저번 주 과제

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.sql.\*;

public class AppletTest extends Applet implements ActionListener {

Choice search; // 선택 박스 선언

Choice status; // 선백 박스 선언

Label label; // 라벨 선언

TextField textField;// 값을받는 텍스트 필드 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("상태"); //선언한 라벨에 "상태"라는 값을 준다.

add(label); //add를 사용에 gui에 추가한다.

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

status.add("삽입"); //선택 항목에 추가

status.add("삭제");

status.add("갱신");

status.add("검색");

add(status); //gui에 추가

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

search.add("전체"); //선택 항목에 추가

search.add("이름");

search.add("나이");

search.add("성별");

add(search); //gui에 추가

textField = new TextField(10); //10칸짜리 TextField를 선언

add(textField); //gui에 추가

button = new Button("execute"); //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 sear = search.getSelectedItem(); //search에서 선택한 값을 가져온다.

String stat = status.getSelectedItem(); //status에서 선택한 값을 가져온다.

// 선택박스에서 선택된 값을 전달 받음

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

String n = textField.getText(); //textField에 값을 가져온다.

String[] split = n.split(", "); //문자열을 특정 구분자 를 기준으로 나눠서 배열에 담아 이용을 합니다

if (stat.equals("삽입")) { //선택 값이 삽입이면

Insert(split[0], Integer.parseInt(split[1]), split[2]); //sql에서 쓰이는 명령어

}

else if (stat.equals("삭제")) { //선택값이 삭제라면

customer\_Delete(n);

}

else if (stat.equals("갱신")) { //선택값이 갱신이라면

customer\_Update();

}

else if (stat.equals("검색")) { //검색일시 다른 조건을 봐야된다

if (sear.equals("전체")) { //다른 선택박스에 검색시 조건을 확인 -> "전체"라면

TotalgetDBSearch(); //모든 데이터를 가져온다

}

else if (sear.equals("이름")) { //이름이 같은 사람을 찾는다.

NamegetDBSearch(n);

}

else if (sear.equals("나이")) { //나이가 같은 데이터를 찾는다.

AgegetDBSearch(n);

}

else if (sear.equals("성별")) { //성별이 같은 사람을 찾는다.

SexgetDBSearch(n);

}

}

}

}

private void TotalgetDBSearch() {

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 customer");

// 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); //0부터 c 문자열 끝까지 문자열을 저장

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

String Name = result.getString(1);

String age = Integer.toString(result.getInt(2));

String sex = result.getString(3);

//result 값을 변수로 받는다.

String value = "이름 : " + Name + " 나이 : " + age + " 성별 : " + sex + "\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 NamegetDBSearch(String n) {

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 customer WHERE name ='" + n + "'");

//커리문에서 name이 같은 것들을 받아온다.

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

int c = count.length(); //길이를 가져온다.

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

while (result.next()) { //검색했던 값이 끝날때 까지 반복

String Name = result.getString(1);

String age = Integer.toString(result.getInt(2));

String sex = result.getString(3);

//받아온 문자열을 조건에 맞게 배치

String value = "이름 : " + Name + " 나이 : " + age + " 성별 : " + sex + "\n"; //문자열 저장

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

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

}

con.close(); //닫기

stmt.close(); //다기

} catch (SQLException se) {

System.out.println(se.getMessage()); //오류 출력

}

}

private void AgegetDBSearch(String n) {

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 customer WHERE age ='" + n + "'"); //나이로 검색

String count = Area.getText(); //area에 text를 받아온다.

int c = count.length(); //길이 확인

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

while (result.next()) { //가져온 문자열 끝까지 반복

String Name = result.getString(1);

String age = Integer.toString(result.getInt(2));

String sex = result.getString(3);

//받아온 문자열을 조건에 맞게 배치

String value = "이름 : " + Name + " 나이 : " + age + " 성별 : " + sex + "\n"; //받아온 내용을 문자열로 저장

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

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

}

con.close(); //닫기

stmt.close(); //닫기

} catch (SQLException se) {

System.out.println(se.getMessage()); //오류 출력

}

}

private void SexgetDBSearch(String n) {

Connection con = null;

Statement stmt = null;

try {

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

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

stmt = con.createStatement();

ResultSet result = stmt.executeQuery("SELECT \* FROM customer WHERE sex ='" + n + "'"); //성별로 검색

String count = Area.getText(); //area의 데이터를 가져온다.

int c = count.length();

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

while (result.next()) {

String Name = result.getString(1);

String age = Integer.toString(result.getInt(2));

String sex = result.getString(3);

