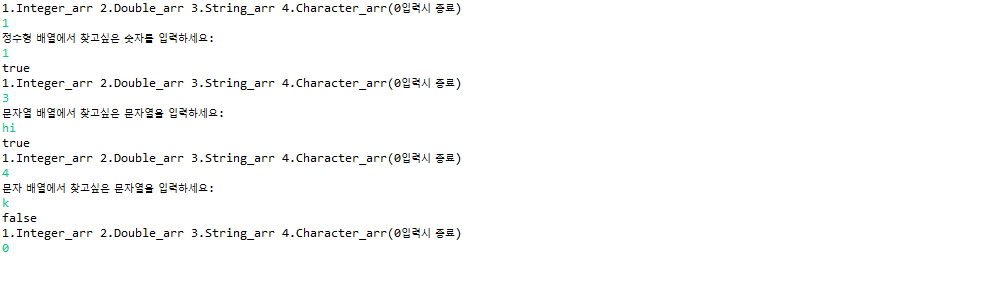
1. ArrayList를 이용한 기초적인 프로그램을 작성해보자.(메인 클래스에서 바로 작성)(기초)

(본 프로그램은 ArrayList에 내가 찾는 값이 있는지 없는지 찾는 프로그램이다. 프로그램의 작성방법은 다양하니 아래에 지문과 다르게 작성해도 결과만 같게 나오면 된다.)

* 1. 랜덤 객체를 생성해라.
  2. ArrayList를 4개 (정수형, 실수형, 문자열, 문자형)를 생성하고 각 List에 임의의 값을 초기화 해줘라
  3. 무한 반복을 돌려라
     1. System.***out***.println("1.Integer\_arr 2.Double\_arr 3.String\_arr 4.Character\_arr(0입력시 종료)");
     2. 0번을 제외한 메뉴를 고르면 값을 입력 받고 list 내장 메소드를 이용해서 배열에 있는 원소 인지 아닌지 판단해라



1. ArrayList를 이용한 프로그램을 작성해보자. (메인클래스에서 바로 작성해라)(기초~중 사이)

(본 프로그램은 ArrayList와 반복자를 사용해서 특정 문자를 입력하기 전까지 계속 입력 받아 입력된 문자를 합쳐 반복자를 이용해 출력해보고 다시 반복문을 이용해 역순 출력하는 프로그램이다. 아래 지문과 다르게 작성해도 결과만 나오면 된다.)

* 1. 문자 타입 ArrayList를 생성해라
  2. 문자 타입 배열을 선언해라
  3. 무한 반복문을 돌려라
     1. 원하는 문자를 입력 받아라
     2. q또는Q가 나오면 프로그램은 종료된다.
     3. 입력 받은 문자를 list에 추가해라
  4. 무한반복문이 종료 된 이후
     1. 문자 배열을 ArrayList의 사이즈 만큼 생성해줘라
     2. 반복자를 선언하고 ArrayList를 반복자화 시켜줘라
     3. 반복문을 돌려 반복자에 원소가 있다면
        1. 문자 배열에 반복자를 이용하여 차례대로 원소를 넣어줘라
     4. 문자배열을 역순 출력해라



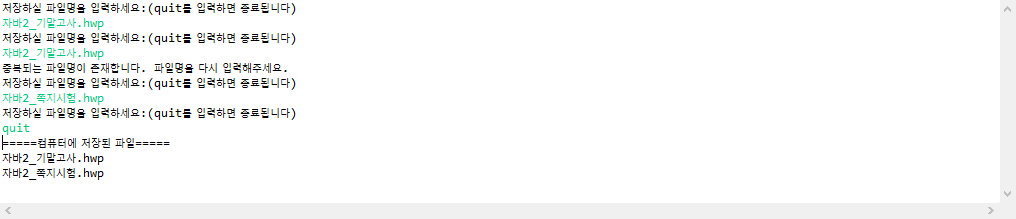
1. Set을 이용한 파일 저장 프로그램을 작성해바라 (초 기초)

(본 프로그램은 set의 특성을 이용한 프로그램이다. Set의 특성은 중복 되는 값을 set에 추가할 수 없다. 그 특성을 알려주기 위한 문제이다. 아래 지문과 달라도 결과만 나오면 된다.(하지만 이 문제는 정답이 정확하게 있다.))

