며 노래를 삭제하는 기능까지 만듭니다.

* 분석

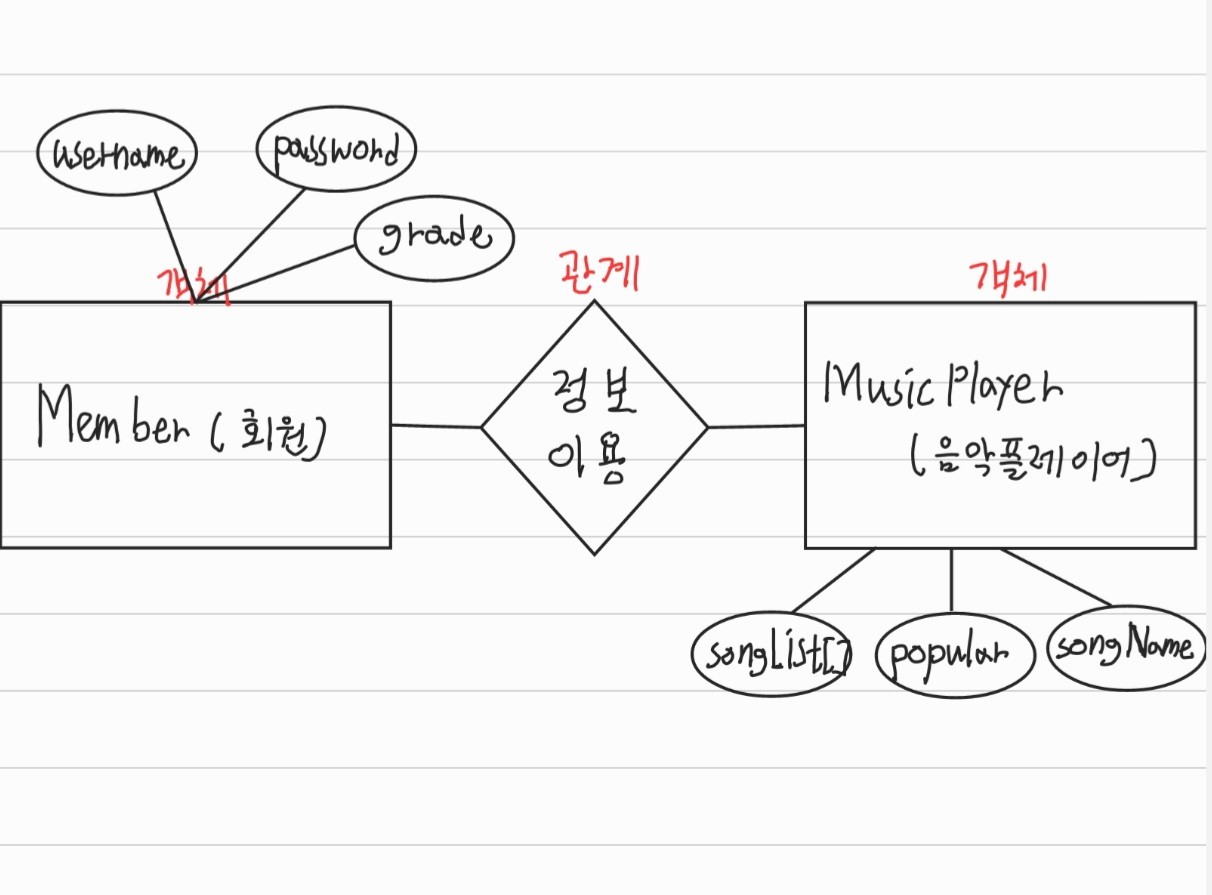
위의 그림과 같이 클래스의 계층구조를 위와 같은 UML로 표현할 수 있습니다.

class Ui는 콘솔창에서 화면을 출력하기 위한 클래스 부분이며 모든 화면에서 Ui를 사용하기 위해 추상 클래스로 만들었습니다. class Ui를 상속하여 두 개의 클래스에서 가상함수 drawMenu()를 사용합니다. 따라서 class Member와 class MusicPlayer에서 상속을 받습니다.   
 class Member는 등록한 회원의 아이디(username)와 비밀번호(password)를 멤버변수로 가지고 있는 클래스이며 여러 기능을 사용하기 위해 위와 같은 함수들을 사용합니다. 그리고 class Ui를 상속받아 Is-a관계로 사용합니다.

class Song은 멤버변수로 popular와 songName이라는 변수를 public로 가지고 있습니다. 이 class Song의 멤버변수들을 가지고 있는 클래스 배열을 생성하기 위해 class MusicPlayer에서 Use-a관계로 사용하고 있는 것입니다.

class MusicPlayer는 classs Member의 username변수를 사용하여 플레이리스트를 만들기 위해 상속을 받았고 따라서 Is-a관계를 사용하고 있습니다.