//받아온 문자열을 조건에 맞게 배치

String value = "이름 : " + Name + " 나이 : " + age + " 성별 : " + sex + "\n"; //문자열에 저장

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

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

}

con.close(); //닫기

stmt.close(); //닫기

} catch (SQLException se) {

System.out.println(se.getMessage()); //오류 출력

}

}

private void Insert(String name, int age, String sex) { //데이터에 추가하는 메소드

try {

Connection con;

Statement stmt; // Connection 객체를 통해 DB에 연결

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

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

stmt = con.createStatement();

stmt.executeUpdate("INSERT INTO customer VALUES(' " + name + " ', ' " + age + " ',' " + sex + " ')");

// insert 명령어를 통해 입력받은 값을 db에 추가

stmt.close(); //닫기

con.close(); //닫기

}

catch (SQLException se) {

System.err.println(se.getMessage()); //에러시 출력

}

}

private void customer\_Delete(String r) { //데이터를 삭제하는 메소드

try {

Connection con;

Statement stmt;

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

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

stmt =con.createStatement();

stmt.executeUpdate("delete from customer where name = ' " + r + " ' ");

//delete 명령어를 통해 입력받은 값을 삭제

stmt.close(); //닫기

con.close(); //닫기

}

catch (SQLException se) {

System.err.println(se.getMessage()); //에러 출력

}

}

private void customer\_Update() {

try {

Connection con;

Statement stmt;

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

con.setCatalog("20145165");

stmt =con.createStatement();

stmt.executeUpdate("UPDATE customer SET age = age+1");

//update를 한다.

stmt.close();

con.close();

String count = Area.getText();

int c = count.length();

