

Webプログラミング実習Ⅱ 授業用レジュメ②

■phpMyAdminで入力するクエリ

- MAMP起動→Open StartPage→Tools→PHPMyAdminからデータベース「nejiko_online」を作成
 - ・使用する文字コードをutf-8に
 - ・*は開講日付を入れること

```
CREATE DATABASE nejiko_online_***** CHARACTER SET utf8;
```

- 画面右のメニューでnejiko_onlineを選択し、「SQL」タブからテーブル「ranking」を作成
 - ・name VARCHAR(20) NOT NULL
 - ・score INT NOT NULL

```
CREATE TABLE ranking(name VARCHAR(20) NOT NULL, score INT NOT NULL);
```

- 画面右のメニューでnejiko_online→rankingを選択し、「SQL」タブからUnity上での動作確認用レコードを作成

```
INSERT INTO ranking(name,score) VALUES ('TOFU',100),('MOFUROID',50),('RYOUSAN',30);
```

■Eclipseで入力するコード

■動的Webプロジェクトの作成

- プロジェクト名：「nejikoRunOnlineServer」

■プロジェクトへの各種ライブラリの導入

以下、そのプロジェクトの「src」パッケージ内へ各クラスを作っていきます。

■Entity(『model』パッケージに作成)

【NejikoMileage】：ユーザー名・スコアのひと組を保持するためのクラス

```
package model;
import java.io.Serializable;

public class NejikoMileage implements Serializable {

    private String name;
    private int score;

    //コンストラクタ
    public NejikoMileage(String name,int score){
        this.name = name;
        this.score = score;
    }
    public NejikoMileage(){

    }
    //ゲッター・セッター
    public String getName(){
        return name;
    }

    public void setName(String name){
        this.name = name;
    }
}
```

```

    }

    public int getScore(){
        return score;
    }

    public void setScore(int score){
        this.score = score;
    }
}

```

■DAOクラス（『dao』パッケージに作成）

【NejikoOnlineDAO】

```

package dao;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import model.NejikoMileage;

public class NejikoOnlineDAO {

    //フィールド
    //接続するデータベースのアドレス(*は開講日付)

```

```

        private final String DB_PATH = "jdbc:mysql://localhost:3306/nejiko_online_*****";

        //送信するクエリの文字コード設定(データベースのアドレスに付け足される)

        private final String DB_CHARACTER_ENCODE = "?useUnicode=true&characterEncoding=utf8";

        //データベースのユーザー名
        private final String DB_USERNAME = "root";

        //データベースのパスワード
        private final String DB_PASSWORD = "root";

        //コンストラクタでJDBCドライバを読み込む
        public NejikoOnlineDAO(){
            try{

                Class.forName("com.mysql.jdbc.Driver");

            }catch (ClassNotFoundException e) {
                e.printStackTrace();
            }
        }

        public List<NejikoMileage> findNejikoMileage()
        {

            //変数
            //取得結果を格納する配列
            List<NejikoMileage> result = new ArrayList<NejikoMileage>();

            //DAOのお約束①: try-with-resources文の中でDBへ接続
            try(Connection conn = DriverManager.getConnection(DB_PATH + DB_CHARACTER_ENCODE,DB_USERNAME,DB_PASSWORD);)

```

```

{

    //String型変数にクエリを用意
    //『SELECT name,score FROM ranking ORD
ER BY score DESC LIMIT 3;』
    //※「rankingテーブルに保存されたレコードのna
meとscoreをscoreの多い順に3つ取得」
    String sql = "";
    sql += "SELECT";
    sql += " name,score";
    sql += " FROM";
    sql += " ranking";
    sql += " ORDER BY";
    sql += " score DESC LIMIT 3;";

    //Connection.prepareStatement(用意したク
エリ)メソッドを使って、
    //用意したクエリを実行するためのPreparedStatement
ement型インスタンスを作成
    PreparedStatement pstmt = conn.prepare
Statement(sql);

    //『executeQuery』メソッドでクエリを実行、
    //ResultSet型変数に結果を代入
    ResultSet rs = pstmt.executeQuery();

    //『rs.next()』で条件分岐や繰り返しを行い、
    //ResultSetからひとつずつ情報を引き出す
    while(rs.next())
    {