* 1. Set에 있는 원소들을 알파벳 순으로 정렬해주는 set을 문자열 타입으로 생성해라

(set에는 저장할 파일명을 입력 받을것이다.)

* 1. 무한반복문을 돌려라
     1. 저장할 파일명을 입력 받아라 (문자열로 입력 받을 것)
     2. Quit를 입력하면 반복문을 종료해라
     3. 만일 저장할 파일명이 이미 set에 있다면 System.***out***.println("중복되는 파일명이 존재합니다. 파일명을 다시 입력해주세요."); 라고 출력해주고 반복문으로 다시 돌아가라.
     4. 파일명이 중복되지 않는다면 set에 추가해줘라
  2. 무한반복문을 나갔다면
     1. 반복자 객체를 선언 후 set을 반복자화 시켜라
     2. Set에 저장된 모든 파일명을 출력해줘라



1. 링크드리스트를 이용한 프로그램을 작성해보자.(상급)

(이 문제는 만들다 보니 무척 어려워졌다….. 만드는데 엄청 오래 걸림….. 그러니까 모르겠으면 나를 귀찮게 하기를)

(링크드리스트를 이용해 은행App을 만드는 프로그램이다.)

(링크드리스트와 set의 차이점은 중복 값을 허용하는 것이다.)

(이 프로그램은 다양한 답이 나올 거 같다. 본인의 뜻이 있다면 꼭 링크드리스트를 사용하지 않아도 된다.)

* 1. Client 클래스를 생성해라
     1. 정수형 변수인 pay와 예금주 변수인 name을 클래스 내부에서만 사용할 수 있게 선언하고 name을 외부에서 사용할 수 있는 get메소드를 만들어라
     2. 정수형 변수인 temp 변수를 내부에서만 사용할 수 있게 선언해라(입,출금 금액 임시 저장 변수)
     3. client타입의 LinkedList를 선언해라
     4. client 생성자를 만들어라
        1. 매개변수는 문자열 타입의 이름 변수와 client타입의 LinkedList 변수를 받아라
        2. 필드에 선언한 이름과 list에 삽입해라
     5. 반환형이 없는 insert 메소드를 만들어라
        1. 매개변수는 정수형 타입의 인덱스 변수를 받아라
        2. System.***out***.println("입금하실 금액을 넣어주세요:");
        3. 입금 금액을 입력 받아라.
        4. 필드에 선언한 temp 변수에 list의 인덱스에 해당하는 pay를 저장해라
        5. 리스트에 인덱스에 해당하는 pay에 입금 금액을 추가해줘라
           1. 힌트 : List.get(index).pay = ~~~~~~~
     6. 반환형이 없는 withdraw 메소드를 만들어라
        1. 매개변수는 정수형 타입의 인덱스 변수를 받아라
        2. 출금할 금액을 입력 받아라
        3. 만일 출금 하려는 금액이 해당 고객의 잔고보다 많다면 잔고가 부족함을 알리고 부족하지 않으면 기존 잔고에서 출금 금액을 빼준다.
           1. 힌트 : List.get(index).pay = ~~~~~~
     7. 반환형이 없는 search 메소드를 만들어라
        1. 매개변수는 정수형 타입의 인덱스 변수를 받아라
        2. 현재 잔고를 출력해줘라
  2. test클래스를 생성해라(메인 클래스)
     1. client 타입의 LinkedList를 생성해라
     2. 반복문에 사용할 int I 를 0으로 초기화 해줘라(예금주 존재 여부 확인 위해)
        1. 검색하려는 데이터가 있는지 없는지 판단 위해 나는 이렇게 반복문 변수를 밖으로 빼줬음
        2. 중복 확인 방법은 여러 가지여서 이 부분은 충분히 다를 수 있음
     3. 예금주 검색을 위해 계속 쓰일 문자열 변수 name을 선언 후 null로 초기화 해줘라
     4. 무한 반복문 진행

