

회원제로 운영되는 UP/DOWN 게임을 만들려고 한다.

아래 프로젝트 구조, 코드를 참고하여 프로그램을 완성시키세요.

[프로젝트명]

UpDownProject

[생성해야되는 클래스]

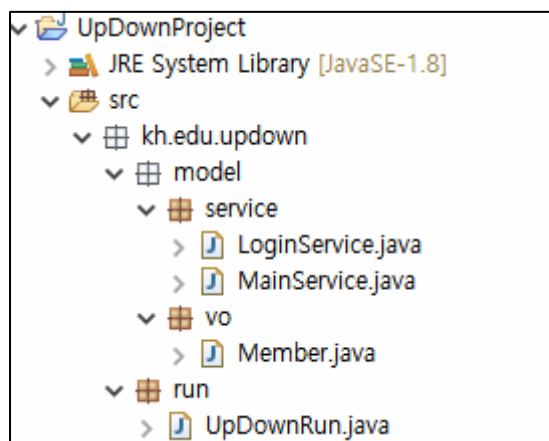
kh.edu.updown.model.service.**MainService**

kh.edu.updown.model.service.**LoginService**

kh.edu.updown.model.vo.**Member**

kh.edu.updown.run.**UpDownRun**

[프로젝트 구조]



Member 클래스

```
public class Member {
    private String memberId;
    private String memberPw;
    private String memberName;
    private int highScore;

    // 생성자

    // getter / setter
}
```

MainService 클래스

```
package kh.edu.updown.model.service;

import java.util.InputMismatchException;
import java.util.Scanner;

import kh.edu.updown.model.vo.Member;

public class MainService {

    private Scanner sc = new Scanner(System.in);

    // 회원 가입된 회원의 정보를 저장할 배열 선언 및 할당
    private Member[] members = new Member[3];

    // 현재 가입한 회원 수 (회원 가입 시 members 인덱스 지정에 사용 가능)
    private int memberCount = 0;

    // 현재 로그인한 회원의 정보를 참조할 변수 선언.
    private Member loginMember = null; // 로그인 X == null, 로그인 O != null

    // LoginService 선언
    private LoginService loginService = new LoginService();

    // 메뉴 출력 메소드
    // * 메뉴 출력 메소드를 분석해보세요!
    public void displayMenu() {

        int sel = 0; // 메뉴 선택용 변수

        do {
            try {

                System.out.println();
                System.out.println("=== UP/DOWN 게임 ===");
                if(loginMember == null) { // 로그인이 되어있지 않은 경우
```

```

        System.out.println("[메인 메뉴]");
        System.out.println("1. 회원가입");
        System.out.println("2. 로그인");
        System.out.println("0. 종료");
        System.out.print("메뉴 선택 >> ");

        sel = sc.nextInt();
        sc.nextLine();
        System.out.println();

        switch(sel) {
            case 1 : signUp(); break;
            case 2 : login(); break;
            case 0 : System.out.println("[프로그램 종료]"); break;
            default : System.out.println("잘못 입력하셨습니다. 메뉴를
다시 선택해주세요.");
        }

    }else { // 로그인이 되어있는 경우

        System.out.println("[로그인 메뉴]");
        System.out.println("1. 업/다운 게임 start");
        System.out.println("2. 내 정보 조회");
        System.out.println("3. 전체 회원 조회");
        System.out.println("4. 비밀번호 변경");
        System.out.println("9. 로그아웃");
        System.out.print("메뉴 선택 >> ");

        sel = sc.nextInt();
        sc.nextLine();
        System.out.println();

        switch (sel) {
            case 1: loginService.startGame(loginMember); break;
            case 2: loginService.selectMyInfo(loginMember); break;
            case 3: loginService.selectAllMember(members); break;
            case 4: loginService.updatePassword(loginMember); break;

            case 9 : System.out.println("로그아웃 되었습니다.");
                    loginMember = null; // loginMember 필드에
아무것도 참조하고 있지 않음을 의미하는 null을 대입
                    break;

            default: System.out.println("잘못 입력하셨습니다. 메뉴를
다시 선택해주세요.");
        }

    }

}

}catch (InputMismatchException e) {

    System.out.println("정수만 입력해주세요.");
    sc.nextLine(); // 버퍼에 남아있는 문자열 제거

