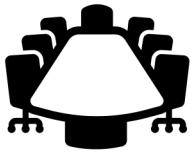


3조  
자바-SQL-JDBC  
프로젝트



## 소개

조 구성원 및  
Class 구성



## 기능

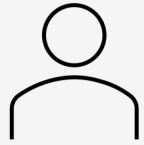
각 기능의 코드 보기



## 실행

코드 실행시키기  
오류사항 확인하기

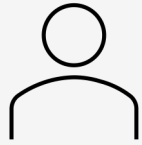
# 조 구성원 소개



11

노영걸

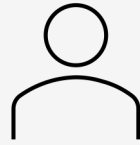
- 부서번호 검색



13

박주혁

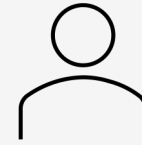
- 도시이름 검색



16

신소정

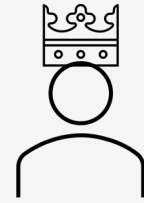
- 통계자료 출력  
- 발표



26

정은빈

- 사용자 이름 검색  
- PPT

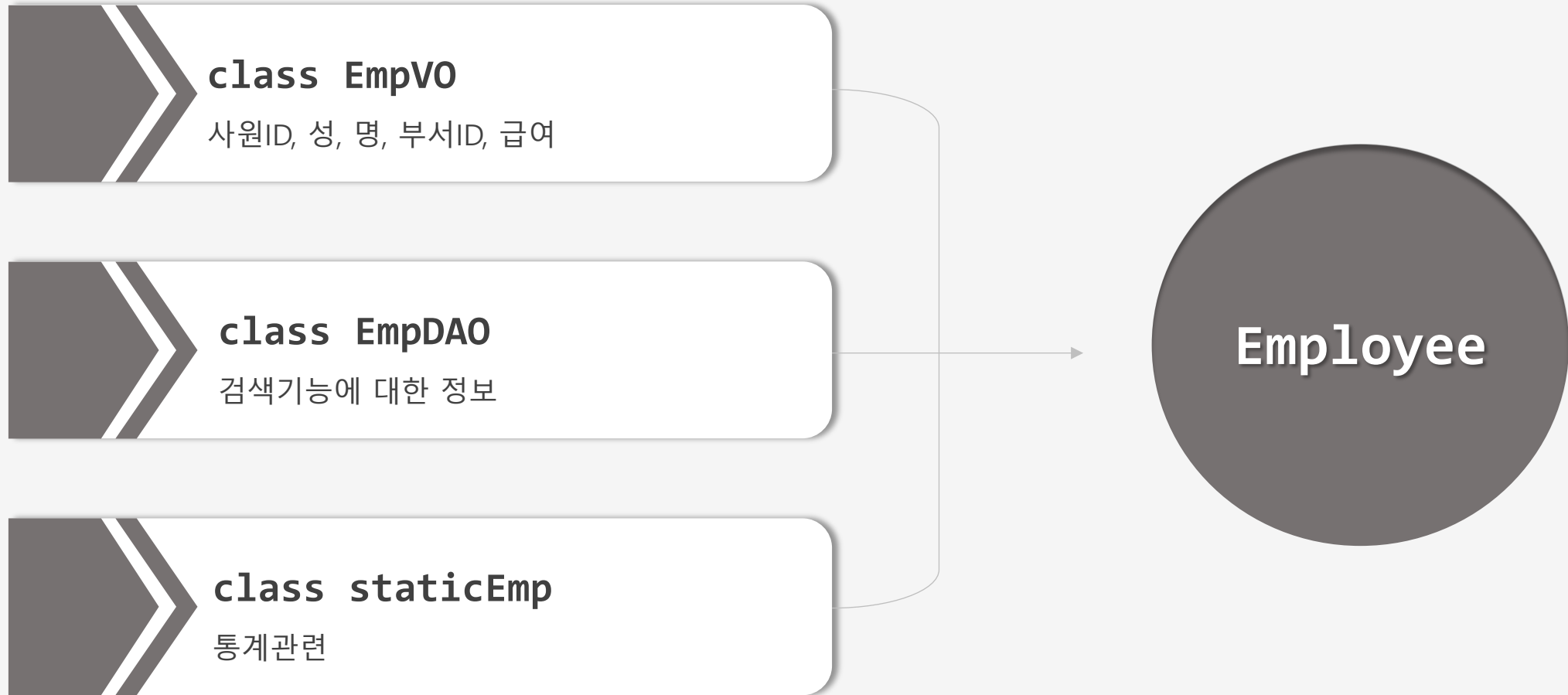


28

최연호

- 입사년도 검색  
- 부서장 이름 검색

# Class 구성



# Class 구성

```
public class Employee {  
    static Scanner scan = new Scanner(System.in);  
  
    public static void main(String[] args) {  
        staticEmp semp = new staticEmp();  
        System.out.println("직원관리정보시스템입니다.");  
        System.out.println();  
        while (true) {  
            System.out.println("1. 검색 2. 통계자료 3. 프로그램 종료");  
            System.out.print("입력 : ");  
            String select1 = scan.nextLine();  
        }  
    }  
}
```



```
switch (select1) {
case "1": // 검색
    EmpSelect();
    break;
case ("2"): // 통계자료
    try {
        semp.statistics();
    } catch (Exception e) {
        e.printStackTrace();
    }
    break;
case ("3"): // 프로그램종료
    System.out.println("프로그램종료");
    System.exit(0);
    break;
default:
```

# Class 구성

```
class EmpVO{
    public int empId;
    public String firstName;
    public String lastName;
    public int deptId;
    public int salary;

    static String[] nameTag = new String[] {
        "부서번호", "부서이름", "최대급여", "최소급여",
        "평균급여", "인원(명)", "사원번호", "성", "명", "입사일자", "급여"
    };
    @Override
    public String toString() {
