1번

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** java.applet.\*;

**import** java.sql.\*;

**public** **class** javaApplet **extends** Applet **implements** ActionListener {

Choice Join; // 선택 박스 선언

Choice View; // 선백 박스 선언

Choice SubQuery; // 선백 박스 선언

Label label; // 라벨 선언

TextArea Area; // 결과 값을 출력하는 텍스트에리어 선언

Button button; // 버튼 선언

**private** String url = "jdbc:inetdae7://210.115.229.77:2433"; //Microsoft SQL/Server용 Inet Software JDBC 드라이버

**private** String user = "20145165"; //ID

**private** String pass = "s145165@Hallym"; //password

**public** **void** init() { //객체를 초기화 하는 메소드

**try** {

Class.*forName*("com.inet.tds.TdsDriver");

} **catch** (ClassNotFoundException e) {

System.***out***.println("Class Loading Failed");

}

}

**public** **void** start() { //start 하는 메소드

setLayout(**new** FlowLayout()); // 레이아웃 배치자

Join = **new** Choice(); //선언한 선택 박스를 gui에서 사용하겠다고 선언한다.

Join.add("LEFT OUTER JOIN"); //선택 항목에 추가

Join.add("RIGHT OUTER JOIN");

add(Join);

button = **new** Button("실행"); //execute라는 버튼 생성

add(button); //gui에 추가

Area = **new** TextArea(10, 50); //TextArea를 선언 10, 50의 크기를 준다.

add(Area); //gui를 추가

button.addActionListener(**this**); //버튼 클릭시 동작

}

**public** **void** actionPerformed(ActionEvent ae) { //버튼을 클릭시 선택박스에 상태를 보고 그게 맞는 작업을 실행한다.

Connection con = **null**; //Connection 객체로 부터 쿼리를 수행하기 위한 PreparedStatement 객체를 받는다.

Statement stmt = **null**; //쿼리를 보내기 위해 사용되는 객체

String st = ae.getActionCommand(); // 검색버튼의 string 값을 전달 받음

String join = Join.getSelectedItem(); //join에서 선택한 값을 가져온다.

// 선택박스에서 선택된 값을 전달 받음

**if** (st.equals("실행")) { //버튼이 클릭되었는지 확인

**if** (join.equals("LEFT OUTER JOIN")) { //선택 값이 삽입이면

leftjoin();

}

**else**{ //선택값이 삭제라면

rightjoin();

}

}

}

**private** **void** leftjoin() {

Connection con = **null**;

Statement stmt = **null**;

**try** {

con = DriverManager.*getConnection*(url, user, pass);

con.setCatalog("20145165"); //Statement 객체를 생성

stmt = con.createStatement();

//커리문을 가져오고 세팅하는 작업

ResultSet result = stmt.executeQuery("select a.Course\_ID , b.Course\_ID\r\n" +

"from Course a LEFT OUTER JOIN section b\r\n" +

"on a.Course\_ID = b.Course\_ID");

// 여기부터 아래 세 번째 까지는 검색 버튼이 클릭될 때 마다 결과 창에 새로운 검색

// 결과를 보여주기 위하여 이전에 검색한 결과를 지우기 위한 작업이다.

String count = Area.getText(); //area에 있는 값을 가져온다.

**int** c = count.length(); //입력받은 문자열 길이를 받는다.

Area.~~replaceText~~(" ", 0, c);

**while** (result.next()) { //쿼리문에 값이 없을때 까지 돌린다.

String course = result.getString(1);

String section = result.getString(2);

//result 값을 변수로 받는다.

String value = "course : " + course + " section : " + section +"\n"; //문자열 저장

**int** index = Area.getText().length();

Area.~~insertText~~(value, index); // 결과창에 결과값 출력

}

con.close(); //Connection 닫기

stmt.close(); //Statement 닫기

} **catch** (Exception ee) {

System.***err***.println(ee); //에러 출력

}

}

**private** **void** rightjoin() {

Connection con = **null**;

Statement stmt = **null**;

**try** {

con = DriverManager.*getConnection*(url, user, pass);

con.setCatalog("20145165"); //Statement 객체를 생성

stmt = con.createStatement();

//커리문을 가져오고 세팅하는 작업

ResultSet result = stmt.executeQuery("select a.Course\_ID , b.Course\_ID\r\n" +

"from Course a RIGHT OUTER JOIN section b\r\n" +

"on a.Course\_ID = b.Course\_ID");

// 여기부터 아래 세 번째 까지는 검색 버튼이 클릭될 때 마다 결과 창에 새로운 검색

// 결과를 보여주기 위하여 이전에 검색한 결과를 지우기 위한 작업이다.

String count = Area.getText(); //area에 있는 값을 가져온다.

**int** c = count.length(); //입력받은 문자열 길이를 받는다.

Area.~~replaceText~~(" ", 0, c);

**while** (result.next()) { //쿼리문에 값이 없을때 까지 돌린다.

String course = result.getString(1);

String section = result.getString(2);

//result 값을 변수로 받는다.

