



```
package iducs.java.pim201912047.pims.view;

public class TUIView { //TUI : Text User Interface, CUI(Character UI)
    public void showMenu(boolean isLogin, boolean isRoot) {
        if(isLogin == false) { // F & F
            System.out.print("1. 등록\t");
            System.out.print("2. 로그인\n");
        } else {
            if(isRoot == false) { // T & F
                System.out.print("3. 정보조회\t");
                System.out.print("4. 정보수정\t");
                System.out.print("5. 로그아웃\t");
                System.out.print("6. 회원탈퇴\n");
            } else { // 관리자만 종료 가능 T & T
                System.out.print("0. 종료\t");
                System.out.print("3. 정보조회\t");
                System.out.print("4. 정보수정\t");
                System.out.print("5. 로그아웃\t");
                System.out.print("7. 목록조회\t");
                System.out.print("8. 번호조회\t");
                System.out.print("9. 회원내림차순정렬 화면출력 \t");
                System.out.print("10. 페이지 출력 \t");
            }
        }
    }
}
```

현재 회원 측에서 회원탈퇴 기능을 구현 하였고
관리자 측에서 번호조회, 정렬, 페이지네이션을 구현 했습니다.
그래서 화면에 출력되는 내용은 위와 같이 구성 했습니다.

```

if(menu == 0 && isRoot == true)
{
    msg = "종료";
    memberService.saveFile(); // memberdb.txt 에 저장
    System.out.println("관리자 정상 종료");
    menu2 = menu;
}

```

```

        msg = "입력 코드 확인 :";
        // System.out.println("권한이
    }

    } while(menu2 != 0);
}
}

```

```

loop2:if(menu == 6 && isRoot == false ) {
    msg = "회원탈퇴";

    if(memberService.deleteMember(sessionMember) == 0) {
        memberService.del_saveFile();
        memberView.printOne(member);
        memberView.printMsg(msg + "를 성공했습니다.");
        System.exit( status: 0);
    }
    else {
        System.out.println("수정에 실패하였습니다. ");
        break loop2;
    }
    /*
    memberView.printOne(memberService.getMember(
        (Member) session.get("member")));

    */

}

if(menu == 7 && isRoot == true) {
    msg = "목록조회";
    memberView.printList(memberService.getMemberList()); // 현재 리스트 들어간거 다
    memberView.printMsg(msg + "를 성공했습니다.");
}

```

기존 pimcontroller에서는 권한에 상관없이 번호가 다 눌렸지만 루트인지 아닌지를 if 문으로 넘겨주어서 회원 탈퇴는 일반 사용자만 그리고 나머지 추가 기능은 관리자만 누를 수 있게 수정했습니다

그리고 0번 같은 경우는 기존에 menu가 0이 들어가는 순간 종료가 되기 때문에 Menu2로 수정하여 관리자 측에서 0을 입력할 때만 menu2에 0값을 넣도록 구현했습니다.

pim201912047

db201912047.txt

프로젝트

pim201912047

.idea

out

src

iducs

java

pim201912047

pims

실행: Main (1) ×

"C:\Program Files\Eclipse Adoptium\jdk-11.0.1-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBr

1. 등록 2. 로그인

2

a@induk.ac.kr

1234

로그인을 성공했습니다.

3. 정보조회 4. 정보수정 5. 로그아웃 6. 회원탈퇴

실행: Main (1) ×

"C:\Program Files\Eclipse Adoptium\jdk-11.0.14.101-hotspot\bin\java.exe" "-javaagent:C:\Program Files

1. 등록 2. 로그인

2

a@induk.ac.kr

1234

로그인을 성공했습니다.

3. 정보조회 4. 정보수정 5

6

a@induk.ac.kr a

회원탈퇴를 성공했습니다.

종료 코드 0(으)로 완료된 프로

MemberRepositoryImpl.java

1 38 induk@ac.kr 1234 바바바 01012341234 goro

2 46 d@induk.ac.kr 1234 라라라라 01099999999 goro

3 41 c@induk.ac.kr 1234 다다다 01011111111 goro

4 26 b@induk.ac.kr 1234 나나난 01011112222 goro

5 1 admin@induk.ac.kr 1234 김김 01012341234 goro

6 11 a@induk.ac.kr 1234 a 01011111111 goro

7

MemberServiceImpl.java

public void del_saveFile() { // throws : 예외 전파, throw : 예외 발생

File file = new File(memberdb);

try {

MemberFileWriter<Member> mfw = new MemberFileWriter<>(file);

mfw.delsaveMember((List<Member>) memberRepository.readList());

} catch (IOException e) { // 예외를 직접 처리, unchecked exception

e.printStackTrace();

