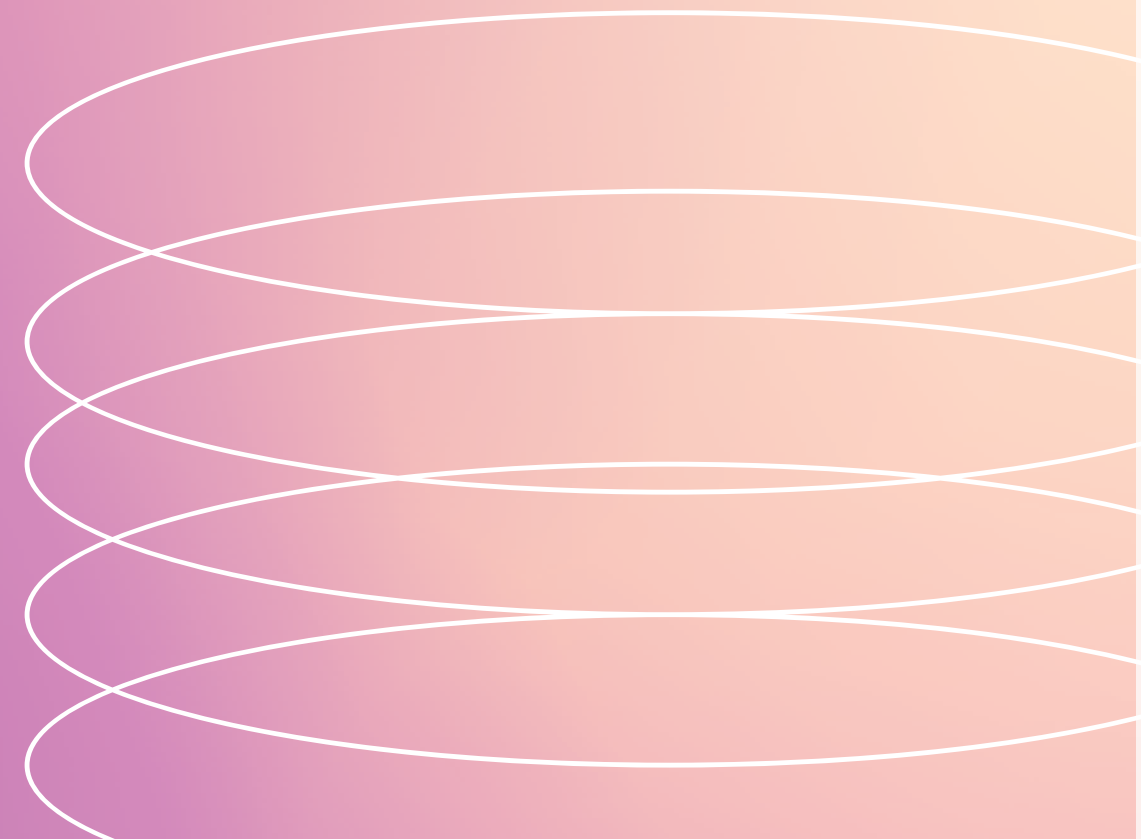


콘솔프로젝트 PilatesClass

유슬비 장민정 김은영 김동혁





필라테스
클래스

콘솔프로젝트

주제선정

필라테스 수업, 회원 관리 프로그램

역할분담

장민정

회원가입,로그인,로그아웃,아이디찾기,비밀번호찾기,
수강내역취소,결제 및 거스름돈

김동혁

모든 강사 출력 , 모든 회원 출력 , 강사 인기순위 출력

유슬비

수업관리[CRUD] , 월별매출, 일자별매출, 메시지 , 포인트

김은영

예약하기,예약목록출력,예약유효성검사,이전수업목록 출력 안내하기

장민정

2/24(금) 주제선정,조장뽑기,DB설계
2/27(월) Github 생성
2/28(화) mvc(dto dao),회원가입,로그인
3/2 (목) 회원창 수정, 수강내역 취소
3/3 (금) 로그아웃,아이디찾기,비밀번호찾기
3/6 (월) 결제 및 거스름돈
3/7 (화) 수정 및 최종점검