String value = "course : " + course + " section : " + section +"\n"; //문자열 저장

**int** index = Area.getText().length();

Area.~~insertText~~(value, index); // 결과창에 결과값 출력

}

con.close(); //Connection 닫기

stmt.close(); //Statement 닫기

} **catch** (Exception ee) {

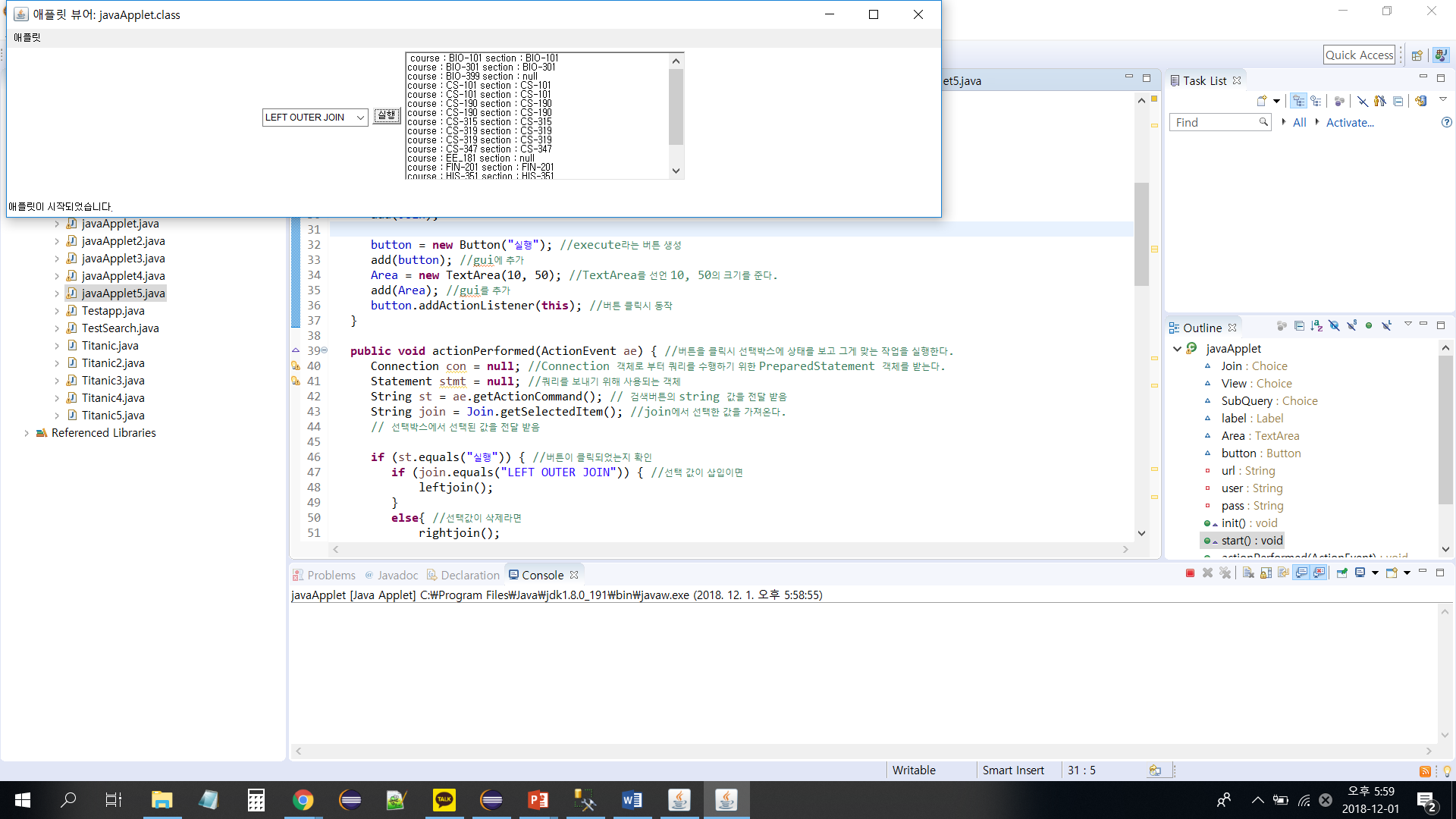
System.***err***.println(ee); //에러 출력

}

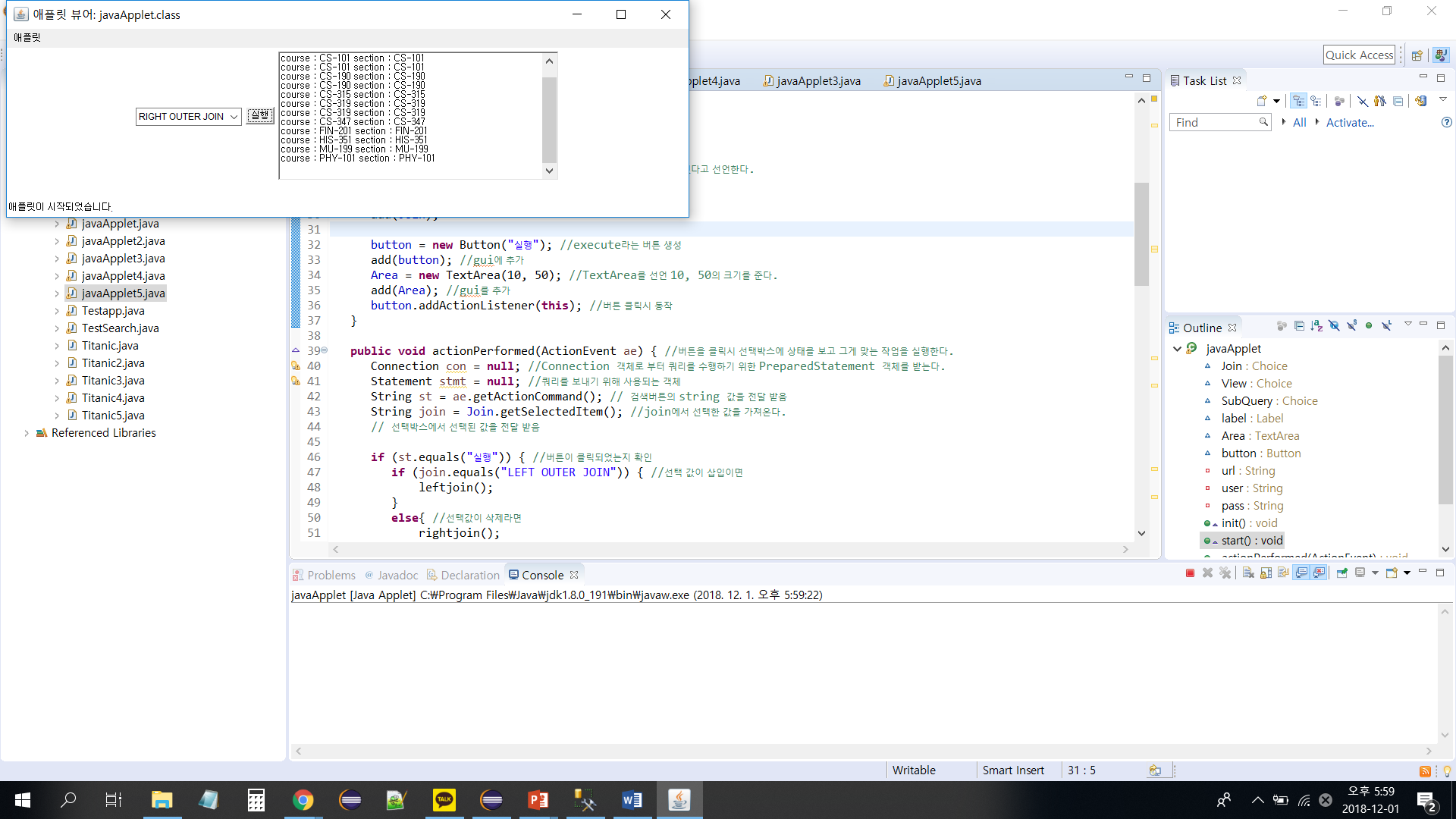
}

}

left



right



2번

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** java.applet.\*;

**import** java.sql.\*;

**public** **class** javaApplet2 **extends** Applet **implements** ActionListener {

Label label; // 라벨 선언

TextArea Area; // 결과 값을 출력하는 텍스트에리어 선언

Button button; // 버튼 선언

**private** String url = "jdbc:inetdae7://210.115.229.77:2433"; //Microsoft SQL/Server용 Inet Software JDBC 드라이버

**private** String user = "20145165"; //ID

**private** String pass = "s145165@Hallym"; //password

**public** **void** init() { //객체를 초기화 하는 메소드

**try** {

Class.*forName*("com.inet.tds.TdsDriver");

} **catch** (ClassNotFoundException e) {

System.***out***.println("Class Loading Failed");

}

}

**public** **void** start() { //start 하는 메소드

setLayout(**new** FlowLayout()); // 레이아웃 배치자

label = **new** Label("모든 교수의 목록을 교수의 ID, 이름, 지금까지 가르친 강의 수를 포함 하시오");

button = **new** Button("실행"); //execute라는 버튼 생성

add(button); //gui에 추가

Area = **new** TextArea(10, 50); //TextArea를 선언 10, 50의 크기를 준다.

add(Area); //gui를 추가

button.addActionListener(**this**); //버튼 클릭시 동작

}

**public** **void** actionPerformed(ActionEvent ae) { //버튼을 클릭시 선택박스에 상태를 보고 그게 맞는 작업을 실행한다.

Connection con = **null**; //Connection 객체로 부터 쿼리를 수행하기 위한 PreparedStatement 객체를 받는다.

