

프로젝트명 : Homework3_ObjectArray

[문제 1] 다음과 같은 조건을 만족하는 프로그램을 작성 하시오

1. 사용 데이터

| id | name | password | email | gender | age |
|-------|------|----------|-------------|--------|-----|
| aaaaa | 박신우 | aa | aaa@aaa.aaa | F | 33 |
| bbbbb | 강고결 | bb | bbb@bbb.bbb | M | 34 |
| ccccc | 남나눔 | cc | ccc@ccc.ccc | M | 28 |
| ddddd | 도대담 | dd | ddd@ddd.ddd | M | 35 |
| eeeee | 류라라 | ee | eee@eee.eee | F | 17 |
| fffff | 문미미 | ff | fff@fff.fff | F | 20 |
| ggggg | 박보배 | gg | ggg@ggg.ggg | M | 26 |
| hhhhh | 송성실 | hh | hhh@hhh.hhh | M | 41 |
| iiii | 윤예의 | ii | iii@iii.iii | F | 28 |
| jjjjj | 정재주 | jj | jjj@jjj.jjj | M | 23 |

2. 구현 클래스 다이어그램 (Class Diagram)

| Member |
|--|
| - id : String - name : String - password : String - email : String - gender : char - age : int |
| + Member() + Member(id:String, name:String, password:String, email:String, gender:char age:int) + setter() / getter() + inform() : String |

| Run |
|-------------------------------------|
| |
| + <u>main(args:String[]) : void</u> |

| MemberMenu |
|---|
| - sc : Scanner = new Scanner(System.in) - MemberController mc = new MemberController |
| + MemberMenu () + mainMenu() : void + insertMember() : void + searchMember() : void + searchId() : void + searchName() : void + searchEmail() : void + updateMember() : void + updatePassword() : void + updateName() : void + updateEmail() : void + deleteMember() : void + deleteOne() : void + deleteAll() : void + printAll() : void |

| MemberController |
|--|
| - Member[] m = new Member[SIZE] + <u>SIZE : int = 10</u> |
| + existMemberNum() : int + checkId(inputId:String) : Boolean + insertMember(id:String, name:String, password:String, email:String, gender:String, age:int) : void + searchId(id:String) : String + searchName(name:String) : Member[] + searchEmail(email:String) : Member[] + updatePassword(id:String, password:String):Boolean + updateName(id:String, name:String):Boolean + updateEmail(id:String, email:String):Boolean + delete(id:String):Boolean + delete() : void + printAll() : Member[] |

3. 구현 클래스 설명

| Package명 | Class명 | Method | 설명 |
|-------------------------------------|------------------|--|----------------------------|
| com.kh.hw .member .model.vo | Member | +Member() | 기본 생성자 |
| | | +Member(id:String, name:String, password:String, email:String, gender:char, age:int) | 6개의 초기값을 받는 매개변수 있는 생성자 |
| | | +getXXX() | 필드의 값을 리턴하는 메소드 |
| | | +setXXX() | 필드의 값을 초기화/변경하는 메소드 |
| | | +inform():String | Member 클래스의 모든 필드값들을 합쳐 반환 |
| com.kh.hw .member .controller | MemberController | +existMemberNum:int | 현재 존재하는 멤버 수 반환 |
| | | +checkId(inputId:String) : boolean | 아이디 중복확인 하는 메소드 |
| | | +insertMember(id:String, name:String, password:String, email:String, gender:char, age:int) | Member객체를 객체배열에 저장하는 메소드 |
| | | +searchId(id:String):String | id로 회원을 조회하는 메소드 |
| | | +searchName(name:String):Member[] | 이름으로 회원을 조회하는 메소드 |
| | | +searchEmail(email:String):Member[] | 이메일로 회원을 조회하는 메소드 |
| | | +updatePassword(id:String, password:String):boolean | 비밀번호 변경 메소드 |
| | | +updateName(id:String, name:String):boolean | 이름 변경 메소드 |
| | | +updateEmail(id:String, email:String):boolean | 이메일 변경 메소드 |
| | | +delete(id:String):boolean | 한 회원만 삭제하는 메소드 |
| | | +delete():void | 전체 회원을 삭제하는 메소드 |
| | | +printAll():Member[] | Member객체 반환 메소드 |