김동혁

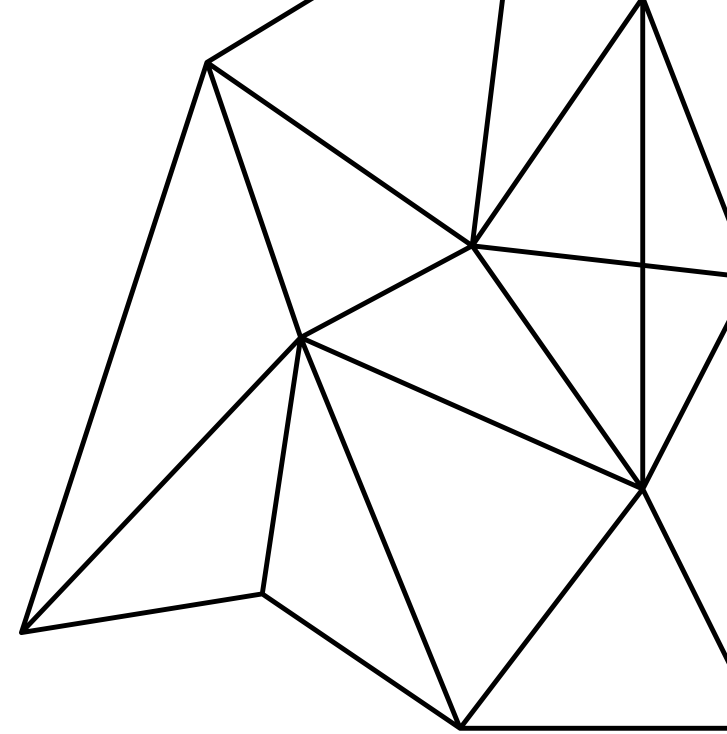
2/24(금) 주제선정 , DB설계
2/27(월) MVC설계 , 역할분담
2/28(화) 모든 강사 출력
3/2 (목) 모든 회원 출력
3/3 (금) 강사 인기순위 DB설계
3/6 (월) 강사 인기순위 출력
3/7 (화) 수정 및 최종점검 , 발표

김은영

2/24(금) 주제선정,조장뽑기,DB설계
2/27(월) Github 생성
2/28(화) mvc(dto dao),예약 하기,예약목록출력
3/2 (목) 예약하기,예약목록 코딩,
3/3 (금) 예약하기,예약목록 수정,예약유효성검사
3/6 (월) 날짜지난수업 출력 없게하기
3/7 (화) 수정 및 최종점검

유슬비

2/24(금) 주제선정 , DB설계
2/27(월) MVC설계 , 역할분담
2/28(화) 수업목록 등록,출력
3/2 (목) 모든 회원 출력
3/3 (금) 수업목록 수정,삭제
3/6 (월) 월별매출, 일자별매출, 메시지 발송,알림,출력
3/7 (화) 결제시 포인트사용, 포인트적립

