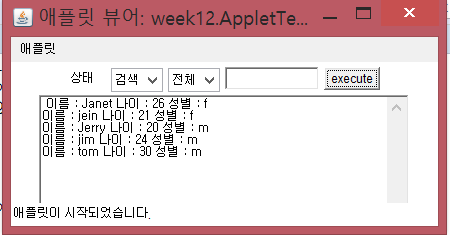
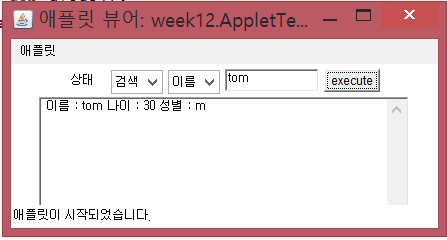
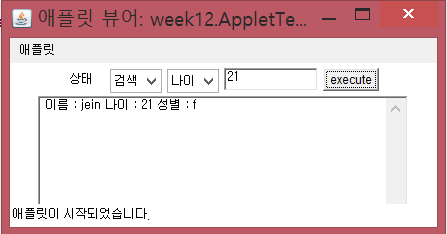
전체 검색



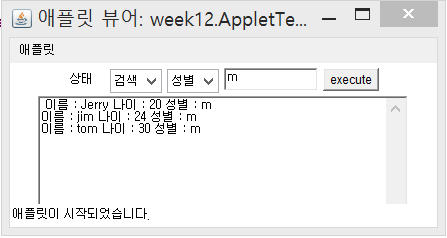
이름 검색



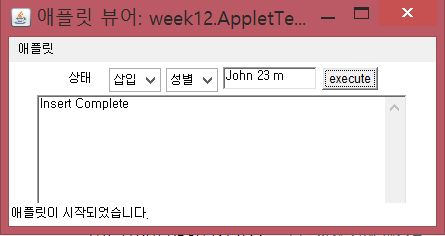
나이 검색

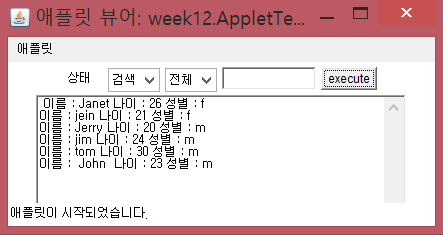


성별 검색

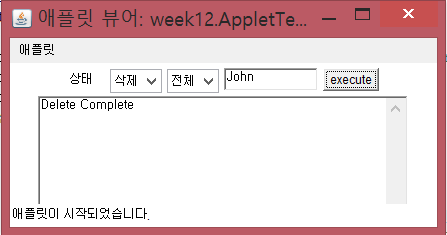


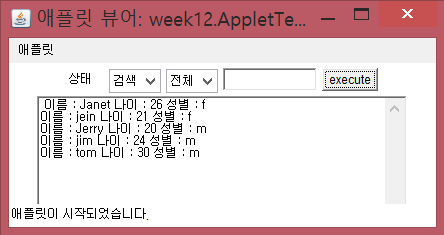
삽입



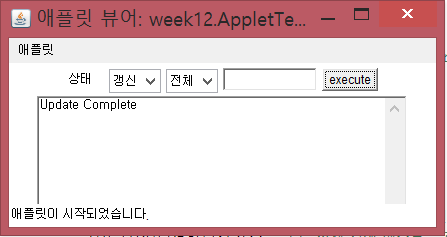


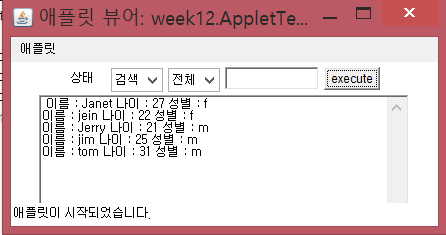
삭제





갱신





소스코드

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

import java.sql.\*;

public class AppletTest extends Applet implements ActionListener {

Choice menu;