System.***out***.println("어서오세요.인규모바일뱅크 입니다.");

System.***out***.println("회원가입을 하셔야만 서비스 이용 가능합니다.");

System.***out***.println("1.서비스이용 2.회원가입 3.종료");

* + - 1. 메뉴를 입력 받고 3번이 입력되면 바로 프로그램은 종료된다.
    1. 1번이 입력될 경우

System.***out***.println("원하시는 메뉴를 선택해주세요.");

System.***out***.println("1.입금 2.출금 3.예금 조회");

* + - 1. 메뉴를 입력 받고 1번이 입력되면 입금할 예금주를 name변수를 이용해 입력 받고
         1. 반복문을 이용해 linkedlist 안에 입력한 예금주가 존재하는지 확인을 한다.

힌트 : list.get(i).~~~~~~~

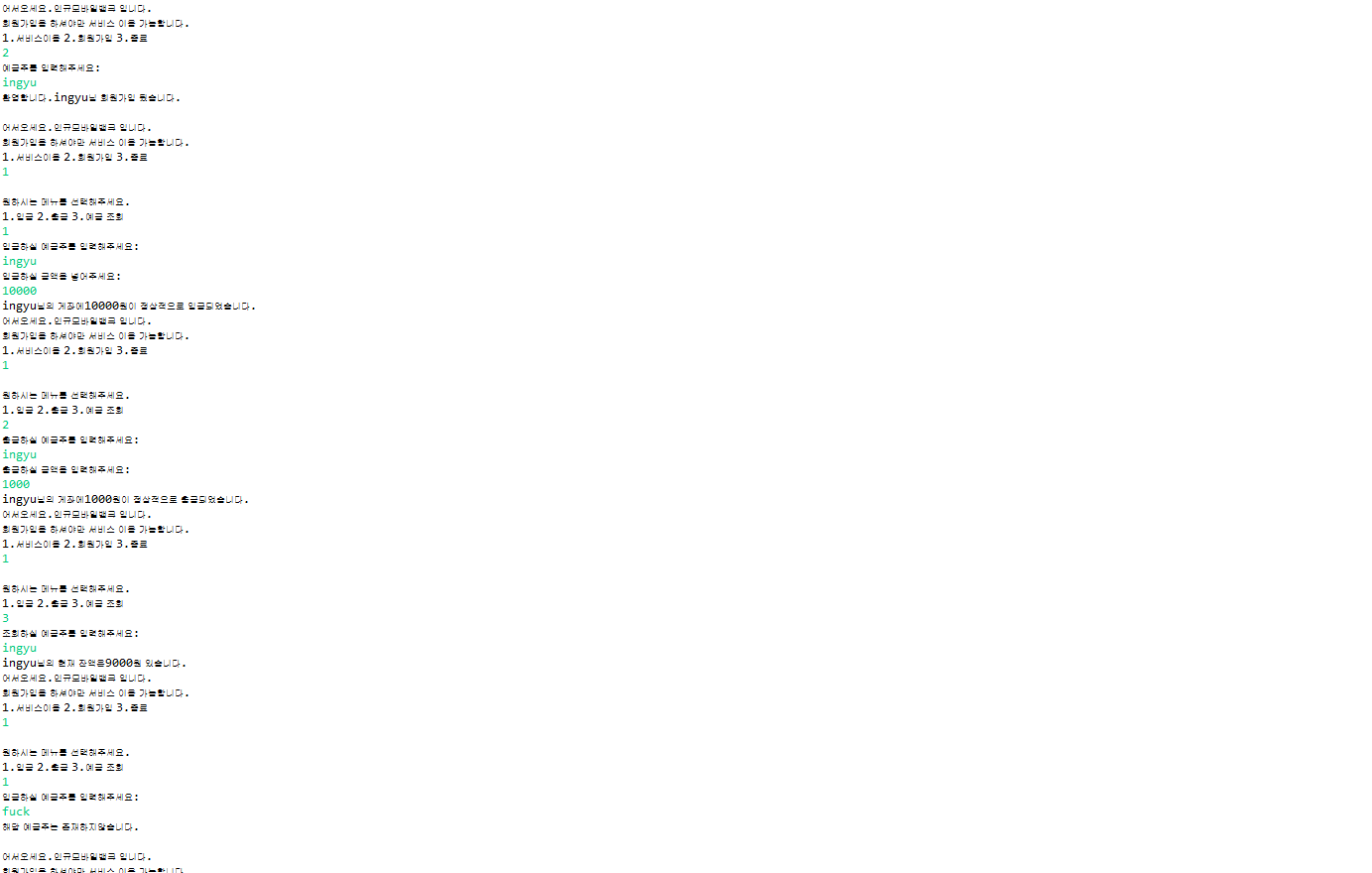
* + - * 1. 만일 예금주가 존재한다면 존재함을 발견한 해당 반복문 변수를 함께 넘겨주면서 insert 메소드를 호출 후 반복문을 정지시켜라