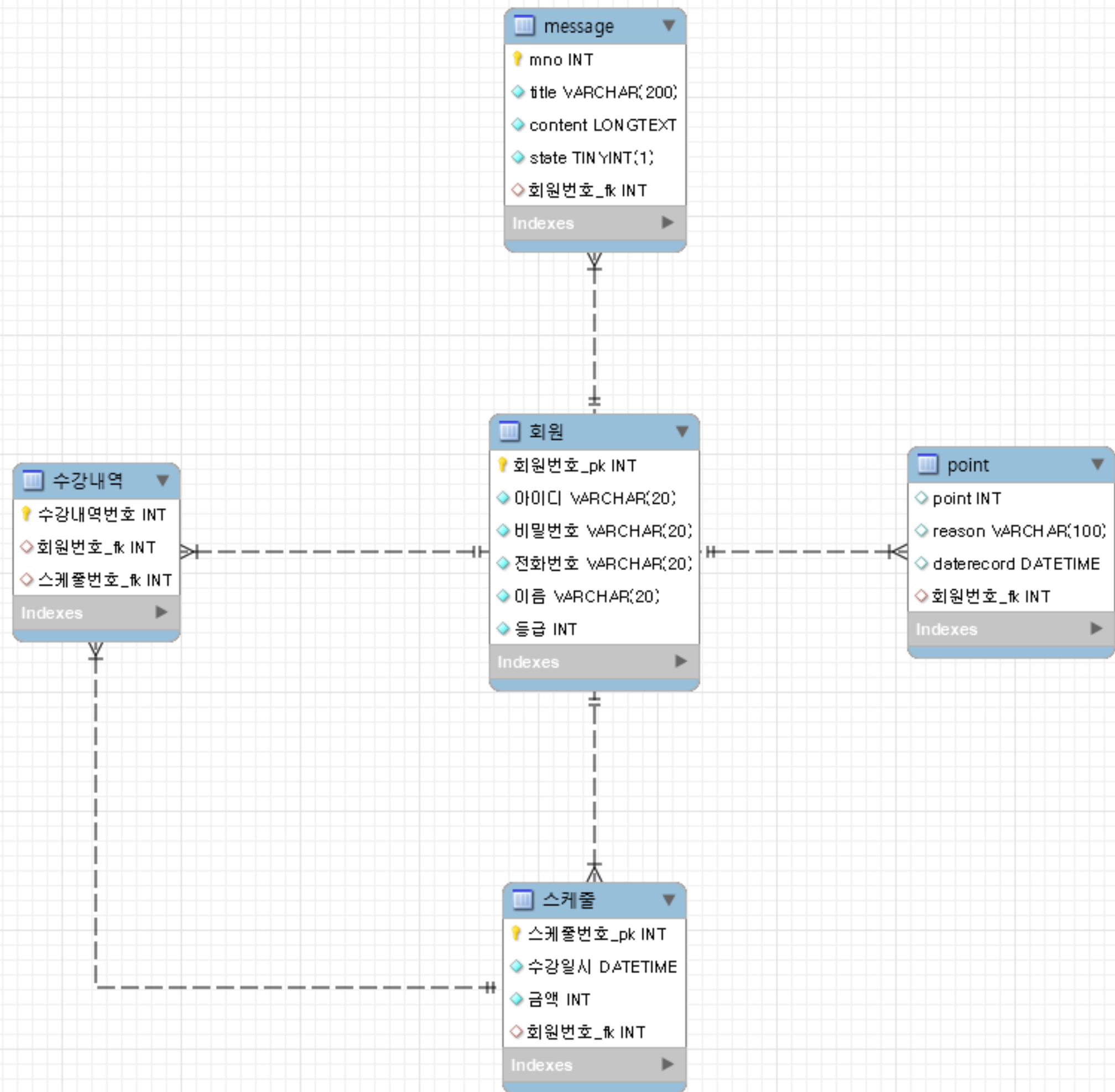


일정

스킬



DB설계도



컨트롤 구조도

HOME

회원

- 회원가입
- 로그인(회원/강사)
- 아이디 찾기
- 비밀번호 찾기
- 로그아웃
- 메시지 확인

예약/결제

- 결제
- 포인트 사용/적립
- 포인트 조회
- 예약
- 예약내역 출력
- 예약취소

수업관리

- 수업 등록
- 수업 출력
- 수업 수정
- 수업 삭제

회원/강사관리

- 회원 조회
- 개인 메시지 발송
- 전체 메시지 발송
- 강사 조회
- 강사 인기순위

매출관리

- 누적 매출 출력
- 월별 매출 출력
- 일자별 매출 출력
- 날짜 직접 입력

주요 기능

강사 인기순위

누적예약수로 강사 인기순위 출력

===== 강사 인기순위 =====		
강사이름	누적수강생	랭킹
강호동	4	1
신동엽	3	2
장도연	1	3
=====		
아무키나 입력하십시오 (뒤로가기)		

회원view

```
public void getTchRank() {
    System.out.println("===== 강사 인기순위 =====");
    System.out.printf("%-10s\t%-5s\t%s\n", "강사이름", "누적수강생", "랭킹");
    ArrayList<RankDto> teacherRankList = 회원controller.getInstance().teacherRank();
    for(RankDto rn : teacherRankList) {
        System.out.printf("%-10s\t%-5d\t%s\n", rn.get회원이름(), rn.get예약수(), rn.get랭킹());
    }
    System.out.println("=====");
    System.out.println("아무키나 입력하십시오 (뒤로가기)");
    String ch = scanner.next();
}
```



```
public ArrayList<RankDto> getTchRank() {
    ArrayList<RankDto> rankList = new ArrayList<>();
    String sql = "select c.이름 as 강사명 , count(*) as 누적수강생 , "
        + "rank() over ( order by count(*) desc ) as 랭킹 "
        + "from 수강내역 a , 스케줄 b , 회원 c "
        + "where a.스케줄번호_fk = b.스케줄번호_pk "
        + "and b.회원번호_fk = c.회원번호_pk "
        + "group by c.회원번호_pk" ;

    try {
        ps=con.prepareStatement(sql);
        rs = ps.executeQuery();
        while(rs.next()) {
            RankDto rankDto = new RankDto( rs.getString(1) , rs.getInt(2) ,rs.getInt(3) );
            rankList.add(rankDto);
        }
        return rankList;
    }catch (Exception e) {System.out.println(e);}
    return null;
}
```

2. rank 함수 이용 - rank()

```
select id, amount, rank() over (order by amount desc) as ranking
from ex_card
```

rank() over (order by 정렬할기준) 으로 하나의 컬럼을 생성할 수 있다.

지금은 금액이 많은 순서대로 랭킹을 부여했다.

아까와 다르게 금액이 같은 경우일 경우 **같은 순위**를 가진다. (amount 2000이 같을 경우 둘다 랭킹 9)

그리고 또하나, 같은 경우가 생기고 난 다음에는 다음순위는 **건너뛴다**.