| | | | |
|---------------------------|------------|---------------------------|-------------------------------------|
| com.kh.hw .member.view | MemberMenu | +MemberMenu() | 기본생성자 |
| | | +mainMenu():void | 메인 메뉴 출력 메소드 |
| | | +insertMember():void | 회원등록하기 위한 데이터를 사용자에게 입력 받는 메소드 |
| | | +searchMember():void | 검색 메뉴 출력 메소드 |
| | | +searchId():void | 아이디 검색을 위한 데이터를 사용자에게 입력 받는 메소드 |
| | | +searchName():void | 이름 검색을 위한 데이터를 사 용자에게 입력 받는 메소드 |
| | | +searchEmail():void | 이메일 검색을 위한 데이터를 사용자에게 입력 받는 메소드 |
| | | +updateMember():void | 수정 메뉴 출력 메소드 |
| | | +updatePassword():void | 비밀번호 수정을 위한 데이터를 사용자에게 입력 받는 메소드 |
| | | +updateName():void | 이름 수정을 위한 데이터를 사 용자에게 입력 받는 메소드 |
| | | +updateEmail():void | 이메일 수정을 위한 데이터를 사용자에게 입력 받는 메소드 |
| | | +deleteMember():void | 삭제 메뉴 출력 메소드 |
| | | +deleteOne():void | 하나의 회원 삭제 결과를 출력 하는 메소드 |
| | | +deleteAll():void | 전체 회원 삭제 결과를 출력하 는 메소드 |
| com.kh.hw. member.run | Run | +main(args:String[]):void | MemberMenu의 mainMenu메 소드 실행 |

* class 명과 method 명은 변경 하지 않는다.

* 모든 클래스 변수의 getter, setter 함수는 직접 구현한다.

4. class 구조

```
public class Run{

    public static void main(String args[]) {

        MemberMenu mm = new MemberMenu();

        mm.mainMenu();

    }
```

```
public class MemberMenu{

    // 멤버 필드

    public void mainMenu() {

        최대 등록 가능한 회원 수는 10명입니다.

        현재 등록된 회원 수는 0명입니다.

        // MemberController(mc)클래스 안에 있는 멤버필드를 이용하여

        // 최대 등록 가능한 회원 수가 몇 명인지 출력

        // 또한 현재 등록된 회원 수는 mc에 existMemberNum()의 반환 값으로

        // 몇 명이 등록되어 있는지 출력


        // 현재 등록된 회원 수에 따라 메인 메뉴를 다르게 출력

        // 현재 등록된 회원 수가 10명이 아닐 때

        1. 새 회원 등록          ➔ insertMember()

        2. 회원 검색            ➔ searchMember()

        3. 회원 정보 수정        ➔ updateMemner()

        4. 회원 삭제             ➔ deleteMember()

        5. 모두 출력            ➔ printAll()

        9. 끝내기

        메뉴 번호 :
```

```

// 현재 등록된 회원 수가 10명일 때

회원 수가 모두 꽉 찼기 때문에 일부 메뉴만 오픈됩니다.

2. 회원 검색          → searchMember()

3. 회원 정보 수정      → updateMemner()

4. 회원 삭제          → deleteMember()

5. 모두 출력          → printAll()

9. 끝내기

메뉴 번호 :

// 두 경우 모두 끝내기를 누르면 "프로그램을 종료합니다" 출력 후

// 프로그램 종료하고 메뉴 번호에 없는 번호를 누를 시

// "잘못 입력하셨습니다. 다시 입력해주세요." 출력 후 메뉴 반복
}

public void insertMember(){

    // 회원 등록에 필요한 아이디, 이름, 비밀번호, 이메일, 성별, 나이를

    // 사용자에게 직접 입력 받되 아이디의 경우 기존 회원의 아이디와

    // 중복되지 않도록 mc의 checkId()메소드를 통해서 중복 검사를 함

    // 중복 검사를 한 결과 값에 따라 중복 값이 없으면 다음으로 넘어가고

    // 있으면 "중복된 아이디입니다. 다시 입력해주세요."라는 문구를 출력 후

    // 다시 아이디를 받도록 함

    // 성별의 경우 M이나 m, F나 f가 아니면 "성별을 다시 입력하세요."를

    // 출력 후 다시 성별을 입력하도록 함

    // 아이디부터 나이까지 모든 데이터를 받았으면

    // mc의 insertMember메소드의 매개변수로 넘김
}