왜냐 이미 내가 예금주를 찾았기 때문에 더 이상 검색할 필요가 없다.

* + - * 1. 만일 다 반복문을 다 돌았는데도 예금주가 없다면 예금주가 없다고 알려줘라.

case1번과 case 2,3번은 동일한 원리로 작동한다.(메소드 호출부분만 다를 뿐 case1번과 case2, 3번 소스는 모두 동일하다)

* + 1. 2번이 입력될 경우 (가장 중요 여기가 똑바로 코딩 되지 않는다면 프로그램은 정상작동 하지 않을 것이다.)
       1. System.***out***.println("예금주를 입력해주세요:");
       2. name변수에 이름을 입력받고
       3. 위의 케이스문들과 동일한 원리로 중복된 예금주가 있는지 없는지 판단해준다.
       4. 중복된 예금주가 없다면
       5. LinkedList 객체에 client객체를 추가해줘라
          1. 우리가 앞서 생성자를 만들 때 이름과 링크드리스트를 넘겨주는 것으로 만들었다. 이것을 이용하면 된다.
          2. 힌트 : list.add(new client(이름,리스트)); (정답 그 자체이다…)



1. Map을 이용하여 욕을 몇 번하는지 체크하는 프로그램을 작성해보자 (기초)

(메인 클래스에 바로 작성 할 것)

(맵의 구조와 사용법을 알려주기 위한 목적)

* 1. Map객체를 두개 생성해라(key값의 타입은 문자열이고 value값의 타입은 정수형이다.)
  2. 욕을 제외한 단어들을 저장할 문자열 타입의 ArrayList를 생성해라
  3. 욕을 저장할 문자열 타입의 ArrayList를 생성해라
     1. 해당 ArrayList에는 아무 욕 3가지 정도 추가해놔라
     2. Ex.) ㄲㅈ, ㅅㅂ, ㅂㅅ
  4. 욕 일 때 저장할 ArrayList를 생성해라
  5. 무한반복문을 진행해라
     1. 문자열을 입력 받아라
     2. Quit를 입력 받으면 프로그램은 종료된다.
     3. 만약 입력한 문자열이 저장해 놓은 욕과 같다면 욕을 저장할 ArrayList에 넣어라
     4. 욕이 아니라면 욕이 아닌 문자열을 저장할 ArrayList에 추가해라
  6. 무한반복문이 종료되면
     1. 욕이 아닌 문자열 Map과 욕을 카운트한 Map에 key값에 대해 value값을 카운트해라
  7. Map을 출력해라



문제 6. Map 을 이용한 문자 보내기

* Rx 인터페이스

**void** ReceiveMessage(LinkedList<String> list);

Message class

**static** String find(Map<String, LinkedList<String>> list, String select)

{

상향된 for문으로 select와 일치하는 Map의 key 를 리턴 합니다. 없으면 “NULL” 리턴 합니다.