마지막으로 main()함수는 두 클래스 class Member, class MusicPlayer를 사용해 프로그램을 움직이기 때문에 Use-a관계를 사용했습니다.

객체 관계 다이어그램

IOPE

I

콘솔창에서 메뉴 로그인, 회원가입, 회원탈퇴, 종료 이렇게 4개의 입력을 받는다. 회원가입, 회원탈퇴는 아이디와 비번을 입력 받아 처리하고 로그인을 입력하면 다음으로 넘어간다. 그 다음의 메뉴는 플레이리스트, 노래보기 두 메뉴가 있으며

플레이리스트를 입력 시 플레이리스트를 확인하고 수정할 수 있으며 노래보기를 누르면 노래가 인기도의 순서대로 뜨게 된다.

메모장에서 회원의 정보(username, password), 노래의 정보(노래, 가수)를 읽어와 각 함수들의 입력으로 넣는다

O

로그인 성공 시에는 “로그인 성공”이 출력되며 이후 플레이리스트, 노래보기의 메뉴를 선택 가능하도록 출력한다. 플레이리스트를 고르면 아이디와 같은 이름의 메모장의 안 노래 내용을 출력한다. 노래보기를 고르면 노래메모장에 있는 노래들의 인기도가 랜덤으로 들어가 인기도 순으로 출력한다.

P

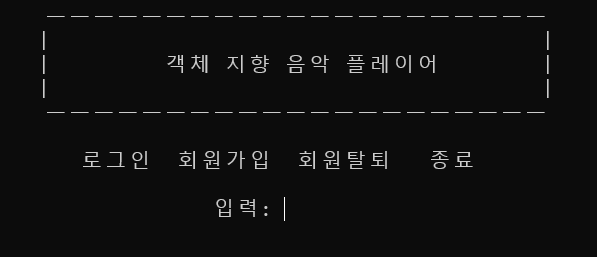
화면을 그리는 Ui는 Ui클래스와 drawMenu()를 통해 그릴 것이다. 사용자의 username과 password를 입력 받고 객체로 가지고 있을 Member클래스를 만들고 클래스의 멤버 변수들로 각각의 함수들을 진행한다. CheckMember(), DeleteUser(), CreateAccount() 등의 함수들은 멤버를 사용하고 login.txt라는 메모장에서 회원 정보를 가져와 로그인, 회원가입, 회원탈퇴를 작동한다. Song클래스는 노래의 제목과 인기도를 활용하기 위해 클래스 배열로 만들었다. Popular과 songName의 멤버를 가지고 있는 클래스 배열을 MusicPlayer클래스에서 vector를 활용한 객체 배열을 만든다. 그 객체 배열의 노래 제목들과 Member클래스의 username을 가지고 플레이리스트를 만든다.

E

회원가입을 해서 username: abc / password: 123을 메모장에 넣는다. 다시 abc와 123으로 로그인을 한다. 로그인 후 플레이리스트, 노래 보기를 고를 수 있다. 노래보기를 하면 랜덤으로 된 인기도를 가진 노래들이 인기도 순으로 출력된다. 플레이리스트를 입력하면 본인의 아이디로 된 플레이리스트를 만들고 노래이름을 입력받아 노래를 추가, 삭제 가능하다. 다시 로그인 했을 때 플레이리스트를 보고싶으면 자신의 아이디로 된 txt파일을 불러온다.

* 설계

클래스는 총 4개로 구성되어 있으며 서로 클래스를 사용하고 상속하며 상속 받습니다.

먼저 Ui클래스입니다. Ui클래스는 콘솔창에서 화면의 ui를 출력하는 함수들을 가지고 있는 클래스입니다. 옆의 그림과 같이 상단의 박스를 그리고 밑의 메뉴를 쉽게 그리기 위해 클래스로 Ui를 생성하게 되었습니다. 상단의 박스는 추상 클래스 Ui의 생성자 Ui()에 넣었고 밑의 메뉴의 출력은 drawMenu()함수를 가상함수 virtual로 만들어 재정의하고 동적 바인딩을 사용하도록 만들었습니다.

