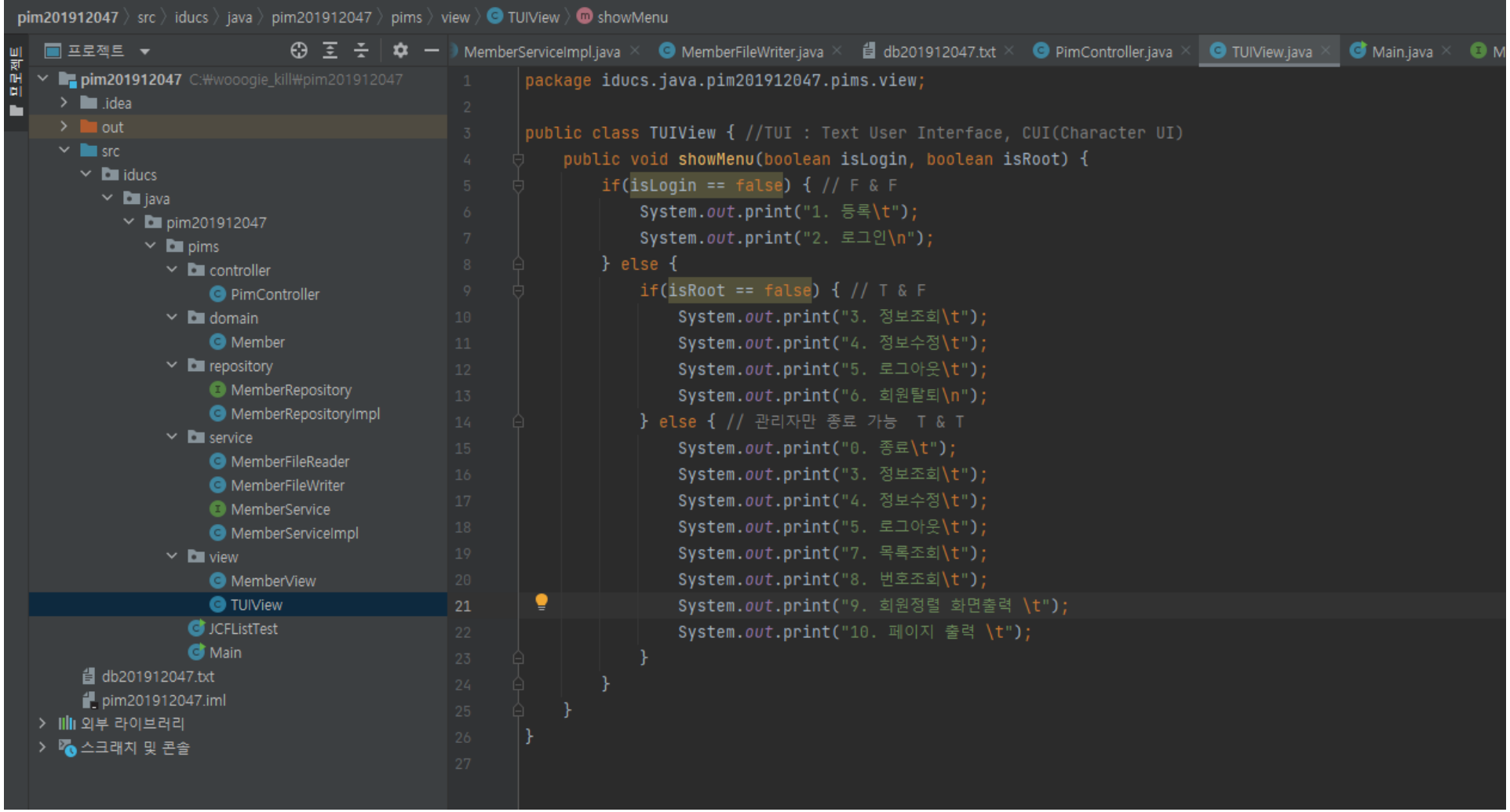


현제 회원 측에서 회원탈퇴 기능을 구현 하였고
관리자 측에서 번호조회, 정렬, 페이지네이션을 구현 했습니다.
그래서 화면에 출력되는 내용은 위와 같이 구성 했습니다.

```
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()); // 현제 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 C:₩wooogie_kill₩pim201912047
.idea
out
src
iducs
java
pim201912047
pims

MemberRepositoryImpl.java | MemberServiceImpl.java | TUIView.java | Objects.j

