

# 프로그래밍 기초 및 응용

## 프로그래밍 언어 응용

날짜 : 2022/ 09/ 23 (금)

이름 : 김재준

# 목차

- › 프로젝트 생성
- › 라이브러리 설정
- › 프로그램 구성
- › 프로그램 작성

# 과정 수행

# 1. 프로젝트 생성

- 프로젝트명 : UserManager
- 자바버전 : Java8 이상
- DBMS : MySQL 8
- 개발도구 : Eclipse, Workbench

> Ch14 [Java main]  
> Ch15 [Java main]  
> Exam [Java main]  
> > UserManager [Java main]

프로젝트명  
: UserManager

> > UserManager [Java main]  
> JRE System Library [JavaSE-17]  
src  
> main  
> Referenced Libraries

자바버전



MySQL Workbench 8.0 CE

DBMS



MySQL  
Workben...

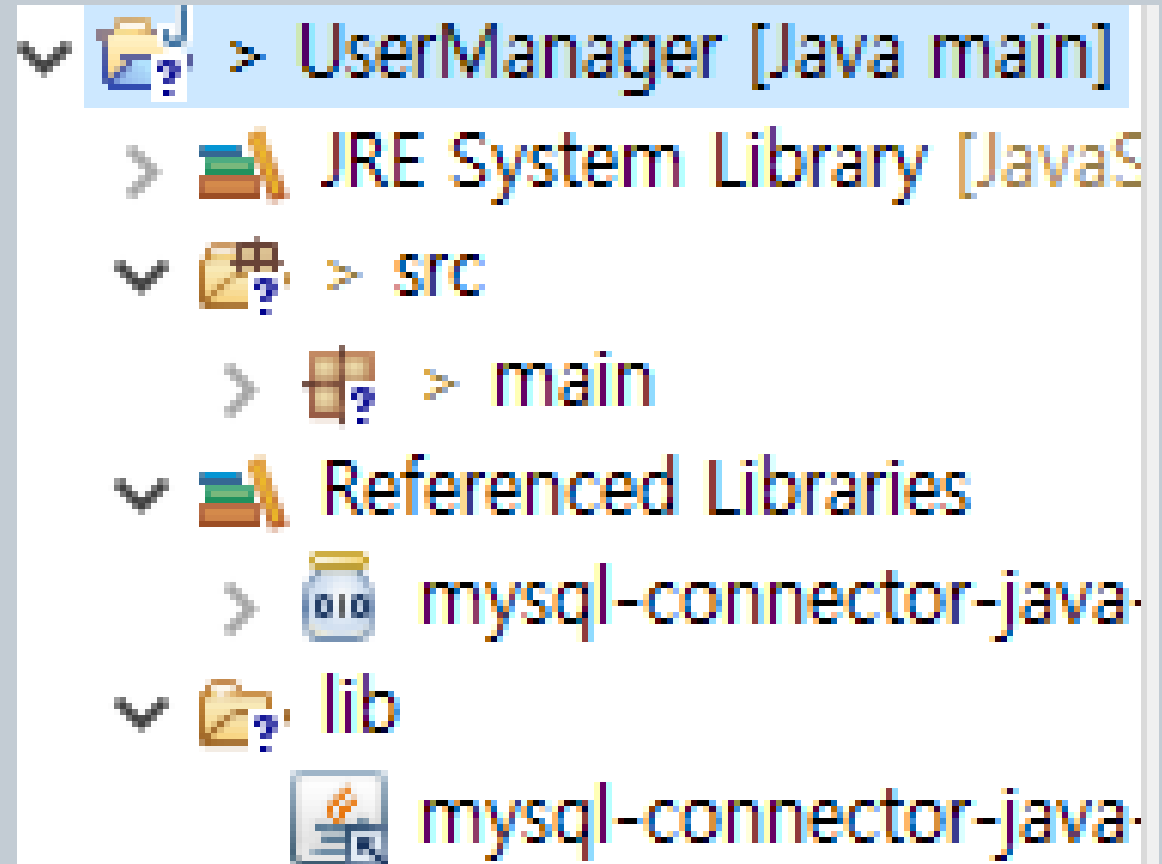


eclipse.exe -  
바로 가기

개발 도구

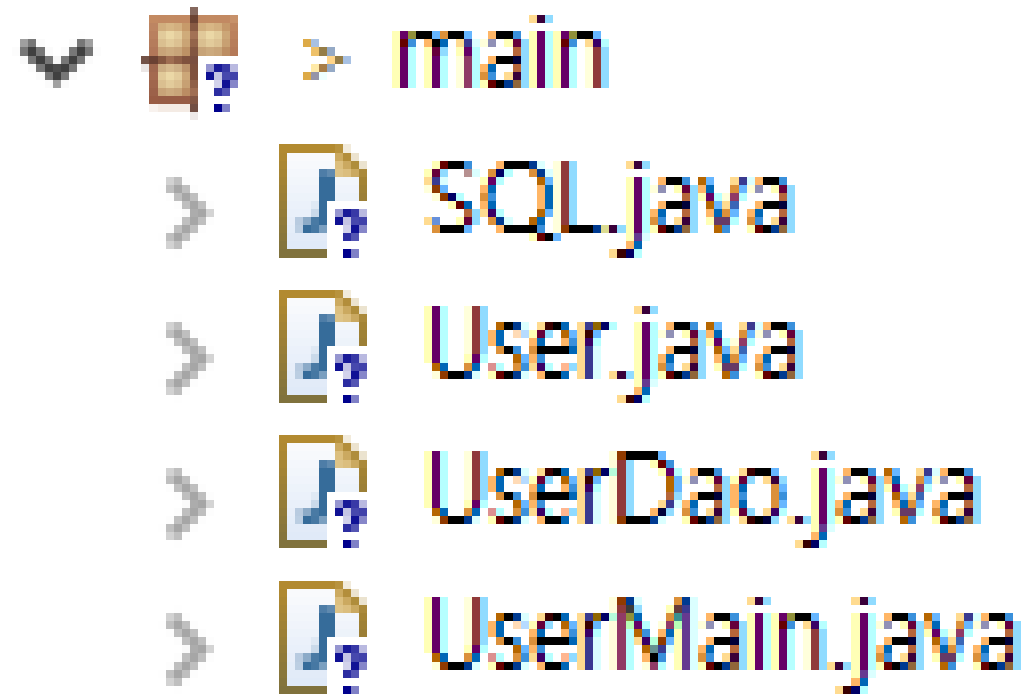
## 2. 라이브러리 설정

- 프로젝트 – 새폴더 – lib 생성
  - mysql-connector-java-8.0.29.jar 이동
- 프로젝트 – Build Path
  - Add External Archives... 선택
  - mysql-connector-java-8.0.29.jar 선택



### 3. 프로그램 구성

- 프로젝트 – new – Class
  - 패키지명 : main / 클래스명 :SQL 생성
- 프로젝트 – new – Class
  - 패키지명 : main / 클래스명 :User 생성
- 프로젝트 – new – Class
  - 패키지명 : main / 클래스명 :UserDao 생성
- 프로젝트 – new – Class
  - 패키지명 : main / 클래스명 :UserMain 생성



## 4. 프로그램 작성

### · 빈칸 코드 완성 및 기능 구현하기

조회, 입력

```
*****
회원관리매니저 v1.0
*****

종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 2

-----전체조회-----
a101,김유신,010-1234-1001,25
a102,김준추,010-1234-1002,23
a103,장보고,010-1234-1003,33
a104,강감찬,010-1234-1004,43
a105,이순신,010-1234-1005,53
a201,김유신,null,25
a301,김유신,null,25

종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 1
아이디 입력 : a103
이름 입력 : 장보고
휴대폰 입력 : 010-1234-1003
나이 입력 : 31
아이디 또는 휴대폰 중복입니다.
입력완료...

종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 0
프로그램 종료...
```