}

TUIView.java

Loop2: if (menu == 6 && isRoot == false) {

msg = "회원탈퇴";

if (memberService.deleteMember(sessionMember) == 0) {

memberService.del_saveFile();

memberView.printOne(member);

memberView.printMsg(msg + "를 성공했습니다.");

System.exit(status: 0);

}

else {

System.out.println("수정에 실패하였습니다. ");

break loop2;

}

Objects.java

public void delsaveMember(List<T> memberList) throws IOException {

for (T member : memberList) { // for each 문장

try {

Member sessionMember = (Member) session.get("member");

if (member == sessionMember)

{

continue;

}

Member m = (Member) member;

fw.write(str: m.getId() + "\t");

fw.write(str: m.getEmail() + "\t");

fw.write(str: m.getPw() + "\t");

fw.write(str: m.getName() + "\t");

fw.write(str: m.getPhone() + "\t");

fw.write(str: m.getAddress() + "\n");

fw.flush(); // 연산 적용

} catch (IOException e) { // 예외 전파하지 않고, 발생한 곳에서 처리

e.printStackTrace();

}

}

fw.close(); // 자원 반납

}

사용자만 회원탈퇴 6번을 누를 수 있
고 버튼을 누르면 현재 로그인해서 세
션에 저장된 객체를 빼고 파일에 다시
저장 하여 탈퇴 하도록 구현했습니다.

```

loop:if(menu == 1)
{
    String same = "";
    member = new Member(); // 등록 하기로 해서

    String fileName = "db201912047.txt";
    Scanner scan = new Scanner(new File(fileName));

    System.out.println("이메일을 입력해주세요");
    same = sc.next();

    //BufferedReader br = new BufferedReader(new FileReader(fileName));
    //파일을 한 문장씩 읽어온다.
    //String str = br.readLine();
    /*
    File file = new File(fileName);
    FileInputStream fis = new FileInputStream(file);
    byte[] data = new byte[(int) file.length()];
    fis.read(data);
    fis.close();
    */
    String text = new String(Files.readAllBytes(Paths.get(fileName)), StandardCharsets.UTF_8);

    //String str = new String(data, "UTF-8");

    if(same.indexOf("@") == -1)
    {
        System.out.println("@ 값을 입력하지 않았습니다");
        break loop;
    }

    if(text.indexOf(same) == -1) // 같은 문자열 없으면 -1 반환함
    {
        // id 자동증가
        double randomValue = Math.random();
        int id_num = (int)(randomValue * 100) + 1;
        member.setId(id_num); // Long 값을 입력받은 자동증가 // 자동 아이디값 증가
        member.setEmail(same); // String 값을 입력 받아 넣음
        System.out.println("비밀번호를 입력해주세요");
        member.setPw(sc.next());
        System.out.println("이름을 입력해주세요");
        member.setName(sc.next());
        System.out.println("휴대폰 번호를 입력해주세요");
        member.setPhone(sc.next());
        System.out.println("주소를 입력해주세요");
        member.setAddress(sc.next());
        memberService.postMember(member); // 생성한걸 담음 즉 하나씩 입력하면 하나하나 들어가고 7번 누르면
        memberView.printOne(member);
        memberView.printMsg(msg + "를 성공했습니다.");
    }
    else{
        System.out.println("같은 이메일이 있습니다.");
        //System.exit(0);
        member = null;
    }
}

```

등록 시 자동번호 증가 기능과 가입 시 이메일 중복
확인 기능을
회원 등록 1번을 누르면 동시에 하도록 구현했습니다.

이메일 입력 시 @값이 없으면 다시 입력을 요청하고
현재 파일을 읽고 입력한 이메일이랑 같은 것이 없으면
이메일을 리스트에 넣을 수 있으며 자동으로 랜덤
한 값이 id로 들어가고 (a 반, b 반이 100명 이하라 1부터
100으로 세팅했습니다)

같은 이메일이 있으면 다시 메뉴 선택하는 곳으로 지정
브레이크 문을 이용해 돌아가게 구현했습니다.

```

1. 등록 2. 로그인
1
이메일을 입력해주세요
qwer
@ 값을 입력하지 않았습니다
1. 등록 2. 로그인
1
이메일을 입력해주세요
admin@induk.ac.kr
같은 이메일이 있습니다.
1. 등록 2. 로그인