```

```
public void searchMember(){
```

```
    1. 아이디로 검색하기    ➔ searchId()
```

```
    2. 이름으로 검색하기    ➔ searchName()
```

```
    3. 이메일로 검색하기    ➔ searchEmail)_
```

```
    9. 메인으로 돌아가기
```

```
    메뉴 번호 :
```

```
    // 메뉴 번호에 따라 각 메소드로 이동되며 9번을 입력했을 경우
```

```
    // "메인으로 돌아갑니다." 출력 후 mainMenu()로 돌아감
```

```
    // 메뉴 번호를 잘못 입력했을 경우 "잘못 입력하셨습니다." 출력 후
```

```
    // mainMenu()로 돌아감
```

```
}
```

```
public void searchId(){
```

```
    // 검색할 아이디를 사용자에게 입력 받고 입력 받은 데이터를
```

```
    // mc의 searchId() 메소드의 매개변수로 넘김, 반환 값에 따라
```

```
    // 검색 결과 없으면 "검색 결과가 없습니다." 출력 후 mainMenu()로 돌아감
```

```
    // 검색 결과가 있으면 "찾으신 회원 조회 결과입니다." 출력 후
```

```
    // 회원 검색 결과 출력
```

```
}
```

```
public void searchName(){
```

```
    // 검색할 이름을 사용자에게 입력 받고 입력 받은 데이터를
```

```
    // mc의 searchName() 메소드의 매개변수로 넘김, 반환 값에 따라
```

```
    // 검색 결과 없으면 "검색 결과가 없습니다." 출력 후 mainMenu()로 돌아감
```

```
    // 검색 결과가 있으면 "찾으신 회원 조회 결과입니다." 출력 후
```

```
    // 회원 검색 결과 출력
```

```
}
```

```

public void searchEmail(){

    // 검색할 아이디를 사용자에게 입력 받고 입력 받은 데이터를

    // mc의 searchEmail() 메소드의 매개변수로 넘김, 반환 값에 따라

    // 검색 결과 없으면 "검색 결과가 없습니다." 출력 후 mainMenu()로 돌아감

    // 검색 결과가 있으면 "찾으신 회원 조회 결과입니다." 출력 후

    // 회원 검색 결과 출력

}

public void updateMember(){

    1. 비밀번호 수정하기    ➔ updatePassword()

    2. 이름 수정하기        ➔ updateName()

    3. 이메일 수정하기      ➔ updateEmail()

    9. 메인으로 돌아가기

    // 메뉴 번호에 따라 각 메소드로 이동되며 9번을 입력했을 경우

    // "메인으로 돌아갑니다." 출력 후 mainMenu()로 돌아감

    // 메뉴 번호를 잘못 입력했을 경우 "잘못 입력하셨습니다." 출력 후

    // mainMenu()로 돌아감

}

public void updatePassword(){

    // 수정할 회원 id와 비밀번호를 사용자에게 입력 받고 입력 받은 데이터를

    // mc의 updatePassword() 메소드의 매개변수로 넘김, 반환 값에 따라

    // 검색결과 없으면 "존재하지 않는 아이디입니다." 출력, mainMenu()로 감

    // 검색 결과가 있으면 "수정이 성공적으로 되었습니다." 출력 후

    // mainMenu()로 돌아감

}