Statement stmt = **null**; //쿼리를 보내기 위해 사용되는 객체

String st = ae.getActionCommand(); // 검색버튼의 string 값을 전달 받음

// 선택박스에서 선택된 값을 전달 받음

**if** (st.equals("실행")) { //버튼이 클릭되었는지 확인

join();

}

}

**private** **void** join() {

Connection con = **null**;

Statement stmt = **null**;

**try** {

con = DriverManager.*getConnection*(url, user, pass);

con.setCatalog("20145165"); //Statement 객체를 생성

stmt = con.createStatement();

//커리문을 가져오고 세팅하는 작업

ResultSet result = stmt.executeQuery("select a.ID, a.name, COUNT(b.id) as Lecture\r\n" +

"from [20145165].dbo.instructor a LEFT OUTER JOIN [20145165].dbo.Teaches b\r\n" +

"on a.ID = b.id\r\n" +

"group by a.ID,a.name");

// 여기부터 아래 세 번째 까지는 검색 버튼이 클릭될 때 마다 결과 창에 새로운 검색

// 결과를 보여주기 위하여 이전에 검색한 결과를 지우기 위한 작업이다.

String count = Area.getText(); //area에 있는 값을 가져온다.

**int** c = count.length(); //입력받은 문자열 길이를 받는다.

Area.~~replaceText~~(" ", 0, c);

**while** (result.next()) { //쿼리문에 값이 없을때 까지 돌린다.

String id = result.getString(1);

String name = result.getString(2);

String Lecture = Integer.*toString*(result.getInt(3));

//result 값을 변수로 받는다.

String value = "ID : " + id + " Name : " + name +" \nLecture count: "+Lecture+"\n"; //문자열 저장