}

**public** **static** **void** main(String[] args) {

Map<key,Value>

Map의 key = 각 사람 이름 (String)

Value= 메시지 함( LinkedList<String>)

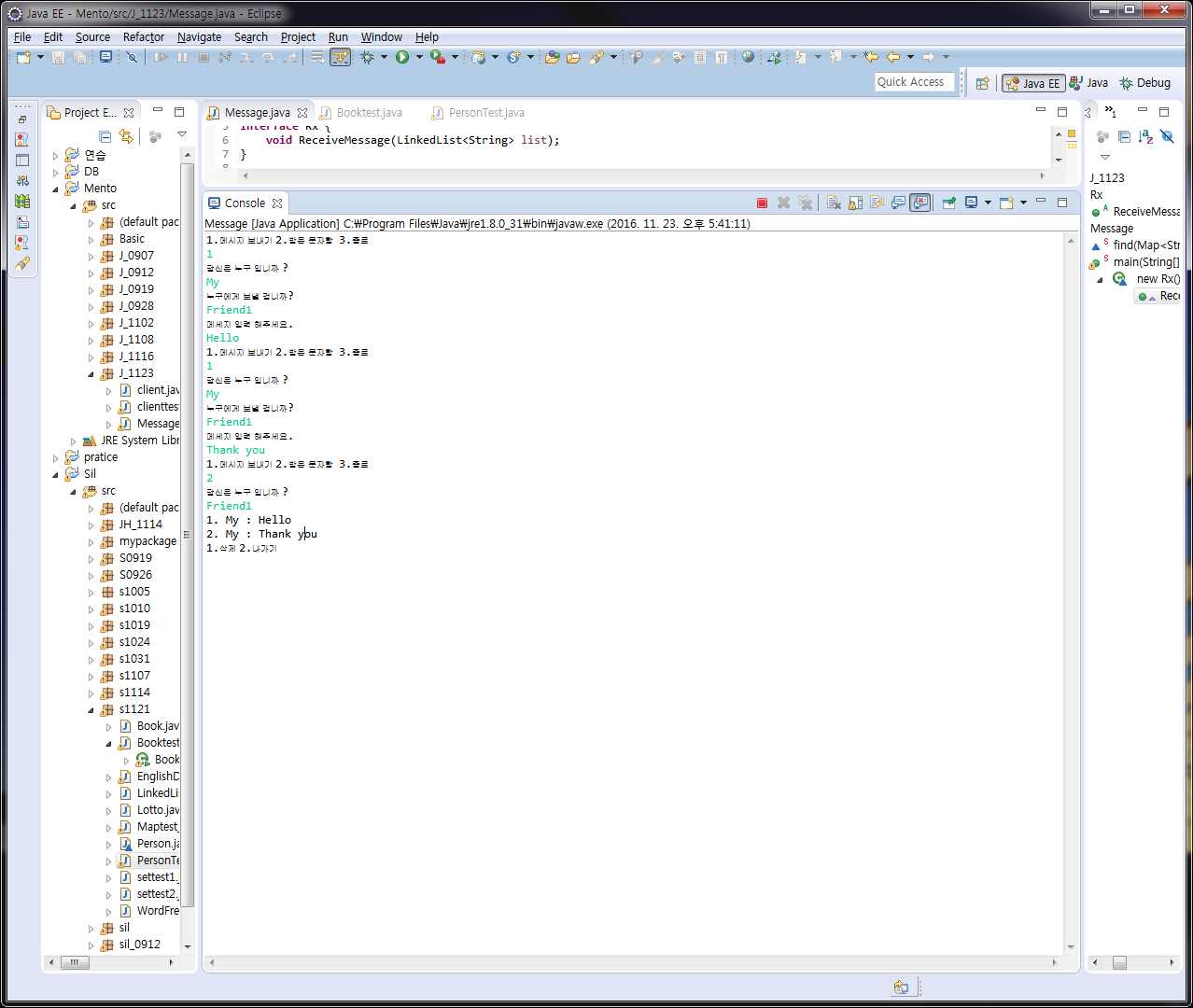
위와같이 Map을 선언해 줍니다. Map (HashMap)

Map 의 Key 에 My,Friend1,Friend2 를 넣어주고 각 Value 는 LinkedList를 생성하여 넣어줍니다.