```

```

    }

    }while(sel != 0);

}

// 회원 가입
// * 회원 가입 코드를 분석해보세요.
public void signUp() {

    System.out.println("[회원 가입]");

    // memberCount가 members 배열 길이 이상인 경우 == 회원 가입 정원이 다 찬 상태
    if(memberCount >= members.length) {
        System.out.println("회원 정원이 가득차 더 이상 가입을 할 수 없습니다.");

    } else { // 회원 가입이 가능한 경우

        String memberId = null; // 아이디를 입력 받아 저장할 변수

        while(true) {
            System.out.print("아이디 : ");
            memberId = sc.next();

            boolean flag = true; // 중복 아이디 체크용 변수

            for(int i=0 ; i<members.length ; i++) {

                if(members[i] != null) { // members 배열 요소가 null이
아닌경우 == 회원 정보가 있는 경우

                    // 입력 받은 아이디와 같은 아이디가 members에
존재하는 경우

                    if( memberId.equals(members[i].getMemberId()) ) {
                        System.out.println("중복되는 아이디 입니다.
다시 입력해주세요.");

                        flag = false;
                        break; // 중복검사 for문 종료
                    }

                }else { // members 배열요소가 null인 경우 == 회원 정보가
없는 경우

                    break; // 중복검사 for문 종료
                }
            } // end for

            if(flag) { // 중복 아이디가 아닌 경우
                break; // 아이디를 반복적으로 입력 받는 while문 종료
            }
        } // end while

        System.out.print("비밀번호 : ");

```

```
String memberPw = sc.next();

System.out.print("이름 : ");
String memberName = sc.next();

// 입력 받은 값을 이용해 새로운 Member 객체를 생성하고 members 배열의 요소

// memberCount번째 요소가 참조할 수 있도록 주소를 저장
members[memberCount] = new Member(memberId, memberPw, memberName);

// 새로운 회원이 가입되었으므로 memberCount를 1 증가
memberCount++;

System.out.println("*** 회원 가입이 완료되었습니다. ***");

    } // end else (회원 가입이 가능한 경우)
} // end signUp()


// 로그인
// 아이디, 비밀번호를 입력 받아 일치하는 회원이 members에 있을 경우 로그인
// 없으면 "아이디 또는 비밀번호가 일치하지 않습니다." 출력
public void login() {
    System.out.println("[로그인]");
}
}
```

LoginService 클래스

```
package kh.edu.updown.model.service;

import java.util.Scanner;

import kh.edu.updown.model.vo.Member;

public class LoginService {

    private Scanner sc = new Scanner(System.in);

    // 업다운 게임 시작
    // 1 ~ 100 사이 숫자 중 랜덤하게 한 숫자를 지정하고 업/다운 게임을 진행
    // 맞춘 횟수가 현재 로그인한 회원의 최초 또는 최고 기록인 경우
    // 회원의 highScore 필드 값을 변경
    public void startGame(Member loginMember) {
        System.out.println("[Game Start...]");
    }

    // 내 정보 조회
    // 로그인한 멤버의 정보 중 비밀번호를 제외한 나머지 정보만 화면에 출력
    public void selectMyInfo(Member loginMember) {
        System.out.println("[내 정보 조회]");
    }

    // 전체 회원 조회
    // 전체 회원의 아이디, 최고점수를 출력
    public void selectAllMember(Member[] members) {
        System.out.println("[전체 회원 조회]");
    }

    // 비밀번호 변경
    // 현재 비밀번호를 입력 받아
    // 같은 경우에만 새 비밀번호를 입력 받아 비밀번호 변경
    public void updatePassword(Member loginMember) {
        System.out.println("[비밀번호 변경]");
    }
}
```

UpDownRun 클래스

```
package kh.edu.updown.run;

import kh.edu.updown.model.service.MainService;

public class UpDownRun {
    public static void main(String[] args) {

        MainService service = new MainService();
        service.displayMenu();
    }
}
```

실행화면 - 회원 가입

=== UP/DOWN 게임 ===

[메인 메뉴]

1. 회원가입

2. 로그인

0. 종료

메뉴 선택 >> 1

[회원 가입]

아이디 : user01

비밀번호 : pass01

이름 : 유저일

*** 회원 가입이 완료되었습니다. ***

=== UP/DOWN 게임 ===

[메인 메뉴]

1. 회원가입

2. 로그인

0. 종료

메뉴 선택 >> 1

[회원 가입]

아이디 : user01

중복되는 아이디 입니다. 다시 입력해주세요.