그래서 랭킹 4가 두개가 나타나서 그 다음엔 5가 아니라 6이 나타나게 된다.

	id	amount	ranking
▶	133875	69950	1
	10902	61000	2
	201783	43570	3
	146876	34000	4
	162678	34000	5
	42495	13200	6
	7803	9700	7
	129788	8000	8
	133221	2000	9
	133223	2000	10

	id	amount	ranking
▶	133875	69950	1
	10902	61000	2
	201783	43570	3
	146876	34000	4
	162678	34000	5
	42495	13200	6
	7803	9700	7
	129788	8000	8
	133221	2000	9
	133223	2000	10

느낀점

장민정

클래스가 한정적이다 보니 다른 팀원들의 작업건과 충돌이 날뻔했지만 자세한 주석 처리와 많은 대화를 통해 문제를 해결할수 있었고,서로만든 기능들을 공유하며 많은 공부가 되었습니다.

김은영

프레젝트를 할때마다 할수있을까 의문 점이 들었지만 팀원들과 모르는부분을 공유하고 조금씩 제공부가 되어가는 과정이었습니다~

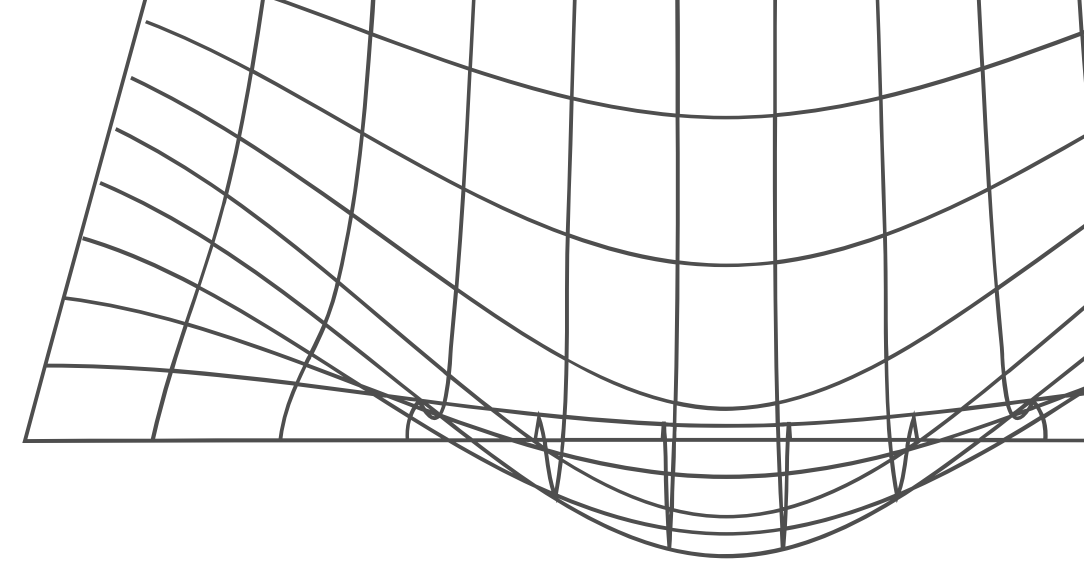
김동혁

프론트때와는 다르게 오류가 발생해도 어디서 발생한건지 찾기가 너무 어려웠고 이외에도 여러모로 힘든점이 많았지만 배려심 넘치는 조원 분들 덕분에 모르는 부분은 서로서로 공부할 수 있는 좋은 시간이 되었던 것 같다 :)

유슬비

팀원들 분위기가 좋아서 더 힘을 내서 작업할 수 있었던 것 같습니다. 서로 공부하는 시간이 되어서 좋았습니다.

보완할 점



01

GitHub 취합시

조원이 pull을
여러번 하지 않아도
되도록 오류 점검 후
취합 안내하기

02

구현 후 조원 서로 피드백 하기

서로 구현된 기능 사
용해서 피드백 주기

JVM

Start 클래스 (메인메소드 실행)

```
package pilatesClass;

import pilatesClass.view.Front;

public class Start {

    public static void main(String[] args) {

        Front.getInstance().index();

    }

}
```

Front클래스

```
package pilatesClass.view;

import java.util.Scanner;

public class Front {

    private static Front front=new Front();
    private Front() {};

    public static Front getInstance() {
        return front;
    }

    public void index() {
        while (true) {
            System.out.println("    PilatesClass");
            System.out.println("
+ "|   |   |   |   |   |\r\n"
+ "|   |   |   ☆   |   |\r\n"
+ "|   |   ☆       ☆   |\r\n"
+ "|   ☆           ☆   |\r\n"
+ "☆               ☆");
            System.out.println(" 1.관리자페이지  2.회원페이지");
        }
    }
}
```



감사합니다!