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

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

String value = "Update Complete"; //출력할 문자열를 저장

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

}

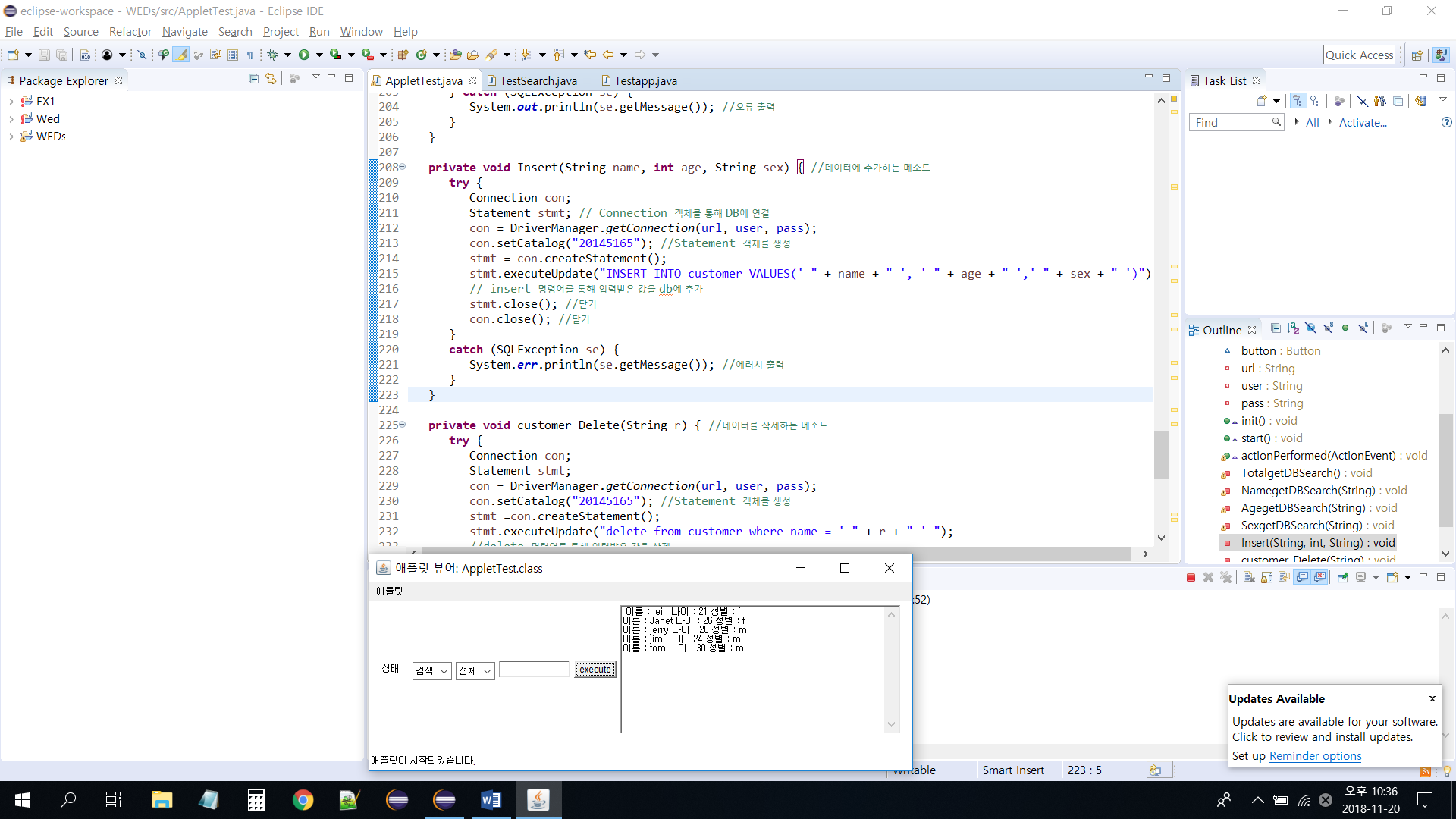
catch (SQLException se) {

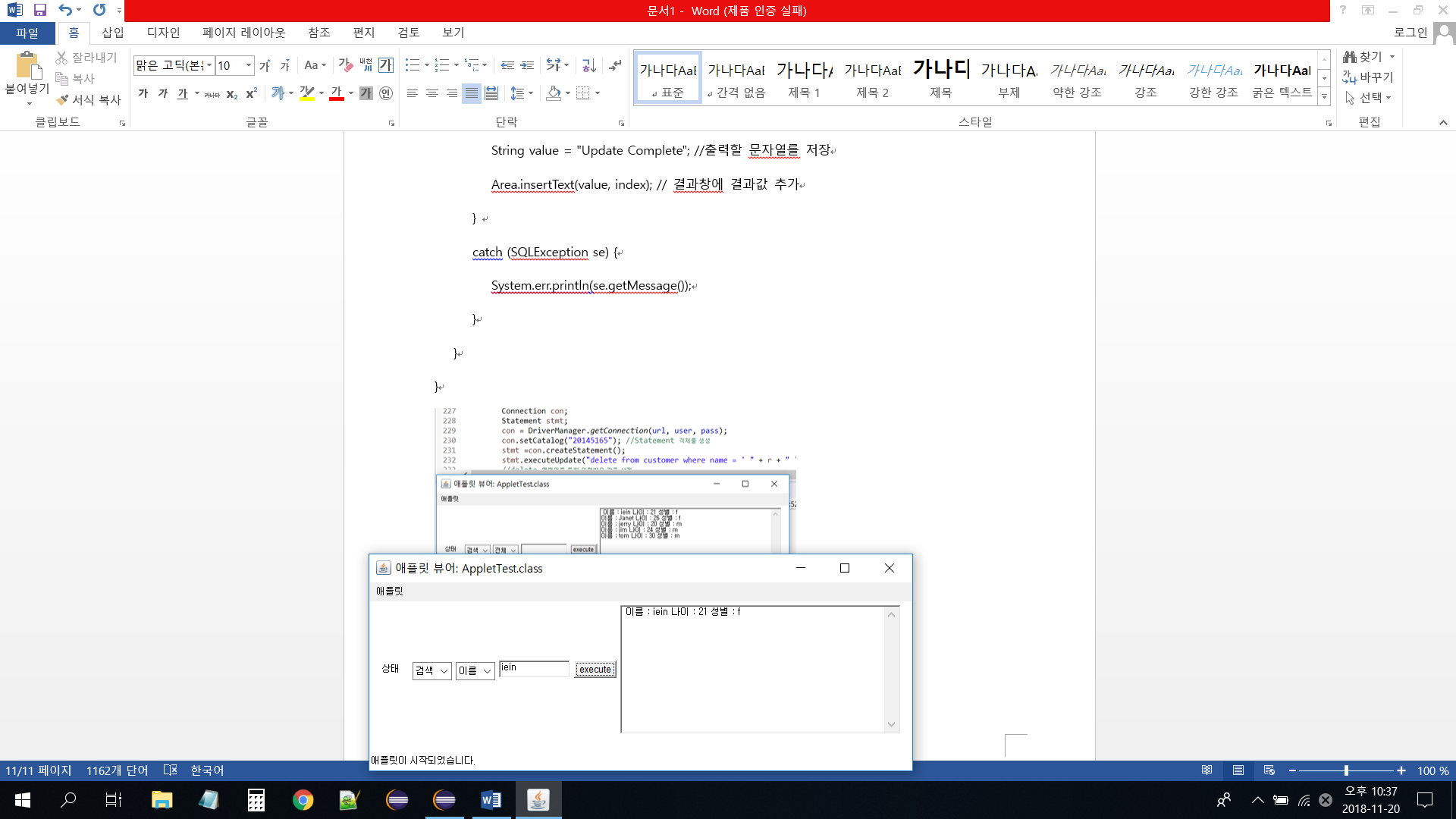
System.err.println(se.getMessage());