```
38  induk@ac.kr 1234    바바바   01012341234 goro
46  d@induk.ac.kr   1234    라라라라  01099999999 goro
41  c@induk.ac.kr   1234    다다다   01011111111 goro
26  b@induk.ac.kr   1234    나나난   01011112222 goro
1   admin@induk.ac.kr   1234    김김 01012341234 goro
11  a@induk.ac.kr   1234    a   01011111111 goro
```

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
로그인를 성공했습니다.
3. 정보조회   4. 정보수정   5. 로그아웃   6. 회원탈
```

```
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;
    }
```

```
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();
    }
}
```

```
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
}
```

out
src
iducs
java
pim201912047
pims

```
41  c@induk.ac.kr   1234    다다다   01011111111 goro
26  b@induk.ac.kr   1234    나나난   01011112222 goro
1   admin@induk.ac.kr   1234    김김 01012341234 goro
```

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
로그인를 성공했습니다.
3. 정보조회   4. 정보수정   5
6
a@induk.ac.kr      a
회원탈퇴를 성공했습니다.

종료 코드 0(으)로 완료된 프로
```

사용자만 회원탈퇴 6번을 누를 수 있고 버튼을 누르면 현제 로그인해서 세션에 저장된 객체를 빼고 파일에 다시 저장 하여 탈퇴 하도록 구현했습니다.

```java
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();

    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 입력하시지 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;
        memberService.id_add(); // 현제 리스트 들어간거 다 출력
        member.setId(memberService.id_add() + 1); // Long 값을 입력받음 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;
        break loop;
```

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
현제 파일을 읽고 입력한 이메일이랑 같은 것이 없으면 이메일을 리스트에 넣을 수 있으며
자동으로 id 값을 넣도록 구현했습니다.
같은 이메일이 있으면 다시 메뉴 선택하는 곳으로 지정 브레이크 문을 이용해 돌아가게 구현했습니다.

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

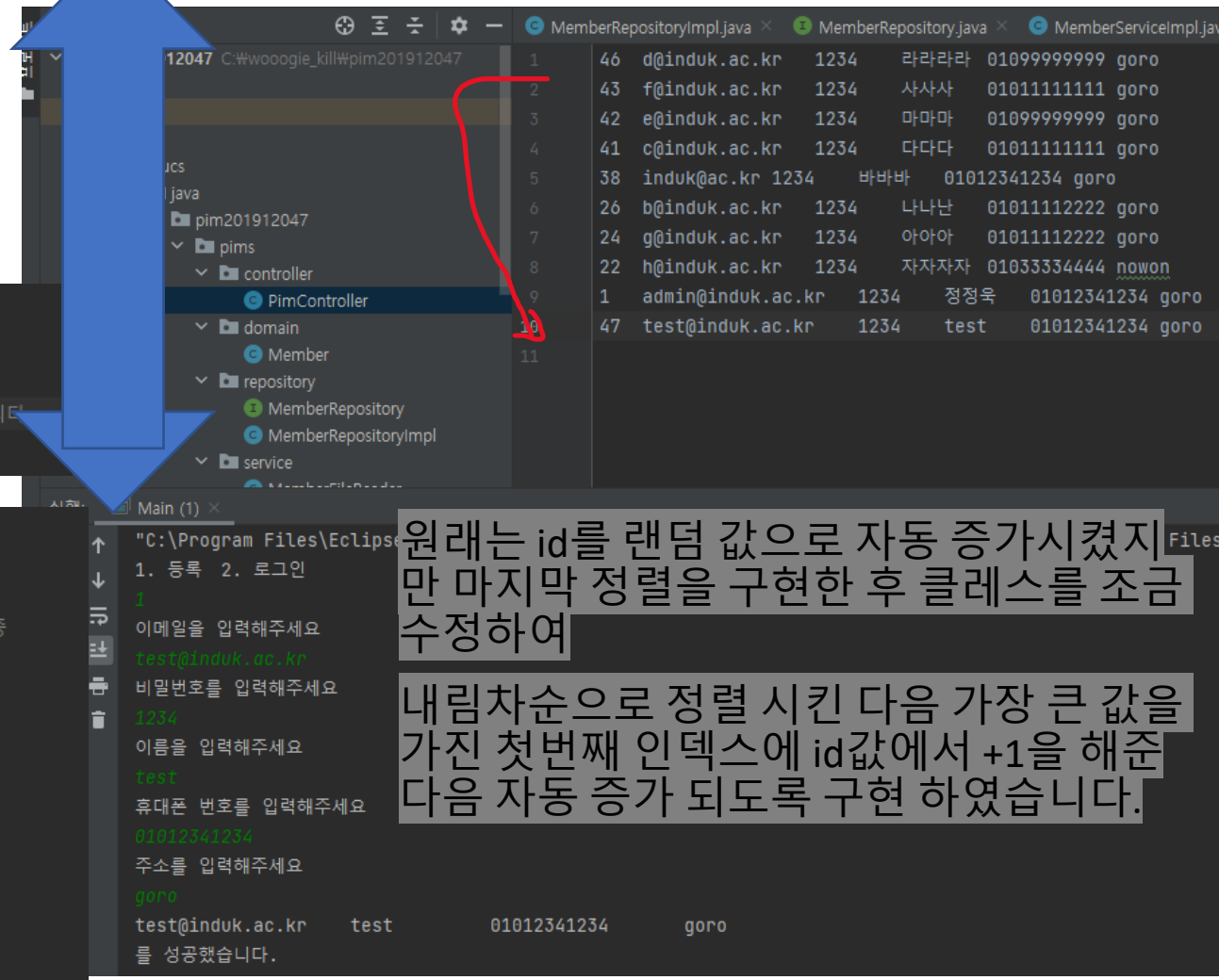
```java
// id 자동증가
//double randomValue = Math.random();
//int id_num = (int)(randomValue * 100) + 1;
memberService.id_add(); // 현제 리스트 들어간거 다 출력
member.setId(memberService.id_add() + 1); // Long 값을 입력받음 자동증가 // 자동 아이디값 증가
member.setEmail(same); // String 값을 입력 받아 넣음
```