```

```
if(menu == 8 && isRoot == true) {
    msg = "번호조회 검색";
    String p = "";
    System.out.println("검색할 휴대폰 번호 전체 자리 혹은 끝 4자리를 입력해주세요.");
    p=sc.next();
    memberService.findMemberByPhone(p); // 현재 리스트 들어간거 다 출력
    memberView.printMsg(msg + "를 성공했습니다.");
}
```

```
@Override
public void findMemberByPhone(String p)
{
    memberRepository.Search_Phone_Number(p);
}
```

```
public void Search_Phone_Number(String p) {
    System.out.println("입력된 정보에 의한 검색 회원");
    System.out.println("이메일    이름    폰번호    주소");
    for (T m : memberList) {
        if (((Member) m).getPhone().indexOf(p) != -1) { // != 이기 때문에 같은 문자열이 있으면 반환
            System.out.printf("%-17s", ((Member) m).getEmail() + "\t");
            System.out.printf("%-17s", ((Member) m).getName() + "\t");
            System.out.printf("%-17s", ((Member) m).getPhone() + "\t");
            System.out.printf("%-17s", ((Member) m).getAddress() + "\t");
            System.out.println();
        }
    }
}
```

실행: Main (1) x

```
0. 종료  3. 정보조회  4. 정보수정  5. 로그아웃  7. 목록조회
검색할 휴대폰 번호 전체 자리 혹은 끝 4자리를 입력해주세요.
1111
입력된 정보에 의한 검색 회원
이메일    이름    폰번호    주소
ㅂㅈㄷㄱ          bbbb          01011111111
qwe          dddd          01022221111
번호조회 검색을 성공했습니다.
0. 종료  3. 정보조회  4. 정보수정  5. 로그아웃  7. 목록조회
검색할 휴대폰 번호 전체 자리 혹은 끝 4자리를 입력해주세요.
01011111111
입력된 정보에 의한 검색 회원
이메일    이름    폰번호    주소
ㅂㅈㄷㄱ          bbbb          01011111111
번호조회 검색을 성공했습니다.
0. 종료  3. 정보조회  4. 정보수정  5. 로그아웃  7. 목록조회
```

입력 창에서 휴대폰 번호 전체자리 혹은 끝번호를
입력하면 각 메소드 들이 호출 되면서 입력한 번호를
포함 하고 있는 것이 있다면 출력 해주도록 구현 했습니다.
뷰 패키지로 보내 서 출력 해도 되지만 시험 대비 때문에
빠른 수정을 위해 호출은 하지 않은 상태 입니다.

```

loop3:if(menu == 9 && isRoot == true) {
    msg = "회원 정렬";
    String order = "";
    System.out.println("값을 입력해 주세요 desc면 내림차순, asc면 오름차순 (이름값을 기준)");
    order = sc.next();
    if (order.equals("desc") || order.equals("asc")) {
        memberService.lowMemberList(order); // 현재 리스트 들어간거 다 출력
        memberView.printMsg(msg + "를 성공했습니다.");
    } else {
        System.out.println("잘못 입력하셨습니다.");
        break loop3;
    }
}

```

```

@Override
public void lowMemberList(String order) {

    if(order.equals("asc"))
    {
        Collections.sort(memberList, new FruitNameComparator());
        for (T m : memberList) {
            System.out.printf("%-17s", ((Member) m).getEmail() + "\t");
            System.out.printf("%-17s", ((Member) m).getName() + "\t");
            System.out.printf("%-17s", ((Member) m).getPhone() + "\t");
            System.out.printf("%-17s", ((Member) m).getAddress() + "\t");
            System.out.println();
        }
    }

    if(order.equals("desc")) {
        //System.out.println("price 순 내림차순 : " + memberList);
        // name순 내림차순으로 정렬
        Collections.sort(memberList, new FruitNameComparator().reversed());
        for (T m : memberList) {
            System.out.printf("%-17s", ((Member) m).getEmail() + "\t");
            System.out.printf("%-17s", ((Member) m).getName() + "\t");
            System.out.printf("%-17s", ((Member) m).getPhone() + "\t");
            System.out.printf("%-17s", ((Member) m).getAddress() + "\t");
            System.out.println();
        }
    }
}

```

@Override

```

public void lowMemberList(String order) {
    memberRepository.lowMemberList(order);
}

```

```

class FruitNameComparator implements Comparator<T> {
    @Override
    public int compare(T f1, T f2) { return ((Member) f1).getName().compareTo(((Member) f2).getName()); }
}

