

예외처리 예시

202.01.18

예시 기본 설계

```
> org.java.connect
> org.java.controller
> org.java.dao
> org.java.dto
> org.java.exceptionEx
> org.java.service
```

VC Model2 기본 패키지 구조 -> 프레임워크 -> Spring boot

1. 패키지 생성
2. 각패키지 별 클래스 생성
3. 케넥션 -> DB연동 객체 생성 static -> 공유

org.java.controller	-> 컨트롤러	-> 일 시키고 결과를 사용자에게 반환(View->브라우저표시)
MemberController		
org.java.service	-> 서비스	-> 컨트롤러에서 시킨일 수행, DB관련일은 DAO 처리-><-결과반환
MemberService	-> 인터페이스	void excuteQueryService() throws SQLException,IOException;
MemberInsertService	-> 회원가입	
MemberSelectService	-> 회원조회	
MemberUpdateService	-> 회원수정	
MemberDeleteService	-> 회원탈퇴	
org.java.dao	-> DAO	-> 서비스에서DB관련 시킨일 수행 DAO-><-DB
MemberDao		insert() update() delete() select()
org.java.dto	-> DTO	-> DB의 데이터(테이블)과 연관(매핑)있는 자바 클래스
MemberDto		-> getters, setters, NoArgsContstruct, AllArgsContstruct, toString() **필드 -> private int no; // 회원번호 String userId; //아이디 String userPw; //비밀번호 String name; //이름 int age; //나이
org.java.connect	-> DB연동	-> DB관련 일을 처리 할 때 기본적으로 연동 되는 클래스
DBConnect		

Controller

```
public class MemberController {  
    public static void main(String[] args) throws SQLException, IOException {  
        System.out.println("Controller");  
  
        // SQL(관계형데이터베이스 CRUD언어) -> RDBMS(관계형데이터베이스): oracle,mysql  
        // insert(C), select(R),update(U),delete(D)  
  
        Scanner input = new Scanner(System.in);  
  
        boolean bool = true;  
  
        while (bool) {  
            System.out.println("쿼리문 입력( insert, select,update,delete)");  
            String query = input.next();  
  
            MemberService service = null; // 객체형 기본값 null  
  
            if (query.equals("insert")) {  
                service = new MemberInsertService();  
                service.excuteQueryService();  
            } else if (query.equals("update")) {  
                service = new MemberUpateService();  
                service.excuteQueryService();  
            } else if (query.equals("select")) {  
                service = new MemberSelectService();  
                service.excuteQueryService();  
            } else if (query.equals("delete")) {  
                service = new MemberDeleteService();  
                service.excuteQueryService();  
            } else if (query.equals("exit")) {  
                System.out.println("종료");  
                bool = false;  
            } else {  
                System.out.println("쿼리문 입력 오류");  
                throw new IllegalArgumentException("쿼리문 입력 오류");  
            }  
        }  
    }  
}
```

Service

org.java.service

- > MemberDeleteService.java
- > MemberInsertService.java
- > MemberSelectService.java
- > MemberService.java
- > MemberUdateService.java
- > MemberUpdate.java

```
public interface MemberService {  
  
    void excuteQueryService() throws SQLException, IOException;  
  
}
```

```
public class MemberInsertService implements MemberService {  
  
    @Override  
    public void excuteQueryService() throws SQLException, IOException {  
        System.out.println("회원가입");  
  
        MemberDao dao = new MemberDao();  
        int result = dao.insert();  
  
        if (result != 1) {  
            System.out.println("회원가입 Fail!");  
        } else {  
            System.out.println("회원가입 Success!");  
        }  
    }  
  
}
```

```
public class MemberUdateService implements MemberService {  
  
    @Override  
    public void excuteQueryService() throws SQLException, IOException {  
        System.out.println("회원수정");  
  
        MemberDao dao = new MemberDao();  
        int result = dao.update();  
  
        if (result != 1) {  
            System.out.println("회원수정 Fail!");  
        } else {  
            System.out.println("회원수정 Success!");  
        }  
    }  
  
}
```

```
public class MemberDeleteService implements MemberService {  
  
    @Override  
    public void excuteQueryService() throws SQLException, IOException {  
        System.out.println("회원탈퇴");  
  
        MemberDao dao = new MemberDao();  
        int result = dao.delete();  
  
        if (result != 1) {  
            System.out.println("회원탈퇴 Fail!");  
        } else {  
            System.out.println("회원탈퇴 Success!");  
        }  
    }  
  
}
```

```
public class MemberSelectService implements MemberService {  
  
    @Override  
    public void excuteQueryService() throws SQLException, IOException {  
        System.out.println("회원조회");  
  
        MemberDao dao = new MemberDao();  
        // 회원 조회 -> DB 테이블 조회 get  
        MemberDto member = dao.select();  
  
        if (member != null) {  
            System.out.println("조회된 회원이 있습니다.");  
        } else {  
            System.out.println("조회된 회원이 없습니다.");  
        }  
    }  
  
}
```

```
public class MemberDao {  
  
    public MemberDao() {  
        System.out.println("DAO");  
    }  
  
    // 회원가입 -> 성공시 1반환, 실패하면 나머지 값 반환  
    public int insert() {..  
  
    // 회원수정  
    public int update() {..  
  
    // 회원탈퇴  
    public int delete() {..  
  
    // 회원조회 -> 성공시 객체반환, 실패하면 null or empty  
    public MemberDto select() {..  
  
}
```

```
// DB의 데이터(테이블)과 연관(매핑)있는 자바 클래스
public class MemberDto {

    private int no; // Long -> 객체형 등록번호(JPA)
    private String userId;
    private String userPw;
    private String name;
    private int age;
    // -> Spring boot Project
    // NoArgsConstructor
    public MemberDto() {}

    // AllArgsConstructor
    public MemberDto(int no, String userId, String userPw, String name, int age) {

    //setters, getters
    public int getNo() {}

    public void setNo(int no) {}

    public String getUserId() {}

    public void setUserId(String userId) {}

    public String getUserPw() {}

    public void setUserPw(String userPw) {}

    public String getName() {}

    public void setName(String name) {}

    public int getAge() {}

    public void setAge(int age) {}

    // toString()
    public String toString() {}
}
```

connect

```
//DB관련 일을 처리 할 때 기본적으로 연동 되는 클래스
public class DBConnect {

    // 공유 -> Connection DB연동 객체
    public static Connection getConnection() {

        Connection conn = null; // 커넥션 객체 기본 null

        String url = "jdbc:oracle:thin:@localhost:1521:xe"; // DB URL
        String user = "system"; // 아이디
        String password = "1234"; // 비밀번호
        String driver = "oracle.jdbc.driver.OracleDriver";

        try {
            // 1. 드라이버찾기
            Class.forName(driver);
            System.out.println("드라이버 OK");
            // DB연동 -> 커넥션 객체 설정
            conn = DriverManager.getConnection(url, user, password);
            System.out.println("DB연동 Success!!");
        } catch (ClassNotFoundException e) {
            System.out.println("드라이버 NO");
            e.printStackTrace();
        } catch (SQLException e) {
            System.out.println("DB연동 Fail!!");
            e.printStackTrace();
        }
        // conn -> null -> DB연동 실패
        return conn;
    }
}
```