```



```

public void updateName(){

    // 수정할 회원 id와 이름을 사용자에게 입력 받고 입력 받은 데이터를

    // mc의 updateName() 메소드의 매개변수로 넘김, 반환 값에 따라

    // 검색결과 없으면 "존재하지 않는 아이디입니다." 출력, mainMenu()로 감

    // 검색 결과가 있으면 "수정이 성공적으로 되었습니다." 출력 후

    // mainMenu()로 돌아감

}

public void updateEmail(){

    // 수정할 회원 id와 이메일을 사용자에게 입력 받고 입력 받은 데이터를

    // mc의 updateEmail() 메소드의 매개변수로 넘김, 반환 값에 따라

    // 검색결과 없으면 "존재하지 않는 아이디입니다." 출력, mainMenu()로 감

    // 검색 결과가 있으면 "수정이 성공적으로 되었습니다." 출력 후

    // mainMenu()로 돌아감

}

public void deleteMember(){

    1. 특정 회원 삭제하기    ➔ deleteOne()

    2. 모든 회원 삭제하기    ➔ deleteAll()

    9. 메인으로 돌아가기

    // 메뉴 번호에 따라 각 메소드로 이동되며 9번을 입력했을 경우

    // "메인으로 돌아갑니다." 출력 후 mainMenu()로 돌아감

    // 메뉴 번호를 잘못 입력했을 경우 "잘못 입력하셨습니다." 출력 후

    // mainMenu()로 돌아감

}

```

```

public void deleteOne(){

    // 삭제할 회원 id를 사용자에게 입력 받고 정말 삭제할 것인지 사용자에게
    // 물어본 뒤, Y나 y를 사용자가 입력할 경우 입력 받은 id를
    // mc의 delete() 메소드의 매개변수로 넘김, 반환 값에 따라
    // 검색결과 없으면 "존재하지 않는 아이디입니다." 출력, mainMenu()로 감
    // 검색 결과가 있으면 "성공적으로 삭제하였습니다." 출력 후

    // mainMenu()로 돌아감

}

public void deleteAll(){

    // 정말 삭제할 것인지 사용자에게 물어본 뒤, Y나 y를 사용자가 입력할 경우
    // mc의 delete() 메소드 호출, "성공적으로 삭제하였습니다." 출력 후

    // mainMenu()로 돌아감

}

public void printAll(){

    // mc의 printAll() 메소드의 반환 값을 가지고 저장된 회원을 출력하는데
    // 저장된 회원의 수가 0명이면 "저장된 회원이 없습니다." 출력,
    // 0명이 아니면 저장된 모든 회원의 정보 출력

}

}

```

5. 실행 결과

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 0명입니다.

1. 새 회원 등록
2. 회원 검색
3. 회원 정보 수정
4. 회원 삭제
5. 모두 출력
9. 끝내기

메뉴 번호 : 1

새 회원을 등록합니다.

아이디 : aaaaa

이름 : 박신우

비밀번호 : aa

이메일 : aaa@aaa.aaa

성별(M/F) : k

성별을 다시 입력하세요.

성별(M/F) : F

나이 : 33

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 1명입니다.

1. 새 회원 등록
2. 회원 검색
3. 회원 정보 수정
4. 회원 삭제
5. 모두 출력
9. 끝내기

메뉴 번호 : 1

새 회원을 등록합니다.

아이디 : aaaaa

중복된 아이디입니다. 다시 입력해주세요.

아이디 : bbbbb

이름 : 강고결

비밀번호 : bb

이메일 : bbb@bbb.bbb

성별(M/F) : m

나이 : 34

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 2명입니다.

1. 새 회원 등록
2. 회원 검색
3. 회원 정보 수정
4. 회원 삭제
5. 모두 출력
9. 끝내기

메뉴 번호 : 1

(중간 생략 ... 위의 샘플 데이터 10개를 모두 넣어줌)

메뉴 번호 : 1

새 회원을 등록합니다.

아이디 : jjjjj

이름 : 정재주

비밀번호 : jj

이메일 : jjj@jjj.jjj

성별(M/F) : M

나이 : 23

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 10명입니다.

회원 수가 모두 꽉 찼기 때문에 일부 메뉴만 오픈됩니다.

2. 회원 검색

3. 회원 정보 수정

4. 회원 삭제

5. 모두 출력

9. 끝내기

메뉴 번호 : 3

1. 비밀번호 수정하기

2. 이름 수정하기

3. 이메일 수정하기

9. 메인으로 돌아가기

메뉴 번호 : 1

수정할 회원의 아이디 : ㅎㅎㅎㅎ

수정할 비밀번호 : ㅎㅎ

존재하지 않는 아이디입니다.