검색, 삭제

```
*****
회원관리매니저 v1.0
*****

종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 3
이름검색 : 김유신

-----검색결과-----
a101,김유신,010-1234-1001,25

종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 4
삭제아이디 : a101
아이디 a101 삭제완료

종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 2

-----전체조회-----
a102,김준추,010-1234-1002,23
a103,장보고,010-1234-1003,33
a104,강감찬,010-1234-1004,43
a105,이순신,010-1234-1005,53
a201,김유신,null,25
a301,김유신,null,25

종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 0
프로그램 종료...
```

- Table은 Java2DB의 User3 테이블을 사용한다.

```
1 package main;
2
3 public class SQL {
4     public static final String INSERT_USER = "INSERT INTO `User3` VALUES (?, ?, ?, ?);";
5     public static final String SELECT_USERS = "SELECT * FROM `User3`;";
6     public static final String SELECT_USER = "SELECT * FROM `User3` WHERE `name`=?;";
7     public static final String DELETE_USER = "DELETE FROM `User3` WHERE `uid`=?;";
8 }
```

The screenshot shows a database management interface. On the left, the 'Schemas' pane displays a tree view of the 'java2db' database, with the 'user3' table selected. The 'Information' pane at the bottom left shows the table's structure:

**Table: user3**

**Columns:**

- uid: varchar(20) PK
- name: varchar(10)
- hp: char(13)
- age: tinyint

The main window displays a query: `SELECT * FROM java2db.user3;` The 'Result Grid' at the bottom right shows the following data:

	uid	name	hp	age
▶	a102	김춘추	010-1234-1002	23
	a103	장보고	010-1234-1003	33
	a104	강감찬	010-1234-1004	43
	a105	이순신	010-1234-1005	53
	a201	김유신	NULL	25
	a301	김유신	NULL	25
*	NULL	NULL	NULL	NULL



# 풀이 과정

## UserDao.java 코드작성

```
1 package main;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.sql.SQLIntegrityConstraintViolationException;
9 import java.util.ArrayList;
10 import java.util.List;
11
12
13 public class UserDao {
14     // 싱글톤
15     private static UserDao instance = new UserDao();
16     public static UserDao getInstance() {
17         return instance;
18     }
19
20     private UserDao() {}
21
22     // DB정보
23     String host = "jdbc:mysql://127.0.0.1:3306/java2db";
24     private String user = "root";
25     private String pass = "1234";
26
27     private Connection getConnection() throws ClassNotFoundException, SQLException{
28         Class.forName("com.mysql.cj.jdbc.Driver");
29         Connection conn = DriverManager.getConnection(host, user, pass);
30         return conn;
31     }
32
33     private Connection conn;
34     private PreparedStatement psmt;
35     private ResultSet rs;
36
```

```
37 public void insertUser(User user) {
38     try {
39         conn = getConnection();
40         psmt = conn.prepareStatement(SQL.INSERT_USER);
41         psmt.setString(1, user.getId());
42         psmt.setString(2, user.getName());
43         psmt.setString(3, user.getHp());
44         psmt.setInt(4, user.getAge());
45         psmt.executeUpdate();
46         close();
47     } catch (SQLIntegrityConstraintViolationException e) {
48         System.out.println("아이디 또는 휴대폰 중복입니다.");
49     } catch (ClassNotFoundException e) {
50         e.printStackTrace();
51     } catch (SQLException e) {
52         e.printStackTrace();
53     }
54 }
55
56 public List<User> selectUsers(){
57     List<User> users = new ArrayList<>();
58
59     try {
60         conn = getConnection();
61         psmt = conn.prepareStatement(SQL.SELECT_USERS);
62         rs = psmt.executeQuery();
63
64         while(rs.next()) {
65             User user = new User();
66             user.setId(rs.getString(1));
67             user.setName(rs.getString(2));
68             user.setHp(rs.getString(3));
69             user.setAge(rs.getInt(4));
70
71             users.add(user);
72         }
73     }
74 }
```

```
73         close();
74     } catch (Exception e) {
75         e.printStackTrace();
76     }
77 }
78
79     return users;
80 }
81
82 public User selectUser(String name) {
83     User user = new User();
84
85     try {
86
87         conn = getConnection();
88         psmt = conn.prepareStatement(SQL.SELECT_USER);
89         psmt.setString(1, name);
90
91         rs = psmt.executeQuery();
92
93         if(rs.next()) {
94             user.setUid(rs.getString(1));
95             user.setName(rs.getString(2));
96             user.setHp(rs.getString(3));
97             user.setAge(rs.getInt(4));
98         }
99         close();
100     } catch (Exception e) {
101         e.printStackTrace();
102     }
103     return user;
104 }
105
106
107 public int deleteUser(String name) {
108
```

```
109     int result = 0;
110
111     try {
112         conn = getConnection();
113         psmt = conn.prepareStatement(SQL.DELETE_USER);
114         psmt.setString(1, name);
115
116         result = psmt.executeUpdate();
117
118         close();
119
120     } catch (Exception e) {
121         e.printStackTrace();
122     }
123
124     return result;
125 }
126
127 private void close() throws SQLException{
128     if(rs != null) {
129         rs.close();
130     }
131
132     if(psmt != null) {
133         psmt.close();
134     }
135
136     if(conn != null) {
137         conn.close();
138     }
139 }
140 } // UserDao end
141
```