**int** index = Area.getText().length();

Area.~~insertText~~(value, index); // 결과창에 결과값 출력

}

con.close(); //Connection 닫기

stmt.close(); //Statement 닫기

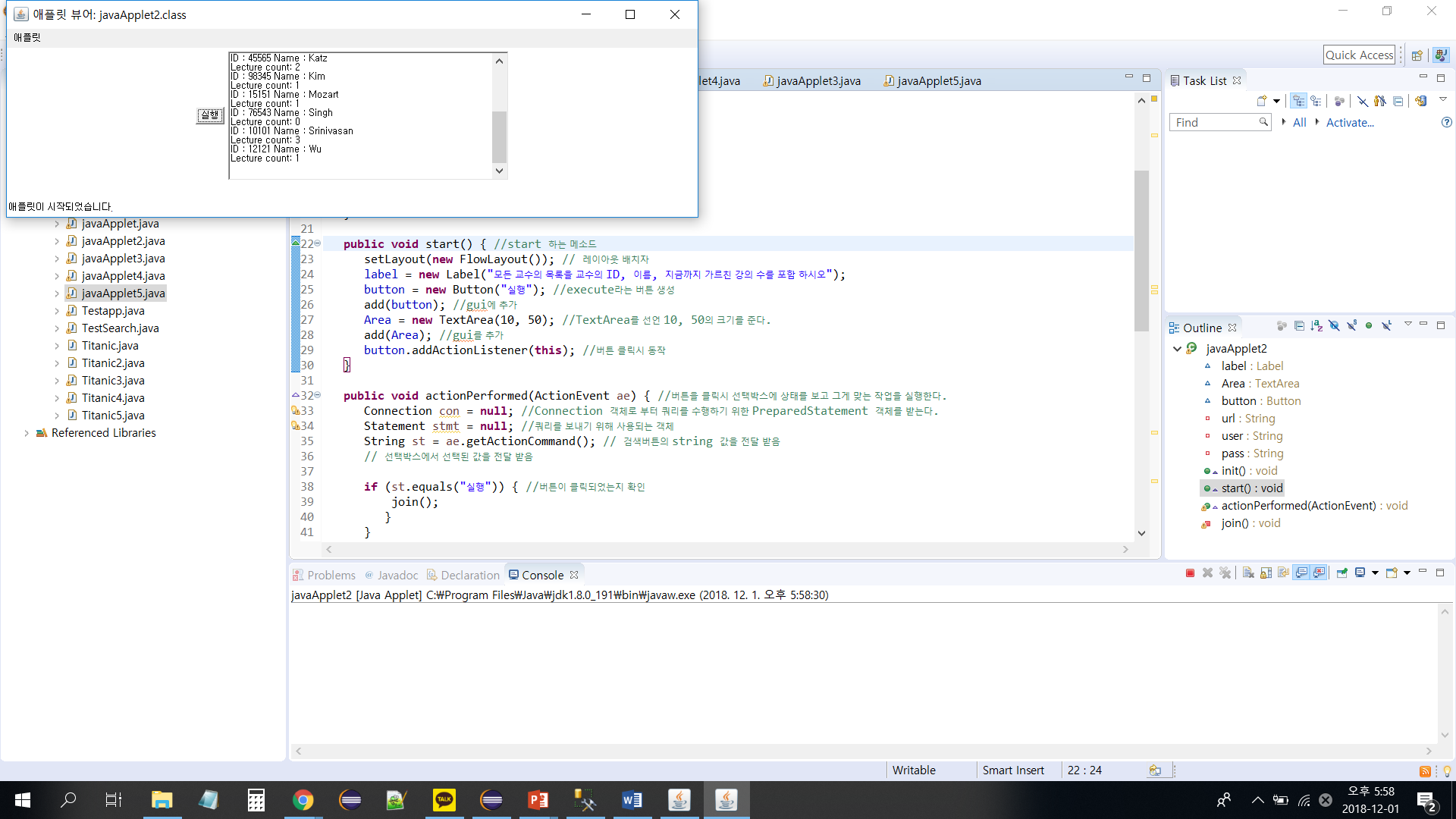
} **catch** (Exception ee) {

System.***err***.println(ee); //에러 출력

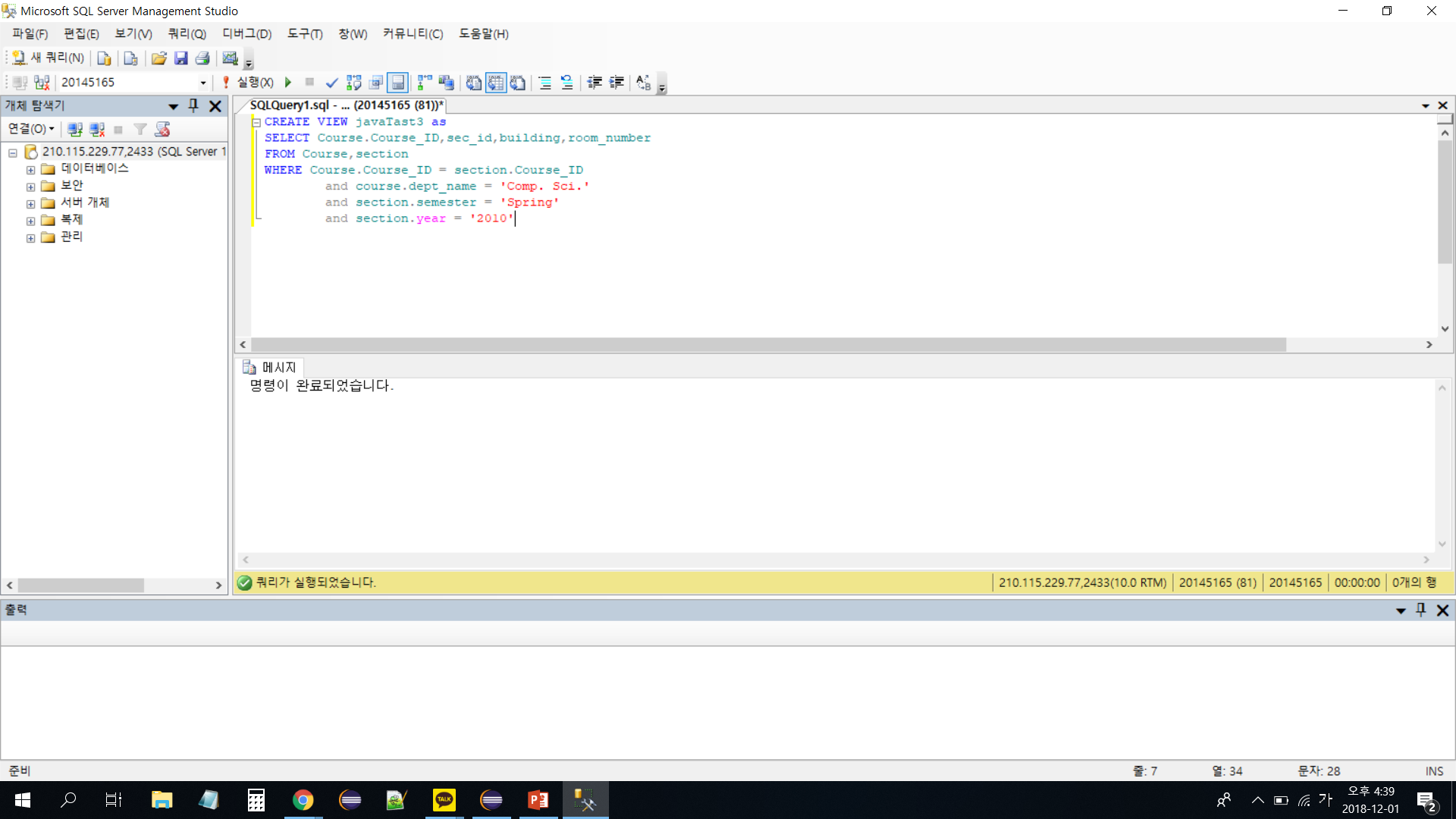
}

}

}



3.1



3번

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** java.applet.\*;

**import** java.sql.\*;

**public** **class** javaApplet3 **extends** Applet **implements** ActionListener {

Label label; // 라벨 선언

TextArea Area; // 결과 값을 출력하는 텍스트에리어 선언

Button button; // 버튼 선언

**private** String url = "jdbc:inetdae7://210.115.229.77:2433"; //Microsoft SQL/Server용 Inet Software JDBC 드라이버

**private** String user = "20145165"; //ID

**private** String pass = "s145165@Hallym"; //password

**public** **void** init() { //객체를 초기화 하는 메소드

**try** {

Class.*forName*("com.inet.tds.TdsDriver");

} **catch** (ClassNotFoundException e) {

System.***out***.println("Class Loading Failed");

}

}

**public** **void** start() { //start 하는 메소드

setLayout(**new** FlowLayout()); // 레이아웃 배치자

label = **new** Label("컴퓨터 공학과");

button = **new** Button("실행"); //execute라는 버튼 생성

add(button); //gui에 추가

Area = **new** TextArea(10, 50); //TextArea를 선언 10, 50의 크기를 준다.

add(Area); //gui를 추가

button.addActionListener(**this**); //버튼 클릭시 동작

}

**public** **void** actionPerformed(ActionEvent ae) { //버튼을 클릭시 선택박스에 상태를 보고 그게 맞는 작업을 실행한다.

Connection con = **null**; //Connection 객체로 부터 쿼리를 수행하기 위한 PreparedStatement 객체를 받는다.

Statement stmt = **null**; //쿼리를 보내기 위해 사용되는 객체

String st = ae.getActionCommand(); // 검색버튼의 string 값을 전달 받음

// 선택박스에서 선택된 값을 전달 받음

**if** (st.equals("실행")) { //버튼이 클릭되었는지 확인

view();

}

}

**private** **void** view() {

Connection con = **null**;

Statement stmt = **null**;

**try** {

con = DriverManager.*getConnection*(url, user, pass);

con.setCatalog("20145165"); //Statement 객체를 생성

stmt = con.createStatement();

//커리문을 가져오고 세팅하는 작업

ResultSet result = stmt.executeQuery("select \* "

+ "from javaTast3");

// 여기부터 아래 세 번째 까지는 검색 버튼이 클릭될 때 마다 결과 창에 새로운 검색

// 결과를 보여주기 위하여 이전에 검색한 결과를 지우기 위한 작업이다.

String count = Area.getText(); //area에 있는 값을 가져온다.

**int** c = count.length(); //입력받은 문자열 길이를 받는다.

Area.~~replaceText~~(" ", 0, c);

**while** (result.next()) { //쿼리문에 값이 없을때 까지 돌린다.

String id = result.getString(1);

String sec = Integer.*toString*(result.getInt(2));

String building = result.getString(3);

String room = Integer.*toString*(result.getInt(4));

//result 값을 변수로 받는다.

String value = "ID : " + id + " sec : " + sec +" building : "+building+" room : "+room+"\n"; //문자열 저장

