프로그래밍 언어 응용

2022/09/15 공민혁

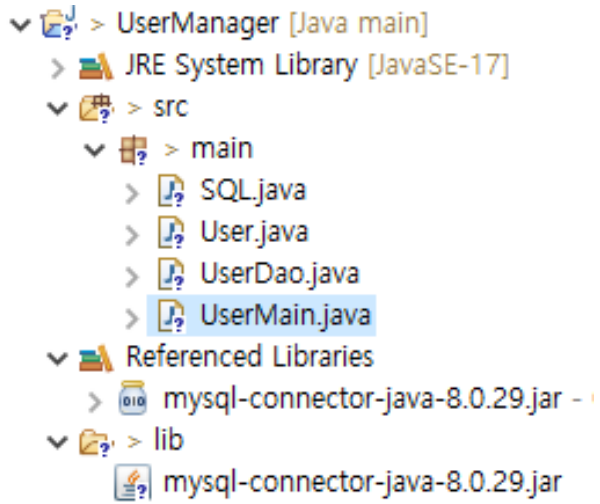
목차

화면구성

코드분석

PART 1

화면구성



이클립스를 mysql 와 연동시키기위해

라이브러리 폴더 안에 connector 파일을 넣음

(connector파일을 프로젝트안에 보관하기 위해)

프로젝트에 bulid path 를 시켜 workbench 와 eclipse 를 연결한다.

```java
public class UserDao {

    private static UserDao instance = new UserDao();
    public static UserDao getInstance() {
        return instance;
    }
    private UserDao() {}

    String host = "jdbc:mysql://127.0.0.1:3306/java1db";
    private String user = "root";
    private String pass = "1234";

    private Connection getConnection() throws ClassNotFoundException, SQLException {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection conn = DriverManager.getConnection(host, user, pass);
        return conn;
    }
    private Connection conn;
    private PreparedStatement psmt;
    private ResultSet rs;
```

UserDao클래스 를 호출받기 위해 메서드 선언

1.DB 정보를 입력

2.Jdbc 드라이버를 로드시킨다.

3.DriverManager 를 통해 데이터베이스 접속

PART 1

화면구성

```java
public class User {
    private String uid;
    private String name;
    private String hp;
    private int age;

    public String getUid() {
        return uid;
    }
    public void setUid(String uid) {
        this.uid = uid;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getHp() {
        return hp;
    }
    public void setHp(String hp) {
        this.hp = hp;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }

    @Override
    public String toString() {
        return uid+","+name+","+hp+","+age;
    }
}
```

주고받을 데이터 형태 클래스로 만들기

코드의 가독성 을 높이기 위해

User.java 클래스 생성

Set 메서드 는 매개변수를 주고

Get 메서드는 리턴값을 받도록함

PART 1

화면구성

```
public class SQL {

    public static final String INSERT_USER = "INSERT INTO `User3` VALUES (?,?,?,?);";
    public static final String SELECT_USERS = "SELECT * FROM `User3`;";
    public static final String SELECT_USER = "SELECT * FROM `User3` WHERE `name`=?;";
    public static final String DELETE_USER = "DELETE FROM `User3` WHERE `Uid`=?";

}
```

SQL 쿼리문을 실행시키기위해
SQL.java 클래스 생성

prepareStatement 클래스를 사용하고
SQL 문의 값을 동적으로 사용하기 위해 기호(?)사용

PART 1

화면구성

```
public User selectUser(String name) {
    User user = new User();

    try {
        conn = getConnection();
        psmt = conn.prepareStatement(SQL.SELECT_USER);
        psmt.setString(1, name);

        rs = psmt.executeQuery();

        if(rs.next()) {
            user.setUid(rs.getString(1));
            user.setName(rs.getString(2));
            user.setHp(rs.getString(3));
            user.setAge(rs.getInt(4));
        }
        close();
    }catch (Exception e) {
        e.printStackTrace();
    }
    return user;
}
```

UserDao의 화면중 select문이며

이클립스에서 SQL 구문을 실행하기 위해 PrepareStatement 클래스 선언

Select 를 실행시키기 위해 executeQuery 메서드 사용 (반환됨)