아이디 : user02

비밀번호 : pass02

이름 : 유저이

*** 회원 가입이 완료되었습니다. ***

=== UP/DOWN 게임 ===

[메인 메뉴]

1. 회원가입

2. 로그인

0. 종료

메뉴 선택 >> 1

[회원 가입]

아이디 : user03

비밀번호 : pass03

이름 : 유저삼

*** 회원 가입이 완료되었습니다. ***

=== UP/DOWN 게임 ===

[메인 메뉴]

1. 회원가입

2. 로그인

0. 종료

메뉴 선택 >> 1

[회원 가입]

회원 정원이 가득차 더 이상 가입을 할 수 없습니다.

실행화면 - 로그인

=== UP/DOWN 게임 ===

[메인 메뉴]

1. 회원가입

2. 로그인

0. 종료

메뉴 선택 >> 2

[로그인]

아이디 : user01

비밀번호 : 1234

아이디 또는 비밀번호가 일치하지 않습니다.

=== UP/DOWN 게임 ===

[메인 메뉴]

1. 회원가입

2. 로그인

0. 종료

메뉴 선택 >> 2

[로그인]

아이디 : user01

비밀번호 : pass01

유저일님 환영합니다.

=== UP/DOWN 게임 ===

[로그인 메뉴]

1. 업/다운 게임 start

2. 내 정보 조회

3. 전체 회원 조회

4. 비밀번호 변경

9. 로그아웃

메뉴 선택 >>

실행화면 - 업/다운 게임 start

=== UP/DOWN 게임 ===

[로그인 메뉴]

1. 업/다운 게임 start
2. 내 정보 조회
3. 전체 회원 조회
4. 비밀번호 변경
9. 로그아웃

메뉴 선택 >> 1

[Game Start...]

1번째 입력 : 50

-- DOWN --

2번째 입력 : 25

-- UP --

3번째 입력 : 37

-- UP --

4번째 입력 : 43

-- UP --

5번째 입력 : 47

-- DOWN --

6번째 입력 : 45

-- UP --

7번째 입력 : 46

정답!!

입력 시도 횟수 : 7

*** 최고 기록 달성 ***

실행화면 - 내 정보 조회

=== UP/DOWN 게임 ===

[로그인 메뉴]

1. 업/다운 게임 start
2. 내 정보 조회
3. 전체 회원 조회
4. 비밀번호 변경
9. 로그아웃

메뉴 선택 >> 2

[내 정보 조회]

아이디 : user01

이름 : 유저일

최고점수 : 7회

실행화면 - 전체 회원 조회

=== UP/DOWN 게임 ===

[로그인 메뉴]

1. 업/다운 게임 start
2. 내 정보 조회
3. 전체 회원 조회
4. 비밀번호 변경
9. 로그아웃

메뉴 선택 >> 3

[전체 회원 조회]

[아이디] [최고점수]

user01	7
user02	0
user03	0

실행화면 - 비밀번호 변경

=== UP/DOWN 게임 ===

[로그인 메뉴]

1. 업/다운 게임 start
2. 내 정보 조회
3. 전체 회원 조회
4. 비밀번호 변경
9. 로그아웃

메뉴 선택 >> 4

[비밀번호 변경]

현재 비밀번호 입력 : 1234

현재 비밀번호가 일치하지 않습니다.

=== UP/DOWN 게임 ===

[로그인 메뉴]

1. 업/다운 게임 start
2. 내 정보 조회
3. 전체 회원 조회
4. 비밀번호 변경
9. 로그아웃

메뉴 선택 >> 4

[비밀번호 변경]

현재 비밀번호 입력 : pass01

새 비밀번호: 123123

비밀번호가 변경되었습니다.

실행화면 - 로그아웃

=== UP/DOWN 게임 ===

[로그인 메뉴]

1. 업/다운 게임 start
2. 내 정보 조회
3. 전체 회원 조회
4. 비밀번호 변경
9. 로그아웃

메뉴 선택 >> 9

로그아웃 되었습니다.

=== UP/DOWN 게임 ===

[메인 메뉴]

1. 회원가입
2. 로그인
0. 종료

메뉴 선택 >>