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 10명입니다.

회원 수가 모두 꽉 찼기 때문에 일부 메뉴만 오픈됩니다.

2. 회원 검색

3. 회원 정보 수정

4. 회원 삭제

5. 모두 출력

9. 끝내기

메뉴 번호 : 5

aaaaa 박신우 aa aaa@aaa.aaa F 33

bbbbbb 강고결 bb bbb@bbb.bbb m 34

ccccc 남나눔 cc ccc@ccc.ccc M 28

dddddd 도대담 dd ddd@ddd.ddd M 35

eeeeee 류라라 ee eee@eee.eee f 17

ffffff 문미미 ff fff@fff.fff F 20

ggggg 박보배 gg ggg@ggg.ggg m 26

hhhhh 송성실 hh hhh@hhh.hhh m 41

iiiii 윤예의 ii iii@iii.iii F 28

jjjjj 정재주 jj jjj@jjj.jjj M 23

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 10명입니다.

회원 수가 모두 꽉 찼기 때문에 일부 메뉴만 오픈됩니다.

- 2. 회원 검색
- 3. 회원 정보 수정
- 4. 회원 삭제
- 5. 모두 출력
- 9. 끝내기

메뉴 번호 : 3

- 1. 비밀번호 수정하기
- 2. 이름 수정하기
- 3. 이메일 수정하기
- 9. 메인으로 돌아가기

메뉴 번호 : 1

수정할 회원의 아이디 : gggggg

수정할 비밀번호 : ㅎㅎ

수정이 성공적으로 되었습니다.

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 10명입니다.

회원 수가 모두 꽉 찼기 때문에 일부 메뉴만 오픈됩니다.

- 2. 회원 검색
- 3. 회원 정보 수정
- 4. 회원 삭제
- 5. 모두 출력
- 9. 끝내기

메뉴 번호 : 2

- 1. 아이디로 검색하기
- 2. 이름으로 검색하기
- 3. 이메일로 검색하기
- 9. 메인으로 돌아가기

메뉴 번호 : 9

메인으로 돌아갑니다.

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 10명입니다.

회원 수가 모두 꽉 찼기 때문에 일부 메뉴만 오픈됩니다.

- 2. 회원 검색
- 3. 회원 정보 수정
- 4. 회원 삭제
- 5. 모두 출력
- 9. 끝내기

메뉴 번호 : 3

1. 비밀번호 수정하기
2. 이름 수정하기
3. 이메일 수정하기
9. 메인으로 돌아가기

메뉴 번호 : 2

수정할 회원의 아이디 : jjjjj

수정할 이름 : 박신우

수정이 성공적으로 되었습니다.

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 10명입니다.

회원 수가 모두 꽉 찼기 때문에 일부 메뉴만 오픈됩니다.

2. 회원 검색
3. 회원 정보 수정
4. 회원 삭제
5. 모두 출력
9. 끝내기

메뉴 번호 : 3

1. 비밀번호 수정하기
2. 이름 수정하기
3. 이메일 수정하기
9. 메인으로 돌아가기

메뉴 번호 : 2

수정할 회원의 아이디 : bbbbb

수정할 이름 : 박신우

수정이 성공적으로 되었습니다.

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 10명입니다.

회원 수가 모두 꽉 찼기 때문에 일부 메뉴만 오픈됩니다.

2. 회원 검색
3. 회원 정보 수정
4. 회원 삭제
5. 모두 출력
9. 끝내기

메뉴 번호 : 3

1. 비밀번호 수정하기
2. 이름 수정하기
3. 이메일 수정하기
9. 메인으로 돌아가기

메뉴 번호 : 3

수정할 회원의 아이디 : fffff

수정할 이메일 : aaa@aaa.aaa

수정이 성공적으로 되었습니다.

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 10명입니다.

회원 수가 모두 꽉 찼기 때문에 일부 메뉴만 오픈됩니다.