**int** index = Area.getText().length();

Area.~~insertText~~(value, index); // 결과창에 결과값 출력

}

con.close(); //Connection 닫기

stmt.close(); //Statement 닫기

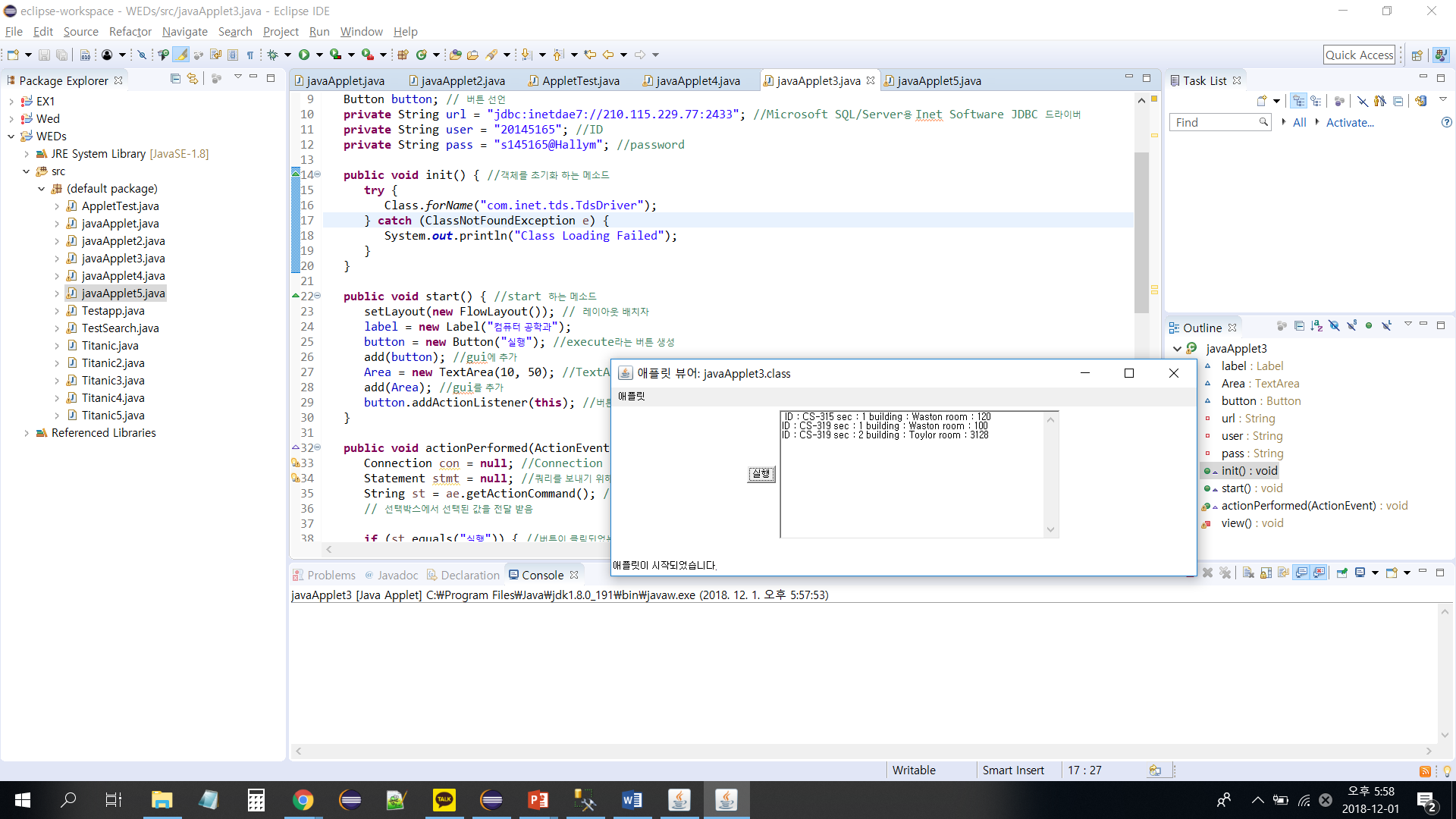
} **catch** (Exception ee) {

System.***err***.println(ee); //에러 출력

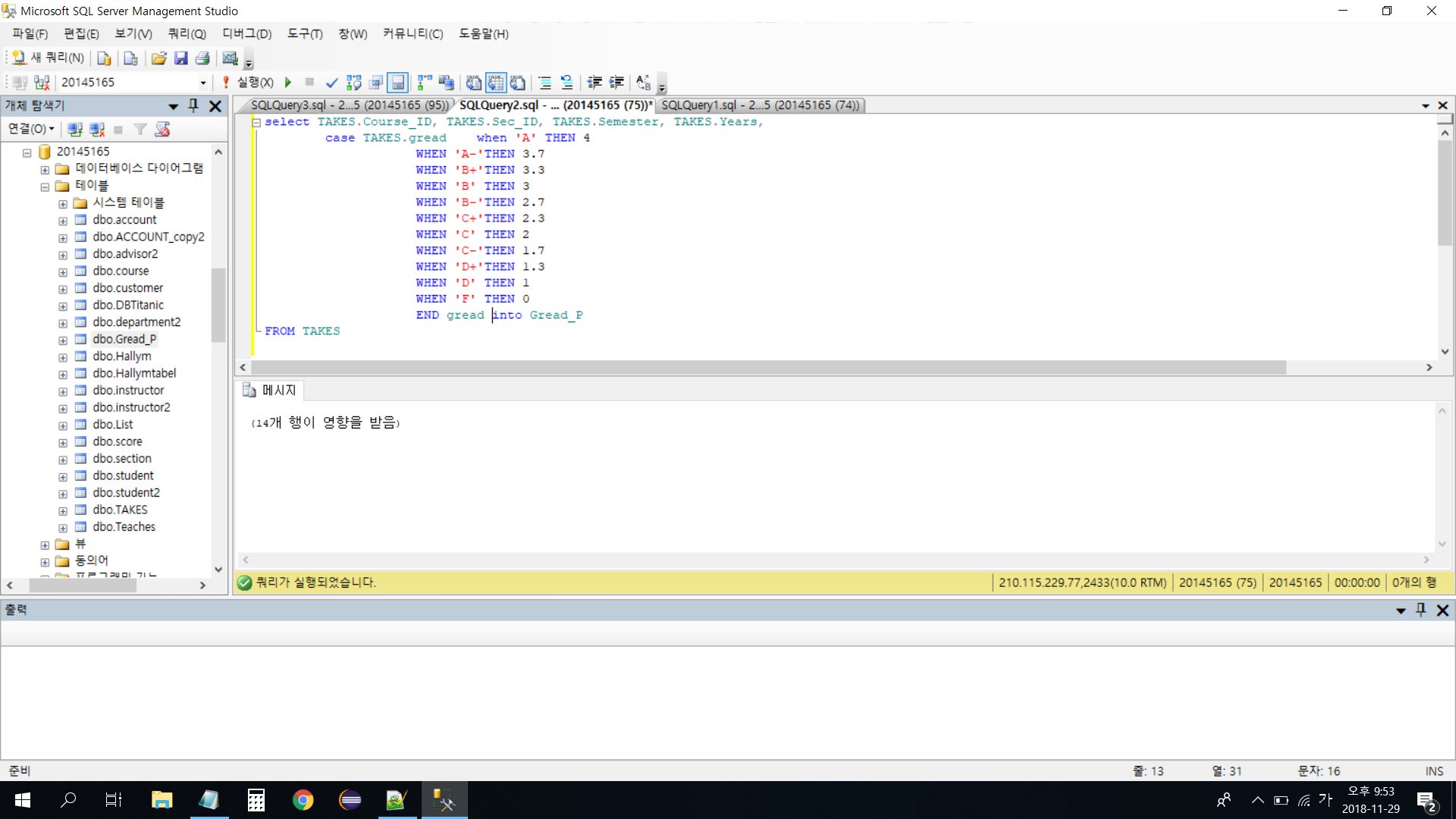
}

}

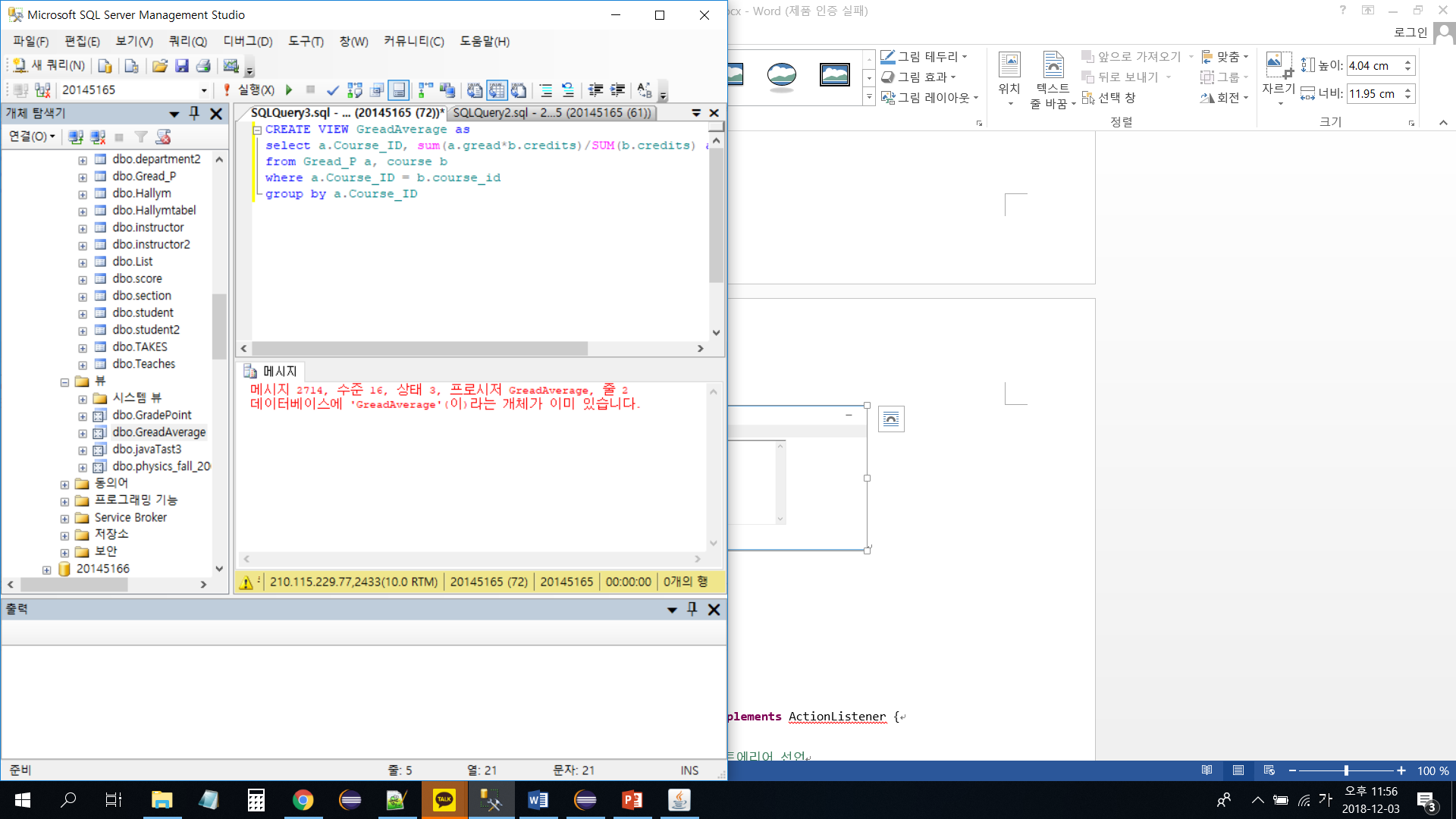
}



4.1



4.2



4번

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.sql.\*;

public class javaApplet4 extends Applet implements ActionListener {

Label label; // 라벨 선언

TextArea Area; // 결과 값을 출력하는 텍스트에리어 선언

Button button; // 버튼 선언

private String url = "jdbc:inetdae7://210.115.229.77:2433"; //Microsoft SQL/Server용 Inet Software JDBC 드라이버

private String user = "20145165"; //ID

private String pass = "s145165@Hallym"; //password

public void init() { //객체를 초기화 하는 메소드

try {

Class.forName("com.inet.tds.TdsDriver");

} catch (ClassNotFoundException e) {

System.out.println("Class Loading Failed");

}

}

public void start() { //start 하는 메소드

setLayout(new FlowLayout()); // 레이아웃 배치자

label = new Label("생성하고 각 학생의 평균 학점");

button = new Button("실행"); //execute라는 버튼 생성

add(button); //gui에 추가

Area = new TextArea(10, 50); //TextArea를 선언 10, 50의 크기를 준다.

add(Area); //gui를 추가

button.addActionListener(this); //버튼 클릭시 동작

}

public void actionPerformed(ActionEvent ae) { //버튼을 클릭시 선택박스에 상태를 보고 그게 맞는 작업을 실행한다.