        return "EmpVO [empId=" + empId + ", firstName=" + firstName +
            ", lastName=" + lastName + ", deptId=" + deptId +
            ", salary=" + salary + "]";
    }
}
```

- 1 사용자의 이름 검색하기
- 2 도시이름으로 검색하기
- 3 입사년도로 검색하기
- 4 부서장 이름으로 부서명 검색하기
- 5 부서번호로 검색하기
- 6 통계자료 출력하기



```
class EmpDAO { }
```

---

# 01. 사용자의 이름 검색하기


```
public void getEmpByName(String firstName, String lastName) throws
Exception {

    EmpVO result = new EmpVO();
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con = DriverManager.getConnection(
        "jdbc:oracle:thin:@localhost:1521:xe", "hr", "hr");

    String sql = "select substr(e.HIRE_DATE,1),e.*"
        + "FROM EMPLOYEES e where FIRST_NAME = ? and LAST_NAME = ?"
        + "order BY EMPLOYEE_ID";
    PreparedStatement stmt = con.prepareStatement(sql);
    stmt.setString(1, firstName);
    stmt.setString(2, lastName);
    ResultSet rs = stmt.executeQuery();
```

```
class EmpDAO { }
```

---



```
int cnt = 0;
System.out.printf("%s\t/%11s /%9s /%s\t/%9s / %s\n", EmpVO.nameTag[6], EmpVO.nameTag[7],
    EmpVO.nameTag[8], EmpVO.nameTag[0], EmpVO.nameTag[9], EmpVO.nameTag[10]);
while (rs.next()) {
    result.empId = rs.getInt("EMPLOYEE_ID");
    result.firstName = rs.getString("FIRST_NAME");
    result.lastName = rs.getString("LAST_NAME");
    result.salary = rs.getInt("SALARY");
    result.hireDate = rs.getString(1);
    result.deptId = rs.getInt("DEPARTMENT_ID");

    System.out.printf("%d\t/%11s /%10s /%d\t/%10s / $ %s\n", result.empId, result.firstName,
        result.lastName, result.deptId, result.hireDate, formatter.format(result.salary));
}
System.out.println();
}
```

```
class EmpDAO { }
```