# 풀이 과정

## UserMain.java 코드작성

```

1 package main;
2
3 import java.util.List;
4 import java.util.Scanner;
5
6 public class UserMain {
7     public static void main(String[] args) {
8         System.out.println("*****");
9         System.out.println("회원관리매니저 v1.0");
10        System.out.println("*****");
11
12        Scanner sc = new Scanner(System.in);
13
14        while(true) {
15            System.out.println("-----");
16            System.out.println("종료:0, 입력:1, 조회:2, 검색:3, 삭제:4");
17            System.out.print("선택 : ");
18            int answer = sc.nextInt();
19
20            if(answer == 0) {
21                break;
22            }
23            else if(answer == 1) {
24                // 데이터 입력
25                User user = new User();
26
27                System.out.print("아이디 입력 : ");
28                user.setUid(sc.next());
29
30                System.out.print("이름 입력 : ");
31                user.setName(sc.next());
32
33                System.out.print("휴대폰 입력 : ");
34                user.setHp(sc.next());
35
36                System.out.print("나이 입력 : ");
37
38
39                user.setAge(sc.nextInt());
40
41                UserDao.getInstance().insertUser(user);
42                System.out.println("입력완료...");
43
44            }else if(answer == 2) {
45                // 데이터 조회
46                List<User> users = UserDao.getInstance().selectUsers();
47
48                System.out.println("-----전체조회-----");
49                for(User user : users) {
50                    System.out.println(user);
51                }
52
53            }else if(answer == 3) {
54                // 데이터 검색
55                System.out.print("이름검색 : ");
56                String name = sc.next();
57
58                User user = UserDao.getInstance().selectUser(name);
59
60                System.out.println("-----검색결과-----");
61                System.out.println(user);
62
63            }else if(answer == 4) {
64                // 데이터 삭제
65                System.out.print("삭제아이디 : ");
66                String uid = sc.next();
67
68                int result = UserDao.getInstance().deleteUser(uid);
69
70                if(result >= 1) {
71                    System.out.println("아이디 " + uid + " 삭제완료");
72                }else {
73                    System.out.println("아이디 " + uid + " 사용자가 존재하지 않습니다.");
74                }
75            }
76        }
77        sc.close();
78        System.out.println("프로그램 종료...");
79    }
80}

```

```
public static void main(String[] args) {
    System.out.println("과제를 종료합니다.");
    System.exit(0);
}
```