```

0. 종료 3. 정보조회 4. 정보수정 5. 로그아웃 7. 목록조회 8. 번호조회 9. 회원내림차순정렬 화면출력 10. 페

값을 입력해 주세요 desc면 내림차순, asc면 오름차순 (이름값을 기준)
 잘못 입력하셨습니다.
 0. 종료 3. 정보조회 4. 정보수정 5. 로그아웃 7. 목록조회 8. 번호조회 9. 회원내림차순정렬 화면출력 10. 페

```

desc
g.induk.ac.kr      g      01012341234      goro
f                  f      01012341234      goro
.induk.ac.kre      e      01012341234      goro
qwe                dddd      01022221111      qwe
d.induk.ac.kr      d      01012341234      goro
rerer             cccc      01022222222      qweqwe
.induk.ac.krc      c      01012341234      goro
ㅂㅈㄸㅊ          bbbb      01011111111      qwer
b.induk.ac.kr      b      01012341234      goro
admin@induk.ac.kr  aaaa      01012341234      goro
a.induk.ac.kr      a      01012341234      goro

```

관리자 페이지에서 문자열을 입력하면 desc, asc 에 따라 오름차순 내림차순을 정렬 하여 출력 합니다. 만약 다른 값을 입력하면 다시 메뉴를 선택 하도록 선택했습니다.

Collection에 sort를 이용, 객체안에 문자열을 비교 하여 리턴 하는 함수를 구현했습니다 현재 이름순으 로 정렬을 합니다.

```

if (menu == 10 && isRoot == true) {
    msg = "페이지 출력";
    Scanner sc2 = new Scanner(System.in);
    int page = 0;
    int per_page = 0;
    System.out.println("지정할 페이지, 페이지당 갯수");
    page = sc.nextInt();
    per_page = sc.nextInt();
    memberView.printList(memberService.paginateByPerPage(page, per_page));
}

```

```

@Override
public List<T> paginateByPerPage(int pageNo, int perPage) {
    return memberRepository.readListByPerPage(pageNo, perPage);
}

```

```

@Override
public List<T> readListByPerPage(int page, int perPage) {

    if(perPage <= 0 || page <= 0) {
        throw new IllegalArgumentException("invalid page size: " + perPage);
    }

    int fromIndex = (page - 1) * perPage;
    if(memberList == null || memberList.size() <= fromIndex){
        return Collections.emptyList();
    }

    // toIndex exclusive
    return memberList.subList(fromIndex, Math.min(fromIndex + perPage, memberList.size()));
}
}

```

지정할 페이지와, 페이지당 개수를
입력하면 리스트를 반환해
출력 하도록 구현 했습니다.

현재 리스트에 읽을 것이 있을 때만
가능하도록 구현 했습니다.

그리고 최종적으로 리턴 시 subList
함수를 이용해 잘라서
리스트를 넘겨 구현 했습니다.

pim201912047 > db201912047.txt

프로젝트

pim201912047 C:\woooogie_kill\pim201912047

.idea

out

src

iducs

java

pim201912047

pims

controller

PimController

domain

Member

repository

MemberRepository

MemberRepositoryImpl

service

MemberFileReader

MemberServiceImpl.java

MemberFileWriter.java

Objects.java

db201912047.txt

MemberFi

1

2

3

4

5

6

7

8

9

38

46

41

26

42

43

24

1

induk@ac.kr

d@induk.ac.kr

c@induk.ac.kr

b@induk.ac.kr

e@induk.ac.kr

f@induk.ac.kr

g@induk.ac.kr

admin@induk.ac.kr

1234

바바바

1234

1234

1234

1234

1234

1234

1234

01012341234

01099999999

01011111111

01011112222

01099999999

01011111111

01011112222

01012341234

goro

goro

goro

goro

goro

goro

goro

goro

실행: Main (1)

↑

↓

↺

↻

📷

📄

🗑️

📊

🌟

"C:\Program Files\Eclipse Adoptium\jdk-11.0.14-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
1. 등록 2. 로그인
2
admin@induk.ac.kr
1234
로그인을 성공했습니다.
0. 종료 3. 정보조회 4. 정보수정 5. 로그아웃 7. 목록조회 8. 번호조회 9. 회원내림차순정렬 화면출력 10. 페이지 출력 10
지정할 페이지, 페이지당 갯수
1
3
이메일 이름 연락처 주소
induk@ac.kr 바바바 01012341234 goro
d@induk.ac.kr 라라라라 01099999999 goro
c@induk.ac.kr 다다다 01011111111 goro
0. 종료 3. 정보조회 4. 정보수정 5. 로그아웃 7. 목록조회 8. 번호조회 9. 회원내림차순정렬 화면출력 10. 페이지 출력 10
지정할 페이지, 페이지당 갯수
2
5
이메일 이름 연락처 주소
f@induk.ac.kr 사사사 01011111111 goro
g@induk.ac.kr 아아아 01011112222 goro
admin@induk.ac.kr 김김 01012341234 goro
0. 종료 3. 정보조회 4. 정보수정 5. 로그아웃 7. 목록조회 8. 번호조회 9. 회원내림차순정렬 화면출력 10. 페이지 출력