Resultset 을 통해 쿼리 결과를 가져 올 수 있음

PART 1

화면구성

```
public void insertUser(User user) {
    try {
        conn = getConnection();
        psmt = conn.prepareStatement(SQL.INSERT_USER);
        psmt.setString(1, user.getUid());
        psmt.setString(2, user.getName());
        psmt.setString(3, user.getHp());
        psmt.setInt(4, user.getAge());
        psmt.executeUpdate();
        close();
    }catch(SQLIntegrityConstraintViolationException e) {
        System.out.println("아이디 또는 휴대폰 중복입니다.");
    }catch(ClassNotFoundException e) {
        e.printStackTrace();
    }catch(SQLException e) {
        e.printStackTrace();
    }
}
```

```
public int deleteUser(String name) {

    int result = 0;

    try {
        conn = getConnection();
        psmt = conn.prepareStatement(SQL.DELETE_USER);
        psmt.setString(1, name);

        result = psmt.executeUpdate();

        close();
    }catch(Exception e) {
        e.printStackTrace();
    }
    return result;
}
```

UserDao의 화면중 insert, delete문이며

이클립스에서 SQL 구문을 실행하기 위해 PrepareStatement 클래스 선언

Connection 객체를 가져오기 위해 getConnection 메서드 사용

(Insert / delete) 를 실행 시키기 위해 executeUpdate 메서드 사용

PART 1

화면구성

```
public List<User> selectUsers() {
    List<User> users = new ArrayList<>();

    try {
        conn = getConnection();
        psmt = conn.prepareStatement(SQL.SELECT_USERS);
        rs = psmt.executeQuery();

        while(rs.next()) {
            User user = new User();
            user.setUid(rs.getString(1));
            user.setName(rs.getString(2));
            user.setHp(rs.getString(3));
            user.setAge(rs.getInt(4));

            users.add(user);
        }
        close();
    }catch (Exception e) {
        e.printStackTrace();
    }
    return users;
}
```

UserDao의 화면중 List메서드 이며

데이터베이스 조회하기 위해 list 메서드 선언

Connection 객체를 가져오기 위해 getConnection 메서드 사용

화면구성

```
public static void main(String[] args) {
    System.out.println("************************");
    System.out.println("회원관리매니저 v1.0");
    System.out.println("************************");

    Scanner sc = new Scanner(System.in);

    while(true) {
        System.out.println("------------------------");
        System.out.println("종료:0, 입력:1, 조회:2, 검색:3, 삭제:4");
        System.out.print("선택 :");
        int answer = sc.nextInt();



    }//while end
    sc.close();
    System.out.println("프로그램 종료...");
}
```

메인화면 코드 구성이다

메인 클래스를 실행 시켰을 때 먼저 출력되는 코드이며

입력받기 위해 Scanner 클래스 객체를 생성시키며

while 반복문을 통해 먼저 실행시키고

입력받은 값을 Int 타입으로 리턴시킨다

화면구성

```
public void close() throws SQLException {
    if(rs != null) {
        rs.close();
    }
    if(psmt != null) {
        psmt.close();
    }
    if(conn != null) {
        conn.close();
    }
}
```

UserDao의 화면중 close 메서드이며

지원 해제 하기위해 close 메서드 선언

PART 1

화면구성

```
if(answer == 0) {
    break ;
```

```
}else if(answer == 1) {
    //데이터 입력
    User user = new User();

    System.out.print("아이디 입력 : ");
    user.setUid(sc.next());

    System.out.print("이름 입력 : ");
    user.setName(sc.next());

    System.out.print("휴대폰 입력 : ");
    user.setHp(sc.next());

    System.out.print("나이 입력 : ");
    user.setAge(sc.nextInt());

    UserDao.getInstance().insertUser(user);
    System.out.println("입력완료...");
```

```
}else if (answer == 2) {
    // 데이터 조회
    List<User> users = UserDao.getInstance().selectUsers();

    System.out.println("-----------전체조회------------");
    for(User user : users) {
        System.out.println(user);
    }
```

```
}else if (answer == 3) {
    System.out.print("이름검색 : ");
    String name = sc.next();

    User user = UserDao.getInstance().selectUser(name);

    System.out.println("-----------검색결과-------------");
    System.out.println(user);
```

```
}else if(answer == 4) {
    System.out.print("삭제아이디 : ");
    String uid = sc.next();

    int result = UserDao.getInstance().deleteUser(uid);

    if(result >= 1) {
        System.out.println("아이디 " + uid + " 삭제완료");
    }else {
        System.out.println("아이디 " + uid + " 사용자가 존재하지 않습니다");
    }
}
```