```java
public long id_add()
{
    return memberRepository.id_add_List();
}
```

```java
public long id_add_List(){
    Collections.sort(memberList, new Add_Id().reversed());  // id 값 기준 내림차순을 수행후
     T f1  = memberList.get(0); // 가장큰 첫번째 인덱스에서
    return((Member) f1).getId(); // id 값을 리턴하여 최종적으로 가장큰 id값에 +1을 더하도록 만들었습니다
}
```

```java
class Add_Id implements Comparator<T> {
    @Override
    public int compare(T f1, T f2) {    // int형 비교 할거 있으면 여기서 비교하면 됨  현제 id 적용중
        if (((Member) f1).getId() > ((Member) f2).getId()) {
            return 1;
        } else if (((Member) f1).getId() < ((Member) f2).getId()) {
            return -1;
        }

        return 0;
    }
}
```



원래는 id를 랜덤 값으로 자동 증가시켰지만 마지막 정렬을 구현한 후 클래스를 조금 수정하여

내림차순으로 정렬 시킨 다음 가장 큰 값을 가진 첫번째 인덱스에 id값에서 +1을 해준 다음 자동 증가 되도록 구현 하였습니다.

```java
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); // 현제 리스트 들어간거 다 출력
    memberView.printMsg(msg + "를 성공했습니다.");

}
```

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

```java
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)
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      q
번호조회 검색를 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 검색를 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.

```java
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); // 현제 리스트 들어간거 다 출력
        memberView.printMsg(msg + "를 성공했습니다.");
    } else {
        System.out.println("잘못 입력하셨습니다.");
        break loop3;
    }

}
```

```java
@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();
        }
```

```java
@Override
public void lowMemberList(String order) {
    memberRepository.lowMemberList(order);
}
```

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

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 현제 이름순으로 정렬을 합니다.

```
0. 종료  3. 정보조회  4. 정보수정  5. 로그아웃  7. 목록조회  8. 번호조회  9. 회원정렬 화면출력  10. 페이지 출력  9
값을 입력해 주세요 desc면 내림차순, asc면 오름차순 (이름값을 기준)
desc
admin@induk.ac.kr   정정욱              01012341234        goro
h@induk.ac.kr       자자자자            01033334444      nowon
g@induk.ac.kr       아아아              01011112222        goro
f@induk.ac.kr       사사사              01011111111        goro
induk@ac.kr       바바바              01012341234        goro
e@induk.ac.kr       마마마              01099999999        goro
d@induk.ac.kr       라라라라            01099999999      goro
c@induk.ac.kr       다다다              01011111111        goro
b@induk.ac.kr       나나난              01011112222        goro
회원 정렬를 성공했습니다.
0. 종료  3. 정보조회  4. 정보수정  5. 로그아웃  7. 목록조회  8. 번호조회  9. 회원정렬 화면출력  10. 페이지 출력  9
값을 입력해 주세요 desc면 내림차순, asc면 오름차순 (이름값을 기준)
asc
b@induk.ac.kr       나나난              01011112222        goro
c@induk.ac.kr       다다다              01011111111        goro
d@induk.ac.kr       라라라라            01099999999      goro
e@induk.ac.kr       마마마              01099999999        goro
induk@ac.kr       바바바              01012341234        goro
f@induk.ac.kr       사사사              01011111111        goro
g@induk.ac.kr       아아아              01011112222        goro
h@induk.ac.kr       자자자자            01033334444      nowon
admin@induk.ac.kr   정정욱              01012341234        goro
회원 정렬를 성공했습니다.
0. 종료  3. 정보조회  4. 정보수정  5. 로그아웃  7. 목록조회  8. 번호조회  9. 회원정렬 화면출력  10. 페이지 출력
```

```java
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");
    page = sc.nextInt();
    System.out.println("페이지당 갯수");
    per_page = sc.nextInt();
    memberView.printList(memberService.paginateByPerPage(page,per_page));


}
```