}

}

}





만든 테이블에 삽입하는 과제

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.sql.\*;

public class Testapp extends Applet implements ActionListener {

Choice search; // 선택 박스 선언

Choice status;

Label label; // 라벨 선언

TextField textFieldname;// 값을 입력받을 텍스트 필드 선언

TextField textFieldage;// 값을 입력받을 텍스트 필드 선언

TextField textFieldID;// 값을 입력받을 텍스트 필드 선언

TextField textFieldpass;// 값을 입력받을 텍스트 필드 선언

TextField textFieldmail;// 값을 입력받을 텍스트 필드 선언

TextField textFieldphone;// 값을 입력받을 텍스트 필드 선언

TextArea Area; // 결과 값을 출력할 텍스트에리어 선언

Button button; // 검색버튼 선언

private String url = "jdbc:inetdae7://210.115.229.77:2433";

private String user = "20145165";

private String pass = "s145165@Hallym";

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() { //start하는 메소드

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

label = new Label("이름"); //라벨에 값 입력

add(label); //라벨 추가

textFieldname = new TextField(10); //textfield 10칸짜리 선언

add(textFieldname); //추가

label = new Label("나이");

add(label);

textFieldage = new TextField(10);

add(textFieldage);

label = new Label("ID");

add(label);

textFieldID = new TextField(10);

add(textFieldID);

label = new Label("password");

add(label);

textFieldpass = new TextField(10);

add(textFieldpass);

label = new Label("e-mail");

add(label);

textFieldmail = new TextField(30);

add(textFieldmail);

label = new Label("phone");

add(label);

textFieldphone = new TextField(10);

add(textFieldphone);

button = new Button("입력");

add(button);

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

}

public void actionPerformed(ActionEvent ae) {

Connection con = null;

Statement stmt = null;

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

if (st.equals("입력")) { //버튼을 클릭시 textfield에있는 값을 변수에 저장

String name = textFieldname.getText();

String age = textFieldage.getText();

String ID = textFieldID.getText();

String passW = textFieldpass.getText();

String mail = textFieldmail.getText();

String phone = textFieldphone.getText();

Insert(name,age,ID,passW,mail,phone); //추가하는 메소드 선언

}

}

private void Insert(String name, String age, String ID, String passW, String mail, String Ph) {

try {

Connection con;

Statement stmt; // Connection 객체를 사용하여 DB에 연결을 설정

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

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

stmt = con.createStatement();

stmt.executeUpdate("INSERT INTO Hallymtabel VALUES(' " + name + " ', ' " + age + " ',' " + ID + " ',' " + passW + " ',' " + mail + " ',' " + Ph + " ')");

//insert를 사용해 문자열를 db에 추가

stmt.close(); //닫기

con.close();