Connection con = null; //Connection 객체로 부터 쿼리를 수행하기 위한 PreparedStatement 객체를 받는다.

Statement stmt = null; //쿼리를 보내기 위해 사용되는 객체

String st = ae.getActionCommand(); // 검색버튼의 string 값을 전달 받음

// 선택박스에서 선택된 값을 전달 받음

if (st.equals("실행")) { //버튼이 클릭되었는지 확인

view();

}

}

private void view() {

Connection con = null;

Statement stmt = null;

try {

con = DriverManager.getConnection(url, user, pass);

con.setCatalog("20145165"); //Statement 객체를 생성

stmt = con.createStatement();

//커리문을 가져오고 세팅하는 작업

ResultSet result = stmt.executeQuery("select \*\r\n" +

"from GreadAverage");

// 여기부터 아래 세 번째 까지는 검색 버튼이 클릭될 때 마다 결과 창에 새로운 검색

// 결과를 보여주기 위하여 이전에 검색한 결과를 지우기 위한 작업이다.

String count = Area.getText(); //area에 있는 값을 가져온다.

int c = count.length(); //입력받은 문자열 길이를 받는다.

Area.replaceText(" ", 0, c);

while (result.next()) { //쿼리문에 값이 없을때 까지 돌린다.

String id = result.getString(1);

String aver = result.getString(2);

Double wd = new Double(aver);

//result 값을 변수로 받는다.

String value = "ID : " + id + " Average : " + wd +"\n"; //문자열 저장

int index = Area.getText().length();

Area.insertText(value, index); // 결과창에 결과값 출력

}

con.close(); //Connection 닫기

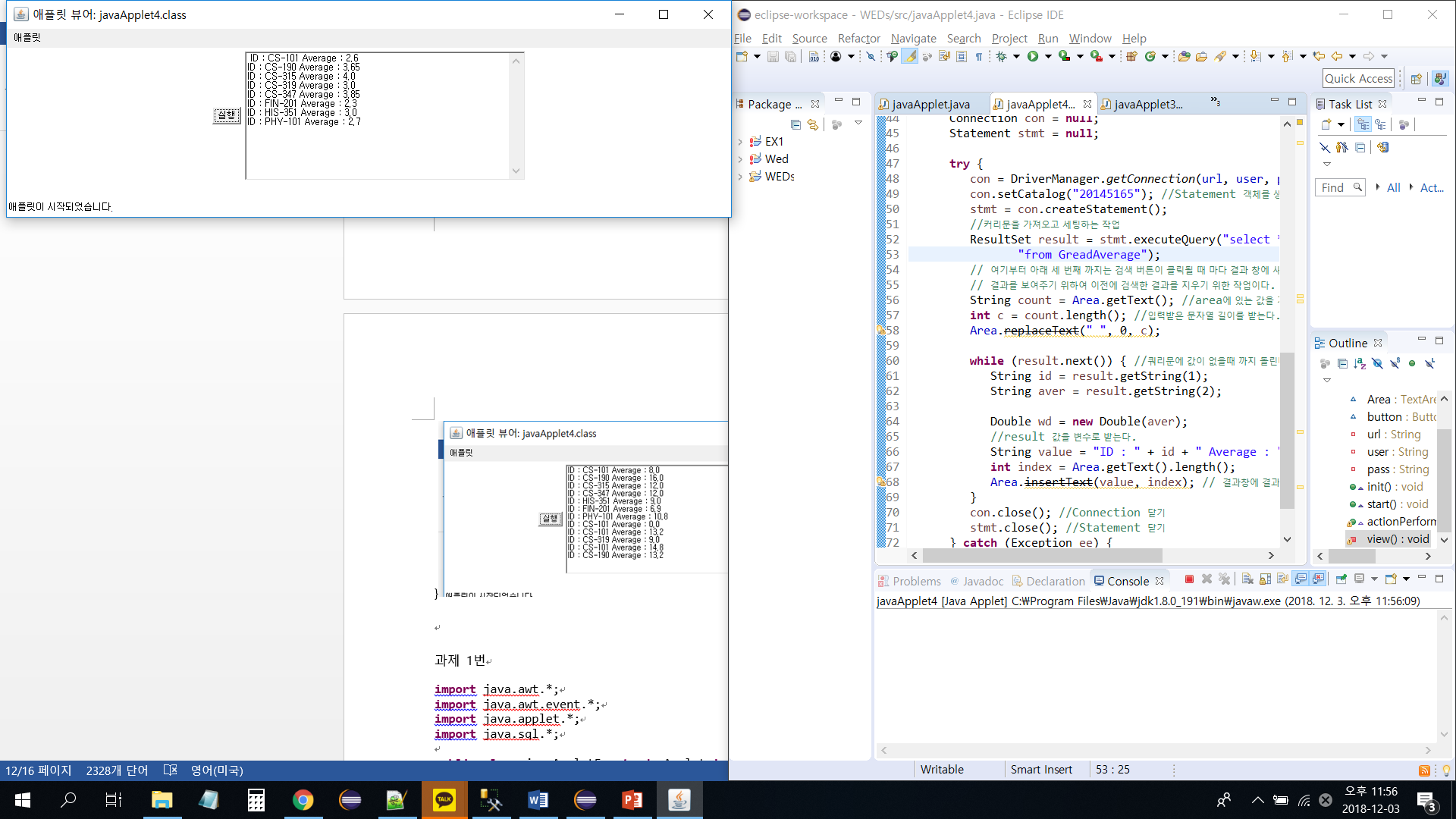
stmt.close(); //Statement 닫기

} catch (Exception ee) {

System.err.println(ee); //에러 출력

}

}

}

과제 1번

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** java.applet.\*;