Member클래스는 사용자의 username(id), password(비밀번호)와 grade(등급을 나타내는 부분) grade\_cal(등급 계산을 위한 변수)로 멤버가 구성되어 있습니다. Ui클래스를 상속받아 ui출력을 사용할 수 있게 만들었습니다. drawMenu()를 재정의하여 메뉴를 그립니다.

CheckMember()함수는 클래스의 멤버인 username과 password와 같은 아이디와 비번이 있는지 파일을 한 줄 씩 읽으며 확인하면서 체크합니다. 같은 아이디와 비번이 있으면 true를 반환 없으면 false를 반환합니다. 이 함수를 활용해 로그인과 회원가입을 실행합니다. 로그인 시 메모장에 클래스의 아이디와 비밀번호가 있는지 확인하고 있다면 true를 반환해 로그인을 하며, false이면 로그인이 실패되고 true가 될 때까지 계속해서 아이디와 비번을 입력 받습니다. 입력 받는 부분은 외부함수 Input()함수를 사용합니다. 회원가입의 경우에는 반대로 겹치는 아이디와 비번이 없어야 함으로 CheckMember()함수가 false로 뜰 때 CreateAccout()함수에서 입력 받은 아이디와 비밀번호로 메모장에 새로운 회원정보, 회원가입을 할 수 있도록 합니다. 회원탈퇴는 DeleteMember()함수를 사용합니다. 삭제할 회원의 아이디와 비번은 회원가입과 같이 새로 아이디와 비번을 입력 받아야 하고 그 값을 메모장에서 삭제하는 방식으로 함수가 진행되게 만들었습니다.

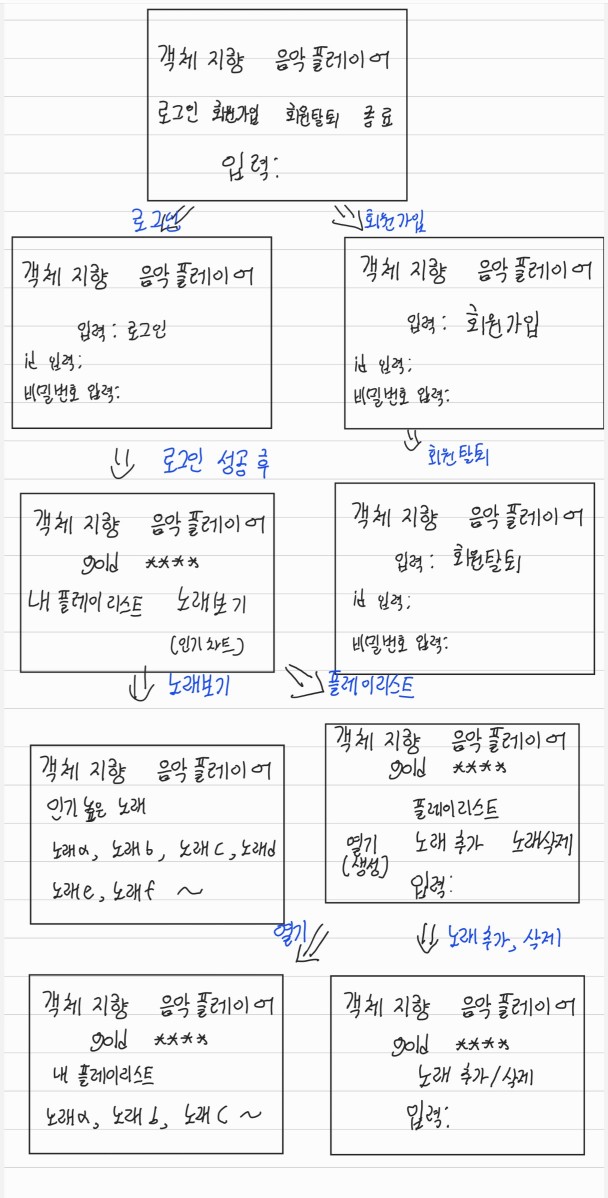
따라서 class안의 함수들은 현재 입력 받은 멤버 username과 password에 더 쉽게 접근하기 위해 클래스로 멤버와 메소드를 정의했습니다.

Song클래스는 popular과 songName을 멤버변수로 가지고 있습니다. 노래 메모장에 들어있는 제목을 가져오기 위해 songName로 변수를 받으며 popular 변수는 노래를 출력 시 인기도 순서대로 노래를 출력하기 위해 만든 변수입니다. 두 개(popular, songName) 멤버를 가진 클래스Song을 생성했고 MusicPlayer클래스에서 Song클래스를 활용한 클래스 배열 즉 노래 배열을 만들기 위해 Song클래스를 만들었습니다.

