

Java based Management System Development

egyout@induk.ac.kr

개인 정보 관리(PIM) 시스템 구현

■ 구현 목표

- 분석된 업무를 구현할 수 있는 능력을 배양한다.
 - PIM(Personal Information Management) : 회원의 정보를 관리할 수 있는 시스템
 - 등록, 조회, 수정, 삭제, 검색, 정렬 등 기본 기능 제공
- Java 클래스 정의, 클래스 상속, 인터페이스 구현 능력을 배양한다.
- 객체 생성과 메소드 호출 관계에 대한 이해도를 배양한다.
- 제너릭, 클래스 형변환에 대한 활용 능력을 배양한다.

New Project

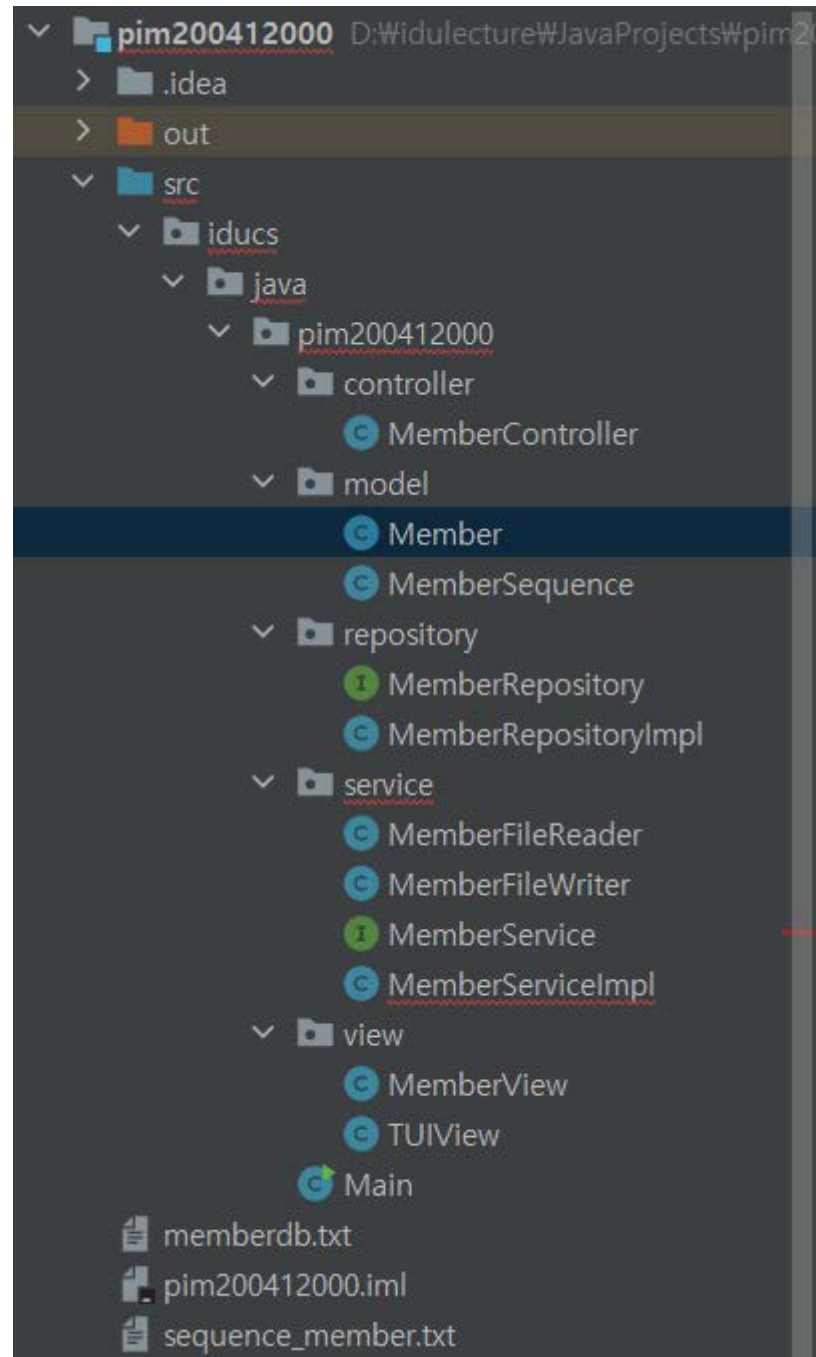
Project name: pim200412000

Project location: D:\wilecture\JavaProjects\pim200412000

Base package: iducs.java.pim200412000

Previous Finish Cancel Help

Project name : pim<학번>
Base Package : iducs.java.pim<학번>



■ 회원정보 클래스 : Member.java

- 필드 정보

```
package iducs.java.pim200412000.model;

public class Member {
    public int id;
    private String email; // 길이 20, 이메일, 유일키
    private String pw; // 길이 15, 암호
    private String name; // 길이 15, 이름
    private String phone; // 길이 20, 전화번호
    private String address; // 길이 30, 주소
```

■ 기본 기능

- 회원 가입 (기본 정보로 가입)
 - 가입 / 수정 시 이메일 형식 확인(@ 기호)
- 회원 로그인
- 회원 정보 조회
- 회원 정보 수정
 - 수정 시 이메일 변경 불가, 이메일 형식 확인
- 회원 탈퇴 (정보 삭제)