2. 회원 검색
3. 회원 정보 수정
4. 회원 삭제
5. 모두 출력
9. 끝내기

메뉴 번호 : 5

aaaaa 박신우 aa aaa@aaa.aaa F 33
bbbbb 박신우 bb bbb@bbb.bbb m 34
ccccc 남나눔 cc ccc@ccc.ccc M 28
ddddd 도대담 dd ddd@ddd.ddd M 35
eeeee 류라라 ee eee@eee.eee f 17
fffff 문미미 ff aaa@aaa.aaa F 20
ggggg 박보배 ㅎㅎ ggg@ggg.ggg m 26
hhhhh 송성실 hh hhh@hhh.hhh m 41
iiiii 윤예의 ii iii@iii.iii F 28
jjjjj 박신우 jj jjj@jjj.jjj M 23

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 10명입니다.

회원 수가 모두 꽉 찼기 때문에 일부 메뉴만 오픈됩니다.

2. 회원 검색
3. 회원 정보 수정
4. 회원 삭제
5. 모두 출력
9. 끝내기

메뉴 번호 : 2

1. 아이디로 검색하기
2. 이름으로 검색하기
3. 이메일로 검색하기
9. 메인으로 돌아가기

메뉴 번호 : 1

검색할 아이디 : iiiii

찾으신 회원 검색 결과입니다.

iiiii 윤예의 ii iii@iii.iii F 28

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 10명입니다.

회원 수가 모두 꽉 찼기 때문에 일부 메뉴만 오픈됩니다.

- 2. 회원 검색
- 3. 회원 정보 수정
- 4. 회원 삭제
- 5. 모두 출력
- 9. 끝내기

메뉴 번호 : 2

- 1. 아이디로 검색하기
- 2. 이름으로 검색하기
- 3. 이메일로 검색하기
- 9. 메인으로 돌아가기

메뉴 번호 : 2

검색할 이름 : 박신우

찾으신 회원 검색 결과입니다.

aaaaa 박신우 aa aaa@aaa.aaa F 33

bbbbbb 박신우 bb bbb@bbb.bbb m 34

jjjjj 박신우 jj jjj@jjj.jjj M 23

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 10명입니다.

회원 수가 모두 꽉 찼기 때문에 일부 메뉴만 오픈됩니다.

- 2. 회원 검색
- 3. 회원 정보 수정
- 4. 회원 삭제
- 5. 모두 출력
- 9. 끝내기

메뉴 번호 : 2

- 1. 아이디로 검색하기
- 2. 이름으로 검색하기
- 3. 이메일로 검색하기
- 9. 메인으로 돌아가기

메뉴 번호 : 3

검색할 이메일 : aaa@aaa.aaa

찾으신 회원 검색 결과입니다.

aaaaa 박신우 aa aaa@aaa.aaa F 33

fffff 문미미 ff aaa@aaa.aaa F 20

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 10명입니다.

회원 수가 모두 꽉 찼기 때문에 일부 메뉴만 오픈됩니다.

- 2. 회원 검색
- 3. 회원 정보 수정
- 4. 회원 삭제
- 5. 모두 출력
- 9. 끝내기

메뉴 번호 : 4

1. 특정 회원 삭제하기

2. 모든 회원 삭제하기

9. 메인으로 돌아가기

메뉴 번호 : 1

삭제할 회원의 아이디 : ggggg

정말 삭제하시겠습니까?(y/n) : y

성공적으로 삭제하였습니다.

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 9명입니다.

1. 새 회원 등록

2. 회원 검색

3. 회원 정보 수정

4. 회원 삭제

5. 모두 출력

9. 끝내기

메뉴 번호 : 5

aaaaa 박신우 aa aaa@aaa.aaa F 33

bbbbb 박신우 bb bbb@bbb.bbb m 34

ccccc 남나눔 cc ccc@ccc.ccc M 28

ddddd 도대담 dd ddd@ddd.ddd M 35

eeeee 류라라 ee eee@eee.eee f 17

ffffff 문미미 ff aaa@aaa.aaa F 20

hhhhh 송성실 hh hhh@hhh.hhh m 41

iiiii 윤예의 ii iii@iii.iii F 28

jjjjj 박신우 jj jjj@jjj.jjj M 23

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 9명입니다.