---

## 02. 도시이름으로 검색하기

```
public ArrayList<EmpVO> getEmpListByCityName(String city) throws
Exception {

    ArrayList<EmpVO> list = new ArrayList<EmpVO>( );
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con = DriverManager.getConnection(
        "jdbc:oracle:thin:@localhost:1521:xe","hr","hr");
    String sql = "SELECT *\r\n"
        + "FROM employees e, DEPARTMENTS d, locations l\r\n"
        + "WHERE d.location_id = l.location_id\r\n"
        + "AND e.DEPARTMENT_ID = d.DEPARTMENT_ID\r\n" + "AND l.city=?";
    PreparedStatement pre = con.prepareStatement(sql);
    pre.setString(1,city);
    ResultSet rs = pre.executeQuery( );
```

```
class EmpDAO { }
```

```
System.out.printf("%s\t/%11s /%9s /%s\t/%9s / %s\n", EmpVO.nameTag[6], EmpVO.nameTag[7],
    EmpVO.nameTag[8], EmpVO.nameTag[0], EmpVO.nameTag[9], EmpVO.nameTag[10]);

while (rs.next( )) {
    EmpVO empvo = new EmpVO( );
    empvo.empId = rs.getInt("EMPLOYEE_ID");
    empvo.firstName = rs.getString("FIRST_NAME");
    empvo.lastName = rs.getString("LAST_NAME");
    empvo.salary = rs.getInt("SALARY");
    empvo.hireDate = rs.getString(1);
    empvo.deptId = rs.getInt("DEPARTMENT_ID");

    list.add(empvo);
    System.out.println("%d\t/%11s /%10s /%d\t/%10s / $ %s\n",
        empvo.empId, empvo.firstName, empvo.lastName, empvo.deptId,
        empvo.hireDate, formatter.format(empvo.salary));
}
rs.close( );
System.out.println();
```

```
class EmpDAO { }
```

---

## 03. 입사년도로 검색하기

```
public void getEmpListByHireDate(int hireYear) throws Exception {  
  
    ArrayList<EmpVO> results = new ArrayList<EmpVO>();  
  
    Class.forName("oracle.jdbc.driver.OracleDriver");  
    Connection con = DriverManager.getConnection(  
        "jdbc:oracle:thin:@localhost:1521:xe", "hr", "hr");  
    String sql = "SELECT substr(e.HIRE_DATE,1),e.*"  
        + "FROM EMPLOYEES e where EXTRACT(year from HIRE_DATE) = ?";  
    PreparedStatement stmt = con.prepareStatement(sql);  
    stmt.setInt(1, hireYear);  
    ResultSet rs = stmt.executeQuery();
```

```
class EmpDAO { }
```

---



```
System.out.printf("%s\t/%11s /%9s /%s\t/%9s / %s\n", EmpVO.nameTag[6], EmpVO.nameTag[7],  
    EmpVO.nameTag[8], EmpVO.nameTag[0], EmpVO.nameTag[9], EmpVO.nameTag[10]);  
while (rs.next()) {  
    EmpVO result = new EmpVO();  
  
    result.empId = rs.getInt("EMPLOYEE_ID");  
    result.firstName = rs.getString("FIRST_NAME");  
    result.lastName = rs.getString("LAST_NAME");  
    result.salary = rs.getInt("SALARY");  
    result.hireDate = rs.getString(1);  
    result.deptId = rs.getInt("DEPARTMENT_ID");  
  
    results.add(result);  
    System.out.printf("%d\t/%11s /%10s /%d\t/%10s / $ %s\n", result.empId,  
        result.firstName, result.lastName, result.deptId, result.hireDate,  
        formatter.format(result.salary));  
}  
System.out.println();
```


## 04. 부서장이름으로 부서명 검색하기

```
public getEmpListByManagerName(String fName, String lName) throws
Exception {
    ArrayList<EmpVO> results = new ArrayList<EmpVO>();

    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con = DriverManager.getConnection(
        "jdbc:oracle:thin:@localhost:1521:xe", "hr", "hr");
    String sql = "select substr(e.HIRE_DATE,1),e.*"
        + "from EMPLOYEES e where MANAGER_ID = "
        + "(select EMPLOYEE_ID from EMPLOYEES"
        + "where FIRST_NAME = ? and LAST_NAME = ?)"
        + "order BY EMPLOYEE_ID";
    PreparedStatement stmt = con.prepareStatement(sql);
    stmt.setString(1, fName);
    stmt.setString(2, lName);
    ResultSet rs = stmt.executeQuery();
```

```
class EmpDAO { }
```