**import** java.sql.\*;

**public** **class** javaApplet5 **extends** Applet **implements** ActionListener {

Label label; // 라벨 선언

TextArea Area; // 결과 값을 출력하는 텍스트에리어 선언

Button button1; // 버튼 선언

Button button2; // 버튼 선언

Button button3; // 버튼 선언

**private** String url = "jdbc:inetdae7://210.115.229.77:2433"; //Microsoft SQL/Server용 Inet Software JDBC 드라이버

**private** String user = "20145165"; //ID

**private** String pass = "s145165@Hallym"; //password

**public** **void** init() { //객체를 초기화 하는 메소드

**try** {

Class.*forName*("com.inet.tds.TdsDriver");

} **catch** (ClassNotFoundException e) {

System.***out***.println("Class Loading Failed");

}

}

**public** **void** start() { //start 하는 메소드

setLayout(**new** FlowLayout()); // 레이아웃 배치자

label = **new** Label("과제");

button1 = **new** Button("1번"); // 버튼 생성

add(button1); //gui에 추가

button2 = **new** Button("2번"); // 버튼 생성

add(button2); //gui에 추가

button3 = **new** Button("3번"); // 버튼 생성

add(button3); //gui에 추가

Area = **new** TextArea(10, 50); //TextArea를 선언 10, 50의 크기를 준다.

add(Area); //gui를 추가

button1.addActionListener(**this**); //버튼 클릭시 동작

button2.addActionListener(**this**);

button3.addActionListener(**this**);

}

**public** **void** actionPerformed(ActionEvent ae) { //버튼을 클릭시 선택박스에 상태를 보고 그게 맞는 작업을 실행한다.

Connection con = **null**; //Connection 객체로 부터 쿼리를 수행하기 위한 PreparedStatement 객체를 받는다.

Statement stmt = **null**; //쿼리를 보내기 위해 사용되는 객체

String st = ae.getActionCommand(); // 검색버튼의 string 값을 전달 받음

// 선택박스에서 선택된 값을 전달 받음

**if** (st.equals("1번")) { //버튼이 클릭되었는지 확인

title();

}**else** **if**(st.equals("2번")) {

two();

}**else** **if**(st.equals("3번")){

three();

}

}

**private** **void** title() {

Connection con = **null**;

Statement stmt = **null**;

**try** {

con = DriverManager.*getConnection*(url, user, pass);

con.setCatalog("20145165"); //Statement 객체를 생성

stmt = con.createStatement();

//커리문을 가져오고 세팅하는 작업

ResultSet result = stmt.executeQuery("select Title\r\n" +

"from Course\r\n" +

"where dept\_name = 'Comp\_Sci' and Credits = 3");

// 여기부터 아래 세 번째 까지는 검색 버튼이 클릭될 때 마다 결과 창에 새로운 검색

// 결과를 보여주기 위하여 이전에 검색한 결과를 지우기 위한 작업이다.

String count = Area.getText(); //area에 있는 값을 가져온다.

**int** c = count.length(); //입력받은 문자열 길이를 받는다.

Area.~~replaceText~~(" ", 0, c);

**while** (result.next()) { //쿼리문에 값이 없을때 까지 돌린다.

String Ti = result.getString(1);

//result 값을 변수로 받는다.

String value = "Title : " + Ti +"\n"; //문자열 저장

**int** index = Area.getText().length();

Area.~~insertText~~(value, index); // 결과창에 결과값 출력

}

con.close(); //Connection 닫기

stmt.close(); //Statement 닫기

} **catch** (Exception ee) {

System.***err***.println(ee); //에러 출력

}

}

**private** **void** two() {

Connection con = **null**;

Statement stmt = **null**;

**try** {

con = DriverManager.*getConnection*(url, user, pass);

con.setCatalog("20145165"); //Statement 객체를 생성

stmt = con.createStatement();

//커리문을 가져오고 세팅하는 작업

ResultSet result = stmt.executeQuery("select TAKES.ID\r\n" +

"from instructor,teaches,TAKES\r\n" +

"where instructor.name = 'Einstein' and TAKES.Course\_id = teaches.Course\_id and teaches.id in(select id from instructor where instructor.name = 'Einstein') ");

// 여기부터 아래 세 번째 까지는 검색 버튼이 클릭될 때 마다 결과 창에 새로운 검색

// 결과를 보여주기 위하여 이전에 검색한 결과를 지우기 위한 작업이다.

String count = Area.getText(); //area에 있는 값을 가져온다.

**int** c = count.length(); //입력받은 문자열 길이를 받는다.

Area.~~replaceText~~(" ", 0, c);

**while** (result.next()) { //쿼리문에 값이 없을때 까지 돌린다.

String id = result.getString(1);

//result 값을 변수로 받는다.

String value = "ID : " + id +"\n"; //문자열 저장

**int** index = Area.getText().length();

Area.~~insertText~~(value, index); // 결과창에 결과값 출력

}

con.close(); //Connection 닫기

stmt.close(); //Statement 닫기

} **catch** (Exception ee) {

System.***err***.println(ee); //에러 출력

}

}

**private** **void** three() {

Connection con = **null**;

Statement stmt = **null**;

**try** {

con = DriverManager.*getConnection*(url, user, pass);

con.setCatalog("20145165"); //Statement 객체를 생성

stmt = con.createStatement();

//커리문을 가져오고 세팅하는 작업

ResultSet result = stmt.executeQuery("select \*\r\n" +

"from instructor\r\n" +

"where salary = (select max(salary) from instructor)\r\n" );

// 여기부터 아래 세 번째 까지는 검색 버튼이 클릭될 때 마다 결과 창에 새로운 검색

// 결과를 보여주기 위하여 이전에 검색한 결과를 지우기 위한 작업이다.

String count = Area.getText(); //area에 있는 값을 가져온다.

**int** c = count.length(); //입력받은 문자열 길이를 받는다.

Area.~~replaceText~~(" ", 0, c);

**while** (result.next()) { //쿼리문에 값이 없을때 까지 돌린다.

String id = result.getString(1);

String name = result.getString(2);

String dept = result.getString(3);

String salary = Integer.*toString*(result.getInt(4));

//result 값을 변수로 받는다.