```

```

        String name = rs.getString("name");
        int score = rs.getInt("score");
        NejikoMileage record = new NejikoMileage(name, score);
        result.add(record);
    }

    //returnで取得結果を返す
    return result;

}

//DAOのお約束②：『SQLException』のcatch文を記述
}catch(SQLException e){
    e.printStackTrace();
}

//何かしらの例外が起きたらnullが返ってくる
return null;
}

public boolean registNejikoMileage(NejikoMileage mileage)
{
    //変数
    //引数で渡したクラスからユーザー名と得点を取得
    String name = mileage.getName();
    int score = mileage.getScore();

    //DAOのお約束①：try-with-resources文の中でDBへ接続
    try(Connection conn = DriverManager.getConnection(DB_PATH + DB_CHARACTER_ENCODE, DB_USERNAME, DB_PASSWORD);)
    {

```

```

//String型変数にクエリを用意
//『INSERT INTO ranking(name,score) VA
LUES (?,?);』

//※「rankingテーブルへ(name,score)のレコ
ードを挿入」

String sql = "";
sql += "INSERT";
sql += " INTO";
sql += " ranking (name,score)";
sql += " VALUES";
sql += " (?,?);";

//↑上記クエリの中の「?」は「プレースホルダ」°
//PreparedStatement.setxx(文頭から数えた?の位置,入れた
い値)

//系メソッドによってクエリの文中に変数を代入できる機能

//Connection.prepareStatement(用意したク
エリ)メソッドを使って、
//用意したクエリを実行するためのPreparedStat
ement型インスタンスを作成

PreparedStatement pstmt = conn.prepare
Statement(sql);

//プレースホルダにnameとscoreを入れてやる
pstmt.setString(1,name);
pstmt.setInt(2,score);

System.out.println("NejikoDAO/DBへ送信
するSQL:" + pstmt.toString());

```

は

```
//DBの内容を書き換えるようなクエリを実行する際
```

```
//『executeUpdate』メソッドを使う
```

```
pStmt.executeUpdate();
```

```
//正常に取得できたらtrueを返す
```

```
return true;
```

```
//DAOのお約束②：『SQLException』のcatch文を記述
```

```
}catch(SQLException e){
```

```
    e.printStackTrace();
```

```
}
```

```
//何かしらの例外が起きたらfalseを返す
```

```
return false;
```

```
}
```

```
}
```

※ここまで完成したら、下のプログラムを「Javaアプリケーション」で実行して動作を確認してください

■動作確認用プログラム1(『test』パッケージに作成)

【NejikoOnlineDAOTest】：DAOクラス単体の動作テスト用プログラム

```
package test;
```

```
import java.util.List;
```

```
import dao.NejikoOnlineDAO;
```

```
import model.NejikoMileage;
```

```
public class NejikoOnlineDAOTest {
```

```
    public static void main(String[] args){
```



```
        findNejikoMileageTest();
        resistNejikoMileageTest();
    }

    public static void findNejikoMileageTest()
    {
        //DAOをインスタンス化
        NejikoOnlineDAO dao = new NejikoOnlineDAO();

        //DBからスコアランキングを取得
        List<NejikoMileage> ranking = dao.findNejikoMileage();

        //取得したスコアランキングをコンソールへ出力
        for(NejikoMileage score : ranking){
            System.out.println("Name : " + score.getName());
            System.out.println("Score : " + score.getScore());
        }
    }

    public static void resistNejikoMileageTest()
    {
        //動作テスト用のダミー記録を作成
        NejikoMileage testScore = new NejikoMileage("テスト",10);

        //DAOをインスタンス化し `ダミー記録をDBへ書き込み
        NejikoOnlineDAO dao = new NejikoOnlineDAO();
```

```

        boolean result = dao.registNejikoMileage(testS
core);

        //結果に応じてコンソールへ文字列を出力
        if(result)
        {
            System.out.println("書き込み成功");
        }else{
            System.out.println("書き込み失敗");
        }
    }
}

```

■BO（『model』パッケージに作成）

【NejikoLogic】：DAOクラスのインスタンスを作り、処理に応じたメソッドを呼び出すクラス

なお、JSONデータ読み込みのために必ず下記ファイルをlibフォルダに入れてください。

- jackson-core-2.9.7
- jackson-databind-2.9.7.jar
- jackson-annotations-2.9.7.jar

```

package model;

import java.util.List;

import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.ObjectMapper;

import dao.NejikoOnlineDAO;