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

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

현제 리스트에 읽을 것이 있을 때만 가능하도록 구현 했습니다.

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

```java
@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; //1,3이면 0이되고 2,5면 5가됨     시작점을 정할수 있는거임

    if(memberList == null || memberList.size() <= fromIndex){
        return Collections.emptyList();
        /*
        Collections.emptyList();
        해당 메서드는 DB에서 User객체가 담긴 List를 조회해주는 메서드입니다.
        만약 결과가 없는 경우 메서드를 사용하는 클라이언트 클래스에서는 결과가 없다는것을 알 수 있도록 비어있는 List를 통해 전달해 줍니다.
         */
    }

    // toIndex exclusive
    return memberList.subList(fromIndex, Math.min(fromIndex + perPage, memberList.size())); //5 부터 10까지
    //Math.min 은 두 인자중  작은 값을 리턴 합니다
```

파일(F) 편집(E) 보기(V) 탐색(N) 코드(C) 리팩터링(R) 빌드(B) 실행(U) 도구(T) VCS(S) 창(W) 도움말(H) pim201912047 [C:₩wooogie_kill₩pim201912047] - db201912047.txt

pim201912047 〉 db201912047.txt

프로젝트

- pim201912047 C:₩wooogie_kill₩pim201912047
  - .idea
  - out
  - src
    - iducs
      - java
        - pim201912047
          - pims
            - controller
              - PimController
            - domain
              - Member
            - repository
              - MemberRepository
              - MemberRepositoryImpl
            - service

MemberRepositoryImpl.java | MemberRepository.java | MemberServiceImpl.java | MemberFileWriter.java

```
46  d@induk.ac.kr   1234    라라라라  01099999999 goro
43  f@induk.ac.kr   1234    사사사   01011111111 goro
42  e@induk.ac.kr   1234    마마마   01099999999 goro
41  c@induk.ac.kr   1234    다다다   01011111111 goro
38  induk@ac.kr 1234    바바바   01012341234 goro
26  b@induk.ac.kr   1234    나나난   01011112222 goro
24  g@induk.ac.kr   1234    아아아   01011112222 goro
22  h@induk.ac.kr   1234    자자자자  01033334444 nowon
1   admin@induk.ac.kr   1234    정정욱   01012341234 goro
47  test@induk.ac.kr    1234    test    01012341234 goro
```

실행: Main (1)

```
"C:\Program Files\Eclipse Adoptium\jdk-11.0.14.101-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2
1. 등록  2. 로그인
2
admin@induk.ac.kr
1234
로그인를 성공했습니다.
0. 종료  3. 정보조회  4. 정보수정  5. 로그아웃  7. 목록조회  8. 번호조회  9. 회원정렬 화면출력  10. 페이지 출력  10
지정할 페이지
1
페이지당 갯수
3
이메일                  이름          연락처              주소
d@induk.ac.kr       라라라라       01099999999       goro
f@induk.ac.kr       사사사         01011111111       goro
e@induk.ac.kr       마마마         01099999999       goro
0. 종료  3. 정보조회  4. 정보수정  5. 로그아웃  7. 목록조회  8. 번호조회  9. 회원정렬 화면출력  10. 페이지 출력  10
지정할 페이지
2
페이지당 갯수
5
이메일                  이름          연락처              주소
b@induk.ac.kr       나나난         01011112222       goro
g@induk.ac.kr       아아아         01011112222       goro
h@induk.ac.kr       자자자자       01033334444       nowon
admin@induk.ac.kr   정정욱         01012341234       goro
test@induk.ac.kr    test         01012341234       goro
0. 종료  3. 정보조회  4. 정보수정  5. 로그아웃  7. 목록조회  8. 번호조회  9. 회원정렬 화면출력  10. 페이지 출력
```