---



```
    int cnt = 0;

    System.out.printf("%s\t/%11s /%9s /%s\t/%9s / %s\n", EmpVO.nameTag[6], EmpVO.nameTag[7],
        EmpVO.nameTag[8], EmpVO.nameTag[0], EmpVO.nameTag[9], EmpVO.nameTag[10]);
    while (rs.next()) {
        EmpVO result = new EmpVO();
        result.empId = rs.getInt("EMPLOYEE_ID");
        result.firstName = rs.getString("FIRST_NAME");
        result.lastName = rs.getString("LAST_NAME");
        result.salary = rs.getInt("SALARY");
        result.hireDate = rs.getString(1);
        result.deptId = rs.getInt("DEPARTMENT_ID");

        results.add(result);
        System.out.printf("%d\t/%11s /%10s /%d\t/%10s / $ %s\n", result.empId,
            result.firstName, result.lastName, result.deptId, result.hireDate,
            formatter.format(result.salary));
    }
    System.out.println();
```




## 05. 부서번호로 검색하기

```
public void getEmpListByDeptId(int deptId) throws Exception {  
  
    ArrayList<EmpVO> list = new ArrayList<EmpVO>( );  
  
    Class.forName("oracle.jdbc.driver.OracleDriver");  
    Connection con = DriverManager.getConnection(  
        "jdbc:oracle:thin:@localhost:1521:xe","hr","hr");  
    String sql = "select employee_id, first_name, last_name,  
        department_id, to_char(salary,$99,999) as money"  
        + "from EMPLOYEES where department_id =?";  
    PreparedStatement stmt = con.prepareStatement(sql);  
    stmt.setInt(1,deptId);  
    ResultSet rs = stmt.executeQuery( );  
}
```

```
class EmpDAO { }
```

---



```
System.out.printf("%s\t/%11s /%9s /%s\t/%9s / %s\n", EmpVO.nameTag[6], EmpVO.nameTag[7],
    EmpVO.nameTag[8], EmpVO.nameTag[0], EmpVO.nameTag[9], EmpVO.nameTag[10]);

while (rs.next( )) {
    EmpVO result = new EmpVO( );
    result.empId = rs.getInt("employee_id");
    result.firstName = rs.getString("first_name");
    result.lastName = rs.getString("last_name");
    result.deptId = rs.getInt("department_id");
    result.salary = rs.getString("money");

    System.out.printf("%d\t/%11s /%10s /%d\t/%10s / $ %s\n", result.empId,
        result.firstName, result.lastName, result.deptId, result.hireDate,
        formatter.format(result.salary));

    list.add(result);
    System.out.println();
}
```

## 06.통계자료 출력하기

```
public void statistics() throws Exception {  
  
    Connection con = null;  
    Class.forName("oracle.jdbc.driver.OracleDriver");  
    String url = "jdbc:oracle:thin:@localhost:49161:xe";  
    String id = "hr";  
    String pw = "hr";  
    con = DriverManager.getConnection(url, id, pw);  
    Scanner scan = new Scanner(System.in);  
    Scanner scan2 = new Scanner(System.in);  
}
```

```
class staticEmp { }
```

---



```
DecimalFormat formatter = new DecimalFormat("###,###");  
  
String sql = null;  
PreparedStatement stmt = null;  
  
int run = 1;  
int deptnum = -1;
```


```
class staticEmp { }
```

---

```
while (true) {
    System.out.printf("1. 최대/최소급여 \t 2. 평균급여 \t 3.부서에 해당하는 인원 \t\n");
    System.out.printf("4.처음 화면이동 \t 5. 프로그램종료 \n");
    System.out.printf("입력 : ");
    int num = scan.nextInt( );

    if(num == 5) {
        System.out.println("프로그램종료!");
        System.exit(0);
    }

    System.out.printf(
        "부서 번호 또는 부서 이름을 검색하세요(모든 정보를 원하면, 0 입력) : ");
    String input_dept = scan2.nextLine( );
    if(isNumber(input_dept)) deptnum = Integer.parseInt(input_dept);
}
```



```
public static boolean isNmber(String s) {  
    try {  
        Integer.parseInt(s);  
        return true;  
    } catch (Exception e) {  
        return false;  
    }  
}
```