Ex: m.put(“Hello”,new LinkedList< >());

인터페이스 Rx를 무명클래스로 선언 해줍니다.

클래스 안의 메소드 내용은



번호 | 내용 순으로 나오게 메시지함 반복문을 돌려

출력합니다.

Do {

System.***out***.println("1.메시지 보내기 2.받은 문자함 3.종료");

**int** select = scan.nextInt();

**if** (select == 1 || select == 2) {

1번, 2번 선택할 경우엔 먼저 누가 보내는건지, 누구의 문자함을 볼껀지 알아야 되기 때문에 공통적으로 [누가] 를 알아야 합니다.

System.***out***.println("당신은 누구 입니까 ?");

// 이름을 입력 받고 그 이름과 Map 전체를 find 메소드에 보내어 리턴값을 어떤 문자열을 선언하고 넣어줍니다.

그 문자열(사람이름)이 NULL 일 경우에는 존재하지 않는 사람 이므로 오류문구 출력 후 처음 단계로 돌아갑니다.

존재 한다면 메시지 보내는건지 그 사람의 받은 문자함을 볼 껀지 조건을걸어줍니다.

1. 메시지 보내기

System.***out***.println("누구에게 보낼 겁니까?");

위의 [누가] 와 똑같이 [누구에게] 를 알아야 합니다.

위와 똑 같이 누구에게 보낼 껀지 문자열 입력 받은 후 find 메소드를 통해

그 사람이 있는지 없는지 확인합니다. (NULL 은 위와 동일 )

존재 한다면

보낼 메시지를 입력합니다.

메시지는 띄어쓰기도 포함 해야 합니다. 하지만 nextLine을 사용 하면 뭔가 이상할 겁니다. 그래서 다음과 같이 사용하도록 합니다.

System.***out***.println("메세지 입력 해주세요.");

scan.nextLine();

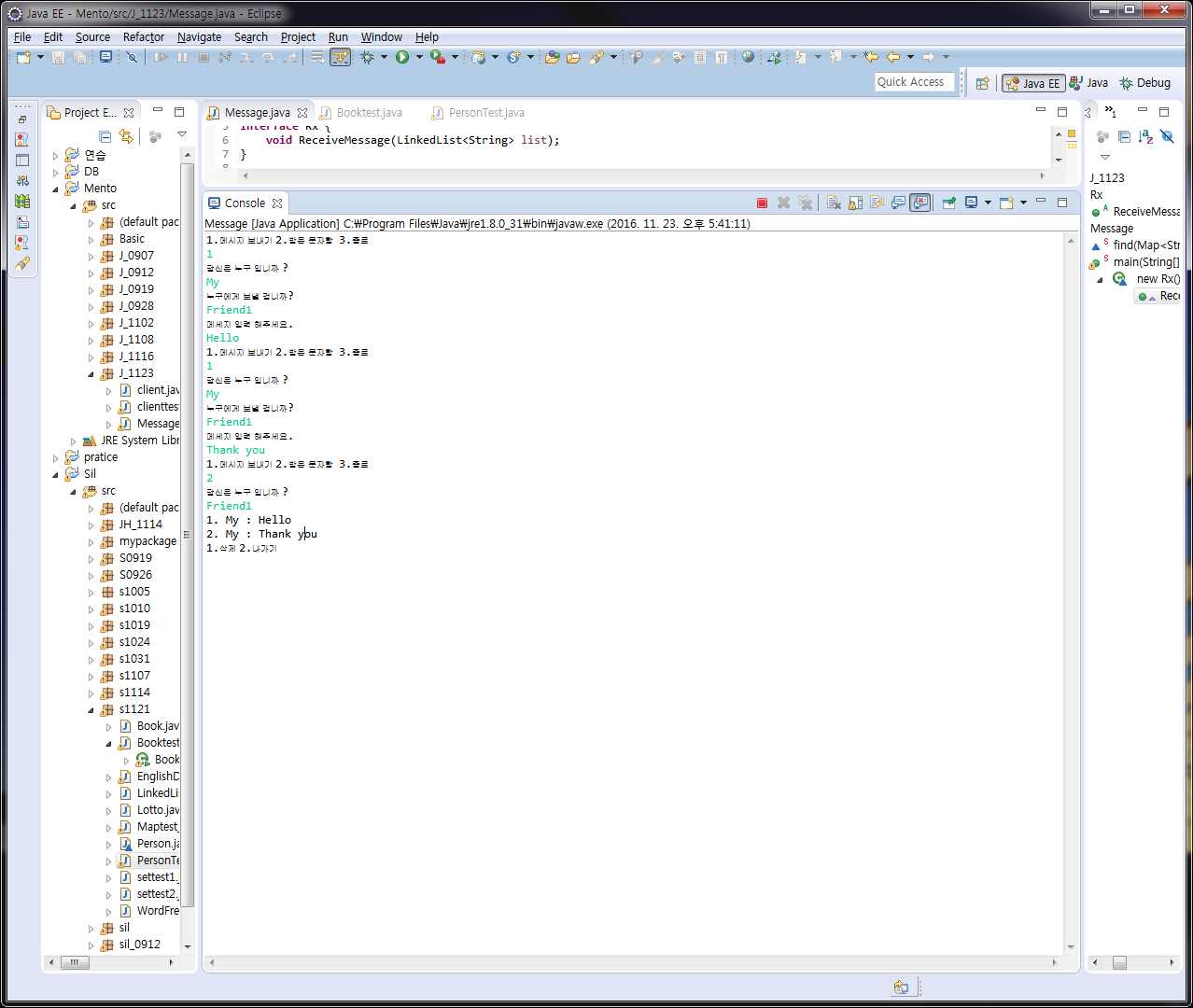
String send = scan.nextLine();