```

```

public class NejikoLogic {

    //daoオブジェクトを作っておく
    NejikoOnlineDAO dao = new NejikoOnlineDAO();

    public String findMileage() throws JsonProcessingException
    {
        //DAOのメソッドを使ってDBにアクセス `結果を変数`に取る
        List<NejikoMileage> mileage = dao.findNejikoMileage();

        //ObjectMapper型インスタンスを作成
        ObjectMapper mapper = new ObjectMapper();

        //ObjectMapperのメソッドを使って取得結果をJSON文字列に変換。
        //変換後のJSON文字列を戻り値にする
        return mapper.writeValueAsString(mileage);
    }

    public boolean registMileage(NejikoMileage record)
    {
        //DAOのメソッドを使ってDBに書き込み
        return dao.registNejikoMileage(record);
    }
}

```

※ここまで完成したら、下のプログラムを「Javaアプリケーション」で実行して動作を確認してください

■動作確認用プログラム2(『test』パッケージに作成)

【NejikoLogicTest】：BOクラスの動作テスト用プログラム

```
package test;

import com.fasterxml.jackson.core.JsonProcessingException;

import model.NejikoLogic;
import model.NejikoMileage;

public class NejikoLogicTest {
    public static void main(String[] args){
        findMileageTest();
        registMileageTest();
    }

    public static void findMileageTest()
    {
        //BOをインスタンス化
        NejikoLogic nejikoLogic = new NejikoLogic();

        try{
            //BOからJSON文字列を取得
            String ranking = nejikoLogic.findMileage();

            //取得したJSON文字列をコンソールへ出力
            System.out.println(ranking);
        }catch(JsonProcessingException e){
            e.printStackTrace();
        }
    }
}
```

```

    }

    public static void registMileageTest()
    {
        //動作テスト用のダミー記録を作成
        NejikoMileage testScore = new NejikoMileage
("テスト",10);

        //BOをインスタンス化し `ダミー記録`をDBへ書き込み
        NejikoLogic logic = new NejikoLogic();
        boolean result = logic.registMileage(testScore);

        //結果に応じてコンソールへ文字列を出力
        if(result)
        {
            System.out.println("書き込み成功");
        }else{
            System.out.println("書き込み失敗");
        }
    }
}

```

■サーブレットクラス

(プロジェクト・エクスプローラーで右クリック→「新規」→「サーブレット」
「servlet」パッケージに作成)

【NejikoMileageServlet】：ランキングの情報を送る側

```

package servlet;

import java.io.IOException;

```

```
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import model.NejikoLogic;
import model.NejikoMileage;

@WebServlet("/NejikoMileageServlet")
public class NejikoMileageServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
    {

        //BOのインスタンスを作成
        NejikoLogic nejikoLogic = new NejikoLogic();

        //BOからランキングの情報が入ったJSON文字列を取得
        String responseJSON = nejikoLogic.findMileage
        ();

        System.out.println("doGet/レスポンスするJSON文字列
        ..." + responseJSON);

        //レスポンスのファイル形式 `文字コードを指定
        response.setContentType("application/json");
    }
}
```

```

        response.setCharacterEncoding("utf-8");

        //レスポンスにJSON文字列を書き込み、クライアントへ返す
        response.getWriter().print(responseJSON);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
    {

        //返すレスポンスの文字コードをUTF-8に揃える
        response.setCharacterEncoding("utf-8");

        //送られてきたリクエストの文字コードをUTF-8に揃える
        request.setCharacterEncoding("utf-8");

        //送られてきた走行記録を取得
        String name = request.getParameter("name");

        //リクエストパラメータはString型でしか取得できないので、
        //走行距離はint型にパースしてやる
        int score = Integer.parseInt(request.getParameter("score"));

        System.out.println("doPost/Postされたパラメータ ...
name : " + name + "/score : " + score);

        //取得した走行記録をJavaのクラスに変換
        NejikoMileage nm = new NejikoMileage(name, score);
    }
}

```

```

        System.out.println("doPost/リクエストパラメータの値
が入ったインスタンス:" + nm);

        //B0のインスタンス作成
        NejikoLogic nejikoLogic = new NejikoLogic();

        //B0のメソッドを使い走行記録をDBに書き込み。
        //結果の成否がboolean型で返ってくるのでif文で成否を表示
        if(nejikoLogic.registMileage(nm))

        {
            //書き込み成功したら
            response.getWriter().print("SUCCESS");
        }else{
            //書き込み失敗したら
            response.getWriter().print("ERROR");
        }
    }
}

```

入力完了後、以下のURLをブラウザに入力してJSON形式の文字列が
ブラウザ上で表示されることを確認すること。

```

http://localhost:8080/nejikoRunOnlineServer/NejikoMileageServlet

```