MusicPlayer클래스에서 Song클래스를 활용한 클래스배열을 vector을 통해서 생성합니다. 그러면 각 노래와 인기도를 가지는 클래스의 배열이 생성되며 각 노래에 접근이 가능합니다.

drawMenu()를 재정의하여 메뉴를 그립니다. getMusic()함수를 통해 노래 메모장에서 노래를 하나씩 가져와 클래스 배열의 songName에 넣습니다. RandomPopularity()함수를 사용해 1부터 5까지의 인기도를 랜덤으로 생성하여 클래스 배열의 popular에 넣습니다. 그리고 OutputMusic()함수를 통해 클래스배열의 노래 내용을 우선순위(popular)가 높은 순서대로 출력합니다. 노래를 인기도 높은 노래부터 순서대로 출력해 볼 수 있게 하기위해 popular 인기도 계산하는 부분을 넣었으며 클래스 배열을 이용했습니다.

MakePlaylist()함수를 사용해 플레이리스트를 만듭니다. 플레이리스트는 노래의 저장을 위해 메모장을 사용해 저장하도록 만들었고 때문에 각각의 사용자의 플레이리스트(메모장)의 구분을 위해 Member클래스의 username 변수를 상속받아 username으로 플레이리스트의 이름을 만들도록 했습니다.

플레이리스트에 노래의 추가는 연산자 중복(+),(-)을 활용하여 입력 받은 노래를 플레이리스트 메모장에 추가, 삭제가 가능 하도록 만들었습니다. 또한 사용자의 등급에 차별점을 두기위해 Member클래스의 grade도 가져와 gold일 경우 플레이리스트에 노래 10곡 넣기 가능, silver일 경우 플레이리스트에 노래 8곡 넣기 가능, bronze일 경우 플레이리스트에 5곡을 넣을 수 있도록 반복하도록 만들었습니다.   
이렇게 Member클래스와 MusicPlayer 클래스의 멤버변수와 함수들을 활용하여 main()에서 플레이리스트를 구현했습니다.

정리하면 Ui와 Menu가 뜨고 그 메뉴 중 회원가입을 입력합니다. Member클래스와 함수들로 입력한 아이디와 비번을 메모장에서 확인 후 저장합니다. 다시 회원가입한 정보로 로그인합니다.

노래보기를 입력 시 class Song에 노래들의 이름이 저장되고 1부터 5의 인기도를 랜덤으로 부과해 노래의 이름과 같이 저장합니다. 그리고 노래들이 인기도가 높은 순으로 차례대로 출력하게 됩니다.

플레이리스트는 username을 이름으로 가져와 메모장을 생성해 그 메모장에 노래들을 저장해 플레이리스트로 사용합니다.

* 팀 구성 및 역할분담

팀 구성은 조장 유동균 그리고 조원 유지원, 이충훈, 권태열로 구성되어 있습니다.

중간보고서 전까지의 할 일은 주제 및 주제의 세부내용 정하기, 클래스와 주제의 알고리즘 생각해 정하기, 코드 작성, 발표자료(ppt)만들기, 보고서 작성 등이 있었습니다.

주제와 주제의 세부내용은 팀원 모두가 모여 의견을 냈습니다. 주제는 권태열 조원이 의견을 내서 모두 동의를 하였고 주제의 세부내용, 기능은 이충훈 조원 그리고 다른 모든 인원이 서로 주제의 세부 기능들의 의견을 제시하고 알고리즘을 생각해 봤습니다.

의견을 내고 정한 주제를 코드의 클래스와 객체, 함수를 어떻게 사용하는지 아이디어, 알고리즘을 유동균 조원과 유지원 조원이 함께 의견을 내며 만들었습니다. 코드 상에서 틀, Ui클래스의 구성과 Member클래스, 함수들을 작성하며 Song 클래스의 객체배열을 작성하였고 코딩하였습니다. 권태열 조원과 이충훈 조원은 함께 MusicPlayer클래스의 함수 구성을 코딩하며 클래스의 main을 작성하는 부분을 코딩하는데 힘썼습니다. 중간 보고서의 발표자료(ppt)는 유지원 조원이 만들었으며 보고서는 유동균 부원이 작성하였습니다.

미팅은 주제와 세부내용을 정할 때까지는 모든 조원이 직접 만나 정했습니다. 이후 클래스와 코드의 세부적인 내용은 통화와 카톡을 통한 소통으로 코드를 절충해 나아갔습니다.

* 미팅 일지(수기로 작성)