send 가 진짜 메시지 입력받는 문자열 입니다.

그 입력 받은 문자열을 수신자 의 메시지 함에 넣어야합니다.

우리는 보통 문자를 받으면 누가 어떤 내용을 보냈는지를 알 수있습니다. 하지만 이 코딩은 [누가]를 알 수 없고 메시지 내용만 볼 수있습니다. 그래서 [누가] 를 알 수 있게 문자열을 만들어 넣어줘야합니다.

문자열 변수 하나를 선언 후 그 문자열에 [누가] 뒤에 “ : “ 를 붙여서 넣어줍니다. 그리고 난 후 그 문자열 변수에 [누구에게] 를 붙여서 변경합니다. 그 다음 그 문자열 변수를 수신자의 메시지함(LinkedList) 에 넣어줍니다 (LinkedList 에 원소 넣을 때 어떤 메소드 쓰는건 알죠?)



원래는 1. Hello 가 나와야 하지만 [누가]를 붙여서

1. My : Hello 로 나오게 하시면 됩니당.
2. 메시지 함 보기

While(true){

반복문을 돌립니다. 그이유는 밑에 설명

일단 메시지함이 비어있는지 확인합니다.

비어있다면 오류문구 출력후 반복문에서 나갑니다.

System.***out***.println("받은 메세지가 없습니다. 이전메뉴로 이동합니다.");

**break**;

메시지가 있다면 아까 무명클래스 만든 변수에 있는 메소드에 메시지 함을 넣어 메시지를 전체 출력합니다.