■ 파일

- 프로그램 시작 시 파일에서 읽기
- 프로그램 종료 시 파일에 저장하기

TUIView.java

```
package iducs.java.pim200412000.view;

import iducs.java.pim200412000.model.Member;

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

public class TUIView {
    Member m = new Member();
    Scanner sc = new Scanner(System.in);

    public void showMenu(boolean isLogin, boolean isRoot) { // 메뉴 보기
        if(isLogin == false) { // 로그인 전
            System.out.print("0. 종료 ");
            System.out.print("1. 등록 ");
            System.out.print("2. 로그인 ");
            System.out.print("\n");
        }
        else {
            if(isRoot == false) { // 로그인 후 일반 사용자
                System.out.print("3. 조회 ");
                System.out.print("4. 수정 ");
                System.out.print("5. 탈퇴 ");
                System.out.print("6. 로그아웃 "); // isLogin값 false로 설정
                System.out.print("\n");
            }
        }
    }
}
```

```
        else { // 관리자
            System.out.print("0. 종료 ");
            System.out.print("4. 수정 ");
            System.out.print("6. 로그아웃 ");
            System.out.print("7. 전체목록 ");
            System.out.print("8. 전화번호 검색 "); // 지정한 조건으로 검색
            System.out.print("9. 이름 내림차순 정렬 "); // 지정한 조건으로 정렬
            System.out.print("10. 범위 지정 page perCount");
            System.out.print("\n");
        }
    }
}

public int inputMenu() { // 메뉴 입력
    int menu = 0 ;
    boolean flag = false;
    do {
        try {
            menu = Integer.parseInt(sc.nextLine());
            if(menu < 0 || menu > 11 ) {
                System.out.println("해당 메뉴 번호를 입력하십시오.");
            }
            else
                flag = true;
        } catch (InputMismatchException | NumberFormatException ime) {
            System.out.println("숫자 형식을 입력하십시오.");
        }
    } while (!flag);
}
```


- 1. 등록 2. 로그인
- 2
- 아이디를 입력하시오 >>
- id01
- 암호를 입력하시오 >>
- pw01
- 로그인 성공작업을 성공하였습니다.
- 3. 조회 4. 수정 5. 탈퇴 6. 로그아웃
- 3
- id01 name01 id1@induk.ac.kr 000-1001-0001 서울시 강북구
- 3. 조회 4. 수정 5. 탈퇴 6. 로그아웃
- 4
- 수정 항목 번호를 입력하시오 >>
- 0. 수정 종료 1. 이름 2. 이메일 3. 연락처4. 주소 2
- 이메일을 입력하시오(id@induk.ac.kr) >>
- id1001@induk.ac.kr
- 수정 항목 번호를 입력하시오 >>
- 0. 수정 종료 1. 이름 2. 이메일 3. 연락처4. 주소 0
- 수정 항목 번호를 입력하시오 >>
- 0. 수정 종료 1. 이름 2. 이메일 3. 연락처4. 주소 0
- id01name01id1001@induk.ac.kr000-1001-0001서울시 강북구
- 3. 조회 4. 수정 5. 탈퇴 6. 로그아웃

작업

■ 프로젝트 이름 변경

- pim202012000 -> pim<학번>

■ 저장 파일 이름 변경

- memberdb.txt -> db<학번>.txt

■ 패키지 이름 변경

- iducs.java.pim202012000 -> iducs.java.pim<학번>

■ 기타 수정

- 등록 예외 처리 필요
- 작업 중 원본 소스 오류 수정은 수정 후

■ 관리자는 초기 기본 생성

- 기본 root@induk.ac.kr을 admin학번@induk.ac.kr로 db<학번>.txt에서 변경
 - memberdb.txt 파일의 위치는 src와 같은 레벨이고,
 - 수정은 **텍스트 편집기에서 수행**해야 함. (tab 코드가 다름)
- 관리자는 모든 멤버를 대상으로 함
- 회원 전화번호 검색
- 회원 이름으로 내림차순 정렬
- 회원 목록 페이지 검색

주요 구현

- 등록 시 자동 번호 증가 기능
- 가입시 회원 이메일 중복 확인 => 예외 처리
- 8. 회원 전화번호 검색 (전체 또는 마지막 4자리)
- 9. 회원 이름 내림차순으로 정렬 후 화면 출력 : JCF
- 10. 범위 지정 회원 목록 출력 : 1 3
 - page 1, perCount 3 : 처음부터 3개까지 목록 출력 1, 2, 3
 - page 2, perCount 5 : 2페이지 시작, 즉 1~5개 지나고, 최대 5개 목록 출력 6, 7, 8, 9, 10

■ 8. 전화번호 뒷 4자로 검색, 여러개가 검색될 수 있음

- service 계층
 - `List<E> findMemberByPhone(E member);`
- repository 계층
 - `List<E> readListByPhone(T member);`

■ 9. 이름으로 정렬

- order가 desc면 내림차순, asc면 오름차순
- service
 - `List<E> sortByName(String order);`
- repository
 - `List<E> readListByName(String order);`

- **10. page 검색 (지정한 페이지, 페이지당 갯수)**
 - service
 - `List<T> paginateByPerPage(int pageNo, int perPage)`
 - repository
 - `List<T> readListByPerPage(int page, int perPage);`

계속

■ 과제 제출 기한

- 18일 오후 18시까지

■ 제출

- 프로젝트 압축 파일
- 작업 설명 pdf<반><학번><이름>.pdf을 함께 압축하여 과제 게시판에 업로드

■ Q&A

- 1. 밴드톡(오후 11시까지) -> 2. 메일 -> 3. 전화
- anydesk 질문권 있는 학생 사용가능