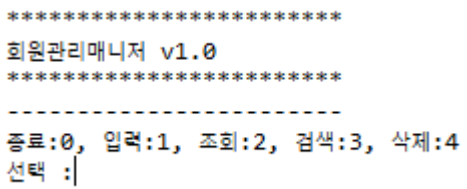
UserMain 화면의 코드들이며

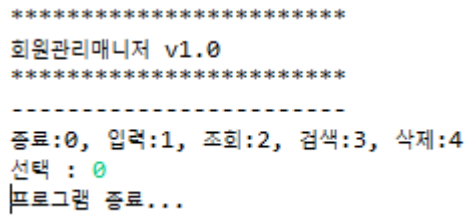
입력 받은값이 1, 2, 3, 4 일때 if문 실행

데이터베이스 작업을하는 클래스 UserDao 를 호출하기위해 인스턴스 메서드 선언

PART 2

결과 출력





실행 결과 화면이며

0을 입력했을때 결과 화면이다

PART 2

결과 출력

```
************************
회원관리매니저 v1.0
************************
------------------------
종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 1
아이디 입력 : a101
이름 입력 : 김유신
휴대폰 입력 : 010-1234-1001
나이 입력 : 25
입력완료...
------------------------
```

실행 결과 화면이며

1을 입력했을때 결과 화면이다

각각의 값을 입력 했을때

Workbench 의 User3 테이블에 데이터를 추가한다

uid	name	hp	age
a102	김춘추	010-1234-1002	23
a103	장보고	010-1234-1003	31
NULL	NULL	NULL	NULL

uid	name	hp	age
a101	김유신	010-1234-1001	25
a102	김춘추	010-1234-1002	23
a103	장보고	010-1234-1003	31
NULL	NULL	NULL	NULL

PART 2

결과 출력

```
회원관리매니저 v1.0
**********************
------------------------
종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 2
-----------전체조회------------
a102,김춘추,010-1234-1002,23
a103,장보고,010-1234-1003,31
------------------------
```

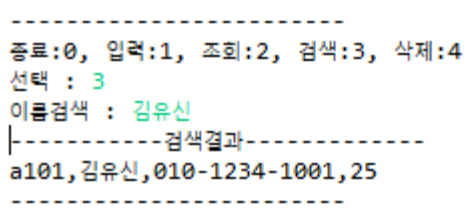
실행 결과 화면이며

2을 입력했을때 결과 화면이다

Workbench 의 User3 테이블 데이터값을 호출한 화면이다

PART 2

결과 출력



실행 결과 화면이며

3을 입력했을때 결과 화면이다

Workbench 의 User3 테이블 데이터값 name 의 김유신 데이터를 호출한 결과이다

PART 2

결과 출력

```
-----------검색결과-------------
a101,김유신,010-1234-1001,25
------------------------
종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 4
삭제아이디 : a101
아이디 a101 삭제완료
------------------------
```

실행 결과 화면이며

4을 입력했을때 결과 화면이다

Workbench 의 User3 테이블 데이터값 a101 의 컬럼값이 삭제가 된 후의 화면이다

	uid	name	hp	age
▶	a101	김유신	010-1234-1001	25
	a102	김춘추	010-1234-1002	23
	a103	장보고	010-1234-1003	31
*	NULL	NULL	NULL	NULL

	uid	name	hp	age
▶	a102	김춘추	010-1234-1002	23
	a103	장보고	010-1234-1003	31
*	NULL	NULL	NULL	NULL

끝