## case 1 . 최대급여, 최소급여

```
switch (num) {
    case 1:
        if(deptnum > 0) {
            sql = "SELECT d.department_id, d.department_name, MAX(salary) as max, MIN(salary) as min\n"
                + "FROM employees e, departments d\n"
                + "WHERE e.department_id = d.department_id AND e.department_id = ? \n"
                + "GROUP BY d.department_id, d.department_name\n" + "ORDER BY d.department_id";
            stmt = con.prepareStatement(sql);
            stmt.setInt(1, deptnum);
        }
        else if(deptnum == -1) {
            sql = "SELECT d.department_id, d.department_name, MAX(salary) as max, MIN(salary) as min\n"
                + "FROM employees e, departments d\n"
                + "WHERE e.department_id = d.department_id AND d.department_name = ? \n"
                + "GROUP BY d.department_id, d.department_name\n"
                + "ORDER BY d.department_id";
            stmt = con.prepareStatement(sql);
            stmt.setString(1, input_dept);
        }
}
```

## case 1 . 최대급여, 최소급여

```
else {
    sql = "SELECT d.department_id, d.department_name, MAX(salary) as max, MIN(salary) as min\n"
        + "FROM employees e, departments d\n" + "WHERE e.department_id = d.department_id \n"
        + "GROUP BY d.department_id, d.department_name\n" + "ORDER BY d.department_id";
    stmt = con.prepareStatement(sql);
}

ResultSet rs = stmt.executeQuery( );
System.out.println("부서번호 / 부서이름 / 최대급여 / 최소급여");

while (rs.next( )) {
    System.out.printf("%d / %s / $ %s / $ %s \n", rs.getInt("department_id"),
        rs.getString("department_name"), formatter.format(rs.getInt("max")),
        formatter.format(rs.getInt("min")));
}

deptnum = -1;
break;
```



## case 2 . 평균급여

case 2:

```
if(deptnum > 0) {
    sql = "SELECT d.department_id, d.department_name, ROUND (AVG(salary)) as avg\n"
        + "FROM employees e, departments d\n"
        + "WHERE e.department_id = d.department_id AND e.department_id = ? \n"
        + "GROUP BY d.department_id, d.department_name\n" + "ORDER BY d.department_id";
    stmt = con.prepareStatement(sql);
    stmt.setInt(1, deptnum);
}
else if(deptnum == -1) {
    sql = "SELECT d.department_id, d.department_name, ROUND(AVG(salary)) as avg\n"
        + "FROM employees e, departments d\n"
        + "WHERE e.department_id = d.department_id AND d.department_name = ? \n"
        + "GROUP BY d.department_id, d.department_name\n"
        + "ORDER BY d.department_id";
    stmt = con.prepareStatement(sql);
    stmt.setString(1, input_dept);
}
```

```
else {
    sql = "SELECT d.department_id, d.department_name, ROUND(AVG(salary)) as avg\n"
        + "FROM employees e, departments d\n" + "WHERE e.department_id = d.department_id \n"
        + "GROUP BY d.department_id, d.department_name\n" + "ORDER BY d.department_id";
    stmt = con.prepareStatement(sql);
}

rs = stmt.executeQuery( );
System.out.println("부서번호 / 부서이름 / 평균급여");

while (rs.next( )) {
    System.out.printf("%d / %s / %s \n", rs.getInt("department_id"),
        rs.getString("department_name"), formatter.format(rs.getInt("avg")));
}

deptnum = -1;
break;
```

### case 3 . 부서에 해당하는 인원 수

case 3:

```
if(deptnum > 0) {
    sql = "SELECT d.department_id, d.department_name, count(e.employee_id) as count\n"
        + "FROM employees e, departments d\n"
        + "WHERE e.department_id = d.department_id AND e.department_id = ? \n"
        + "GROUP BY d.department_id, d.department_name\n" + "ORDER BY d.department_id";
    stmt = con.prepareStatement(sql);
    stmt.setInt(1, deptnum);
}
else if(deptnum == -1) {
    sql = "SELECT d.department_id, d.department_name, count(e.employee_id) as count\n"
        + "FROM employees e, departments d\n"
        + "WHERE e.department_id = d.department_id AND d.department_name = ? \n"
        + "GROUP BY d.department_id, d.department_name\n"
        + "ORDER BY d.department_id";
    stmt = con.prepareStatement(sql);
    stmt.setString(1, input_dept);
}
```