1. 새 회원 등록

2. 회원 검색

3. 회원 정보 수정

4. 회원 삭제

5. 모두 출력

9. 끝내기

메뉴 번호 : 4

1. 특정 회원 삭제하기

2. 모든 회원 삭제하기

9. 메인으로 돌아가기

메뉴 번호 : 1

삭제할 회원의 아이디 : bbbbb

정말 삭제하시겠습니까?(y/n) : n

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 9명입니다.

1. 새 회원 등록
2. 회원 검색
3. 회원 정보 수정
4. 회원 삭제
5. 모두 출력
9. 끝내기

메뉴 번호 : 4

1. 특정 회원 삭제하기
2. 모든 회원 삭제하기
9. 메인으로 돌아가기

메뉴 번호 : 9

메인으로 돌아갑니다.

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 9명입니다.

1. 새 회원 등록
2. 회원 검색
3. 회원 정보 수정
4. 회원 삭제
5. 모두 출력
9. 끝내기

메뉴 번호 : 2

1. 아이디로 검색하기
2. 이름으로 검색하기
3. 이메일로 검색하기
9. 메인으로 돌아가기

메뉴 번호 : 9

메인으로 돌아갑니다.

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 9명입니다.

1. 새 회원 등록
2. 회원 검색
3. 회원 정보 수정
4. 회원 삭제
5. 모두 출력
9. 끝내기

메뉴 번호 : 4

1. 특정 회원 삭제하기
2. 모든 회원 삭제하기
9. 메인으로 돌아가기

메뉴 번호 : 1

삭제할 회원의 아이디 : bbbbbb

정말 삭제하시겠습니까?(y/n) : y

성공적으로 삭제하였습니다.

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 8명입니다.

1. 새 회원 등록
2. 회원 검색
3. 회원 정보 수정
4. 회원 삭제
5. 모두 출력
9. 끝내기

메뉴 번호 : 5

aaaaa 박신우 aa aaa@aaa.aaa F 33
ccccc 남나눔 cc ccc@ccc.ccc M 28
ddddd 도대담 dd ddd@ddd.ddd M 35
eeeee 류라라 ee eee@eee.eee f 17
fffff 문미미 ff aaa@aaa.aaa F 20
hhhhh 송성실 hh hhh@hhh.hhh m 41
iiiii 윤예의 ii iii@iii.iii F 28
jjjjj 박신우 jj jjj@jjj.jjj M 23

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 8명입니다.

1. 새 회원 등록
2. 회원 검색
3. 회원 정보 수정
4. 회원 삭제
5. 모두 출력
9. 끝내기

메뉴 번호 : 2

1. 아이디로 검색하기
2. 이름으로 검색하기
3. 이메일로 검색하기
9. 메인으로 돌아가기

메뉴 번호 : 2

검색할 이름 : 박신우

찾으신 회원 검색 결과입니다.

aaaaa 박신우 aa aaa@aaa.aaa F 33
jjjjj 박신우 jj jjj@jjj.jjj M 23

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 8명입니다.

1. 새 회원 등록
2. 회원 검색
3. 회원 정보 수정
4. 회원 삭제
5. 모두 출력
9. 끝내기

메뉴 번호 : 4

1. 특정 회원 삭제하기

2. 모든 회원 삭제하기

9. 메인으로 돌아가기

메뉴 번호 : 2

정말 삭제하시겠습니까?(y/n) : y

성공적으로 삭제하였습니다.

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 0명입니다.

1. 새 회원 등록

2. 회원 검색

3. 회원 정보 수정

4. 회원 삭제

5. 모두 출력

9. 끝내기

메뉴 번호 : 5

저장된 회원이 없습니다.

최대 등록 가능한 회원 수는 10명입니다.

현재 등록된 회원 수는 0명입니다.

1. 새 회원 등록

2. 회원 검색

3. 회원 정보 수정

4. 회원 삭제

5. 모두 출력

9. 끝내기

메뉴 번호 : 9

프로그램을 종료합니다.