String value = "ID : " + id +" name: "+name+" dept: "+dept+" salary: "+salary+"\n"; //문자열 저장

**int** index = Area.getText().length();

Area.~~insertText~~(value, index); // 결과창에 결과값 출력

}

con.close(); //Connection 닫기

stmt.close(); //Statement 닫기

} **catch** (Exception ee) {

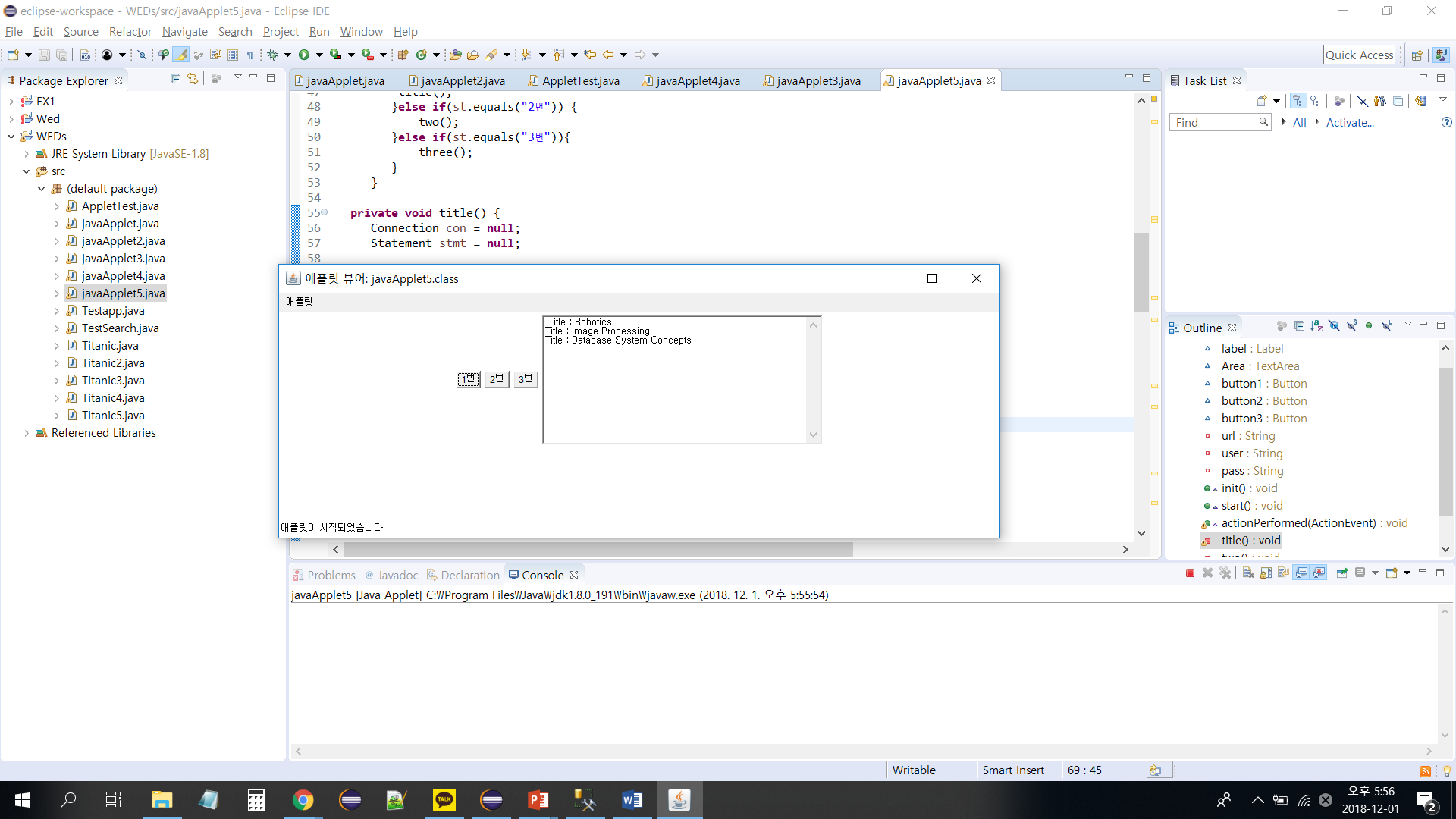
System.***err***.println(ee); //에러 출력

}

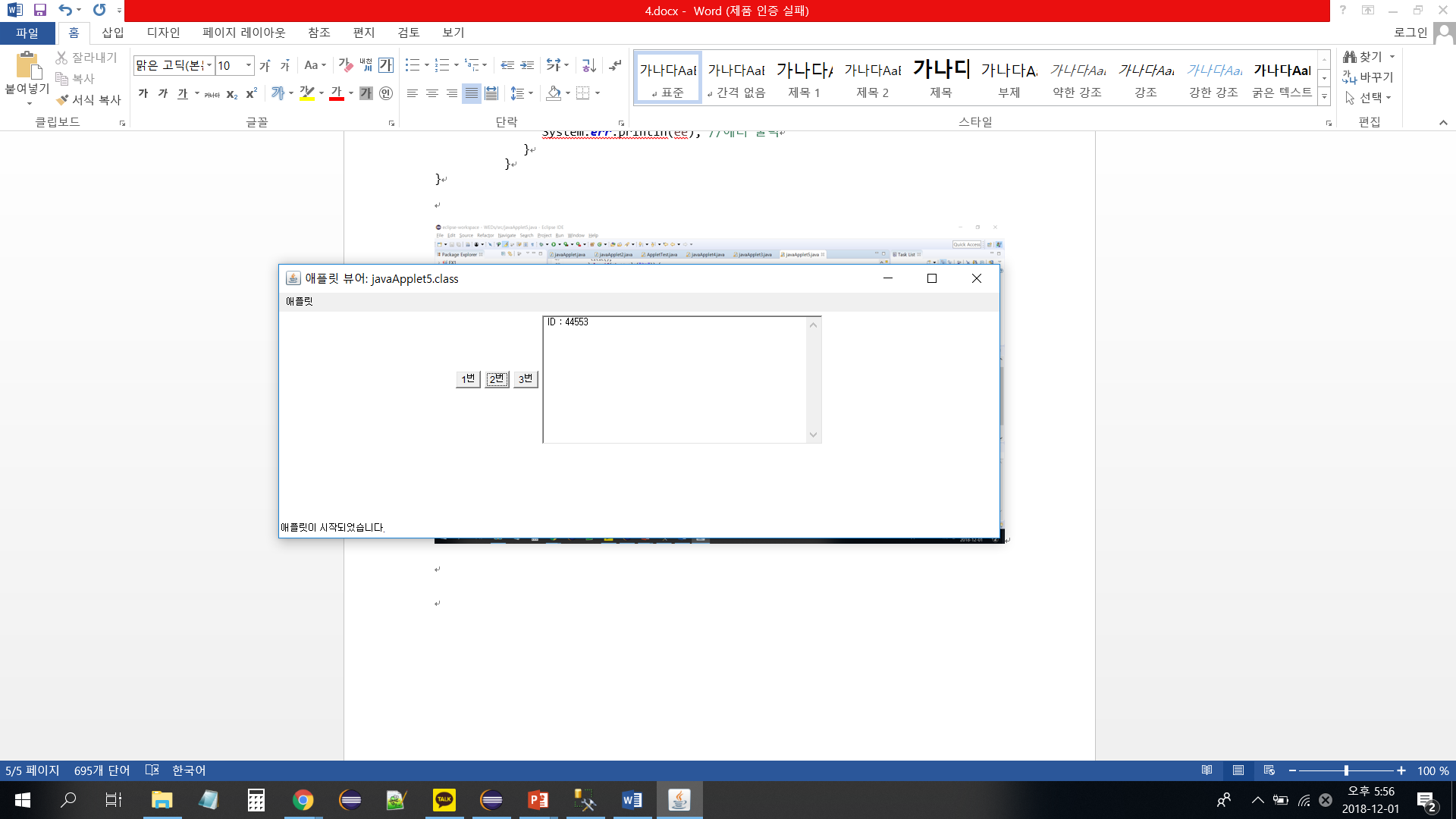
}

}

1



2



3