Choice search; // 선택 박스 선언

Label label; // 라벨 선언

TextField textField; // 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 = "20165164";

private String pass = "123123";

public void init() {

try {

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

} catch (ClassNotFoundException e) {

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

}

}

public void start() {

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

label = new Label("상태");

add(label);

menu = new Choice();

menu.add("삽입");

menu.add("검색");

menu.add("삭제");

menu.add("갱신");

add(menu);

search = new Choice();

search.add("전체");

search.add("이름");

search.add("나이");

search.add("성별");

add(search);

textField = new TextField(10);

add(textField);

button = new Button("execute");

add(button);

Area = new TextArea(10, 50);

add(Area);

button.addActionListener(this);

}

public void actionPerformed(ActionEvent ae) {

Connection con = null;

Statement stmt = null;

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

String menuitem = menu.getSelectedItem();

String item = search.getSelectedItem();

// 선택박스에서 선택된 값을 전달 받음

if (st.equals("execute")) {

String n = textField.getText();

if (menuitem.equals("검색")) {

if (item.equals("전체")) { // 선택박스에서 ‘전체’ 가 선택 됐을 경우

TotalgetDBSearch(); // 전체 검색 메소드를 호출

} else if (item.equals("이름")) {

NamegetDBSearch(n);

} else if (item.equals("나이")) {

AgegetDBSearch(n);

} else if (item.equals("성별")) {

SexgetDBSearch(n);

}

} else if (menuitem.equals("삽입")) {

String[] str = n.split(" ");

Insert(str[0], Integer.parseInt(str[1]), str[2]);

} else if (menuitem.equals("삭제")) {

customer\_Delete(n);

} else if (menuitem.equals("갱신")) {

Update();

}

}

}

private void TotalgetDBSearch() {

Connection con = null;

Statement stmt = null;

try {

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

con.setCatalog("20165164");

stmt = con.createStatement();

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();

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);

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

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

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

}

con.close();

stmt.close();

} catch (Exception ee) {

System.out.println(ee);

}

} // try-end

private void NamegetDBSearch(String n) {

Connection con = null;

Statement stmt = null;

try {

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

con.setCatalog("20165164");

stmt = con.createStatement();

ResultSet result = stmt.executeQuery("SELECT \* FROM customer WHERE name ='" + 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();

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);

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

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

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

}

con.close();

stmt.close();

} catch (Exception ee) {

System.out.println(ee);

}

}

private void AgegetDBSearch(String n) {

Connection con = null;

Statement stmt = null;

try {

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

con.setCatalog("20165164");

stmt = con.createStatement();

ResultSet result = stmt.executeQuery("SELECT \* FROM customer WHERE age =" + 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();

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);

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

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

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

}

con.close();

stmt.close();

} catch (Exception ee) {

System.out.println(ee);

}

}

private void SexgetDBSearch(String n) {

Connection con = null;

Statement stmt = null;

try {

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

con.setCatalog("20165164");

stmt = con.createStatement();

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

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);

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

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

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

}

con.close();

stmt.close();

} catch (Exception ee) {

System.out.println(ee);

}

}

private void Insert(String name, int age, String sex) {

try {

Connection con;

Statement stmt;

// Connection 객체를 사용하여 DB에 연결을 설정

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

con.setCatalog("20165164");

// 질의문을 작성하기 위하여 Statement 객체를 생성

stmt = con.createStatement();

// 생성된 테이블에 값을 삽입

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

String count = Area.getText();

int c = count.length();

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

Area.insertText("Insert Complete", 0);

// Statement,Connection 객체를 닫음

stmt.close();

con.close();

} catch (SQLException se) {

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

}

}

private void customer\_Delete(String r) {

try {

Connection con;

Statement stmt;

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

con.setCatalog("20165164");

// 연결된 DB에서 질의문을 작성하기 위하여 Statement 객체를 생성

stmt = con.createStatement();

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

String count = Area.getText();

int c = count.length();

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

Area.insertText("Delete Complete", 0);

stmt.close();

con.close();

} catch (SQLException se) {

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

}

}

private void Update() {

try {

Connection con;

Statement stmt;

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

con.setCatalog("20165164");

stmt = con.createStatement();

stmt.executeUpdate("update customer set age=age+1");

String count = Area.getText();

int c = count.length();

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

Area.insertText("Update Complete", 0);

stmt.close();

con.close();

} catch (SQLException se) {

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

}

}

} // end