}

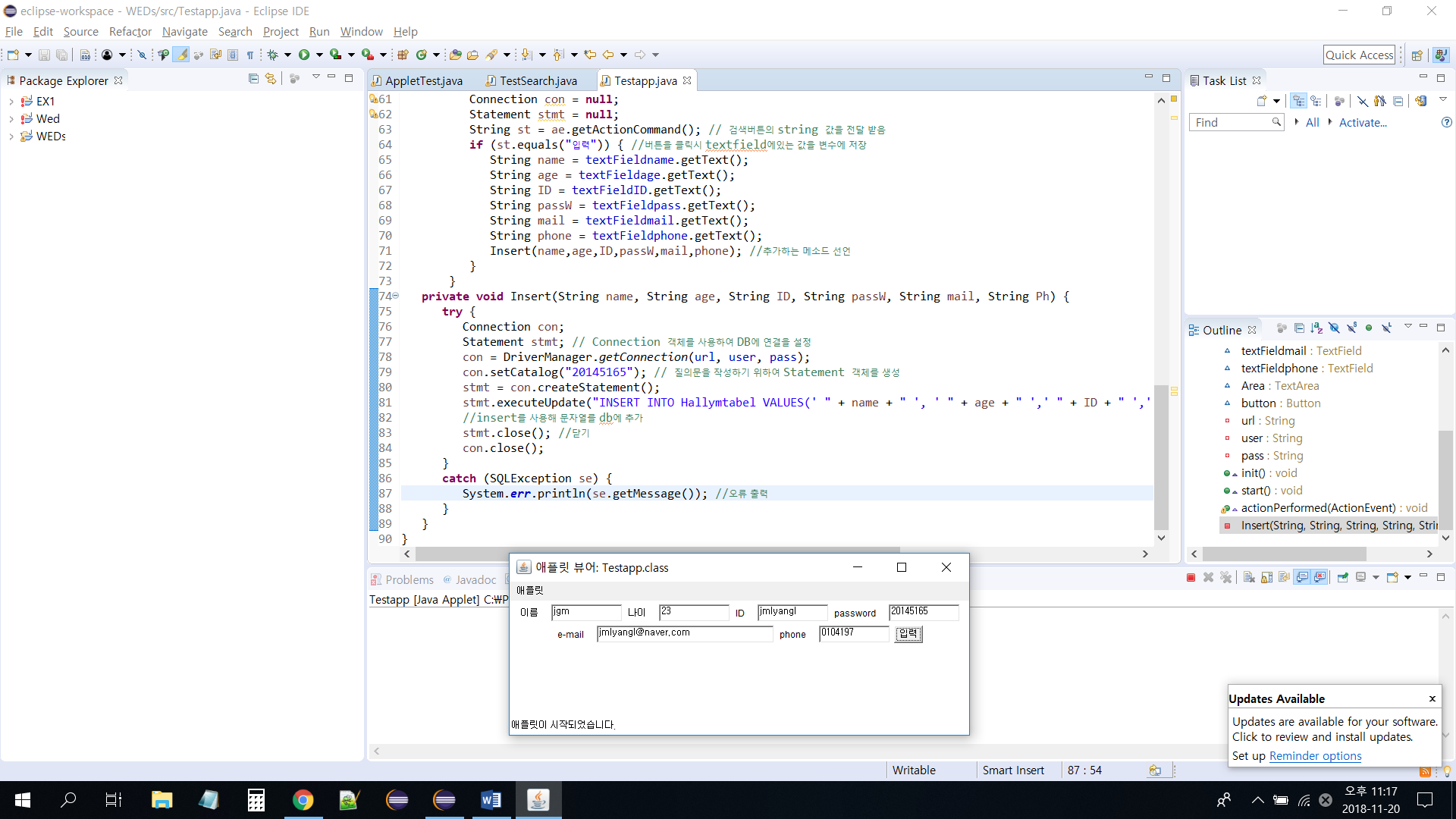
catch (SQLException se) {

System.err.println(se.getMessage()); //오류 출력

}

}

}



출력하는 과제

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.sql.\*;

public class TestSearch extends Applet implements ActionListener {

Label label; // 라벨 선언

TextField textField; // 값을받는 텍스트 필드 선언

TextArea Area; // 결과 값을 출력할 에리어 선언

Button button; //검색버튼 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() { //start하는 메소드

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

label = new Label("검색"); //라벨에 문자 입력

add(label); //라벨 추가

button = new Button("search"); //버튼에 문자 입력

add(button); //버튼 추가

Area = new TextArea(10, 50); //area 범위 지정

add(Area); //area 추가

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

}

public void actionPerformed(ActionEvent ae) { //버튼을 클릭시 작업을 실행한다.

Connection con = null;

Statement stmt = null;

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

if (st.equals("search")) { //문자열 내용이 search와 같으면

TotalgetDBSearch(); //실행

}

}

private void TotalgetDBSearch() {

Connection con = null;

Statement stmt = null;

try {

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

con.setCatalog("20145165");

stmt = con.createStatement();

ResultSet result = stmt.executeQuery("SELECT \* FROM Hallymtabel"); //모든 값을 검색한다.

while (result.next()) {

String Name = result.getString(1);

String age = Integer.toString(result.getInt(2));

String ID = result.getString(3);

String pass = result.getString(4);

String mail = result.getString(5);

String phone = result.getString(6);

//result값이 없을때까지 반복하면서 하나하나 변수 문자열을 저장

String value = "이름 : " + Name + " 나이 : " + age + " ID : " + ID + "password" + pass + "e-mail" + mail + "phone"+ phone + "\n";

//문자열 저장

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

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

}

con.close(); //닫기

stmt.close(); //닫기

} catch (Exception ee) {

System.err.println(ee); //오류 출력

}

}

}