System.***out***.println("1.삭제 2.나가기");

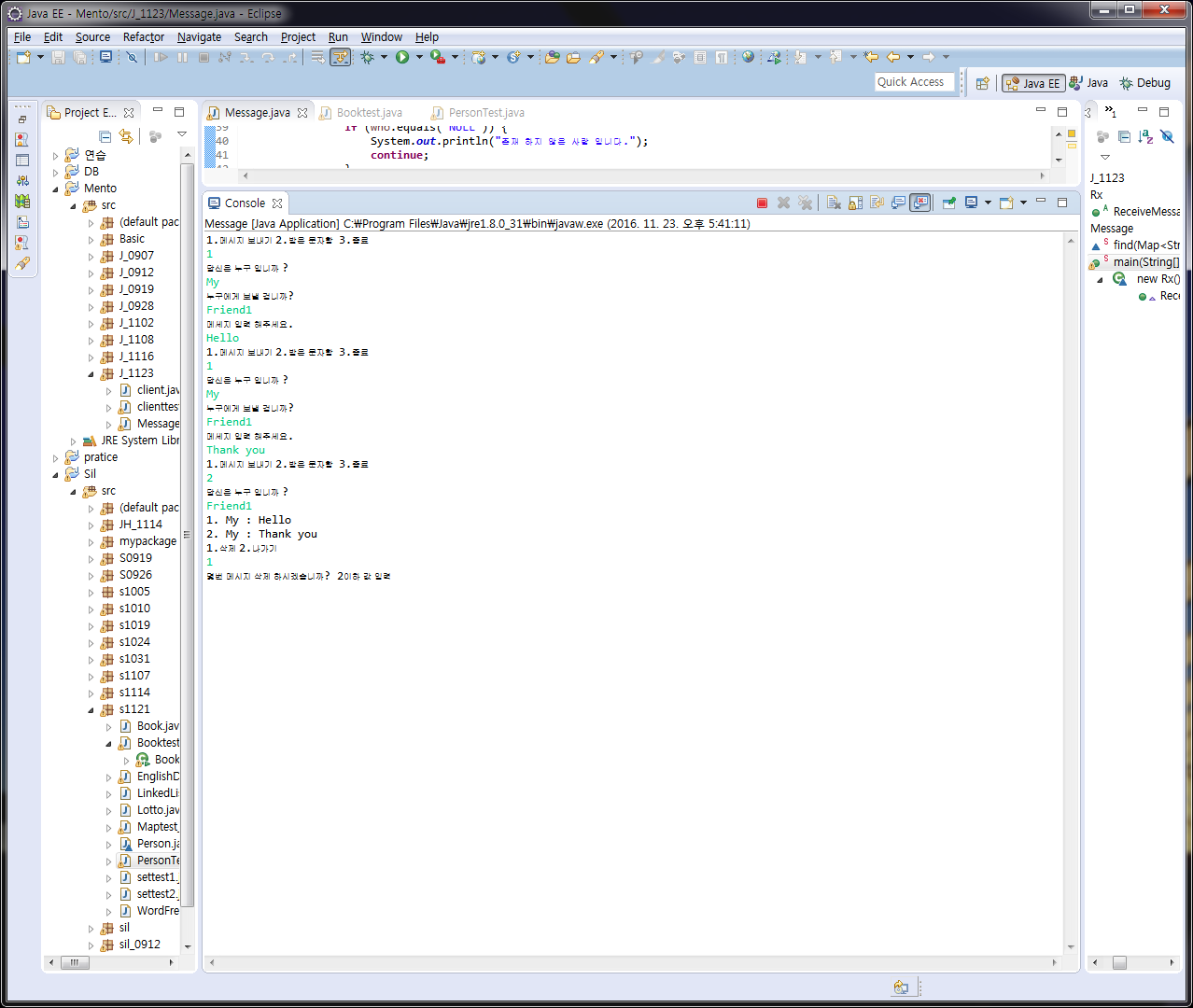
메시지 전체 출력 후 위와 같이 문구를 출력합니다.

삭제할 지 나갈지 입력 받고

2번은 나가고

1번일 경우엔

몇 번 메지지 삭제할 껀지 그리고 몇 번 메시지 까지 있다고 알려주는 출력문을 만듭니다.



삭제 할 번호 입력 받고

그 번호를 제거해 주시면 됩니다.

(예를 메시지 번호는 5번이라고 써있지만 원소의 방 번호는 4번째 인건 알고 있죠?)

삭제 하고 나선 다시 메시지 함을 출력해주는 반복문으로 돌아가면 됩니다.