### case 3 . 부서에 해당하는 인원 수

---

```
else {
    sql = "SELECT d.department_id, d.department_name, count(e.employee_id) as count\n"
        + "FROM employees e, departments d\n" + "WHERE e.department_id = d.department_id \n"
        + "GROUP BY d.department_id, d.department_name\n" + "ORDER BY d.department_id";
    stmt = con.prepareStatement(sql);
}

rs = stmt.executeQuery( );
System.out.println("부서번호 / 부서이름 / 인원(명)");

while (rs.next( )) {
    System.out.printf("%d / %s / %d명\n", rs.getInt("department_id"),
        rs.getString("department_name"), rs.getInt("count"));
}

deptnum = -1;
break;
```

실행하기

# 소감

## 11. 노영걸

코딩을 처음접하는 교육이어서 열심히 배우고 팀원들 발목잡지 않도록 각오를 했지만 수업 외 복습을 했음에도 불구하고 많은 코드를 실수하고 조장님, 소정님 도움을 너무 많이 받은 것 같습니다.  
어느 정도 알 것 같다는 생각과 수업을 잘 따라가는 느낌이 들었는데 막상 프로젝트를 처음부터 혼자 작성해보니 많은 부족함을 느꼈습니다.  
이와 동시에 팀원들과의 프로젝트 구상, 회의, 수정을 거치며 코딩에 대한 흥미를 더 돋우는 계기가 됐습니다.

## 13. 박수혁

처음으로 팀 프로젝트를 진행하게 되어 처음엔 많이 부족한걸 알기에 팀원들에게 피해가지 않도록 열심히 해 내자라는 생각으로 시작하였는데 막상 프로젝트를 시작하고 나니 직접 조원들에게 이부분 도전해보겠다고 말을 한 뒤 코딩을 시작하였는데 아는 부분을 적용하고 2주간의 교육을 통해 배운 내용을 활용해서 자신감을 갖고 시작을 하다 오류 표시가 나는걸 보게되고 어떤 코드를 사용하면 좋을지 막혀오기 시작하면서 머리 아파오기 시작했지만 조원들에게 도움을 요청하여 어떤 방법으로 해결을 하는게 좋은지 좋은 방법을 알려준 것을 토대로 재시도 하여 작성해보면서 최종적으로 맡은 부분에 대해서 출력화면에 결과 도출된 것을 보고 뿌듯 하였습니다

## 16. 신소정

이 프로젝트를 통해, 자바와 SQL 그리고 JDBC까지 정리할 수 있는 시간이여서 좋았습니다. 또한, 혼자 모든 것을 구현하는 것이 아닌 팀원들과 같이 회의를 하고, 구상하는 과정에 있어서 매우 만족했습니다. 그리고, 모르는 것이 있다고 하면 제가 알고 있는 지식을 통해 가르쳐줌으로써 한번 더 나의 것으로 만들 수 있는 시간이 되었습니다. 마지막으로 다들 프로젝트를 하면서 많은 의견도 내주시고, 코딩에 대한 열정적인 모습을 보며 저 또한 더 열심히 코딩을 하여 내것으로 만들고, 알려줄 수 것이라면 더 잘가르쳐줘야겠다는 생각을 했습니다. 다음프로젝트일 때도, 좋은 팀원들을 만나 좋은 프로젝트를 완성하면 좋겠습니다.

## 26. 정은빈

2주간 혼자 하던 코딩을 팀원분들과 함께 하며 많이 부족하고 죄송했습니다. 그러나 저를 도와주시고 질문에 항상 저의 눈높이에 맞춰 설명해주신 덕분에 더 발전할 수 있었습니다.  
다음 2주 후 혹은 다른 팀프로젝트를 진행한다면 팀에 도움이 될 수 있을만큼 성장하겠습니다.  
감사합니다.

## 28. 최연호

2주간의 강의와 프로젝트를 통해 자바에 대하여 특히 클래스와 변수에 대하여 다시 정리하는 시간이 되었으며 또한 다른 팀원들에게 제가 알고있는 것들을 알려주면서 제가 아직 부족하다는것을 다시 알게 되었습니다. 부족한 부분은 다시 공부하고 알고있던부분은 확실하게 정리하여 온전히 저의 지식으로 만들기위해 노력할 것이며 차후에 제가 아직 못배운 부분이 있을때 이를위한 배경지식이 되어 더빠르게 그리고 바르게 습득하기 위해 열심히 하겠습니다.

감사합니다