Webシステム基礎実験レポート

(ポケモン出現db) 2132023 内山裕太

1. 概要

今回作成したポケモン出現dbの使い方について説明する．

(URL…http://localhost:8080/s2132023/AppearServlet)

URLをクリックすると使用可能状態となる．初期画面の表示内容について説明する．

* ポケモン出現DB

→画面をリロードする．

* 並び替え

→選択された条件で，登録されているポケモンを並び替

えたうえで表示する．

* 登録

→pokemon.csvに登録されているポケモンの番号を指定

することで登録できる．市コード，時間を入力するこ

とで詳細な情報まで登録できる．下部の「全国の市コ

ード一覧」をクリックすることで確認できる．

* 削除

→出現情報に表示されているポケモンの中から，IDを

指定することで，任意のポケモン情報を削除すること

ができる．

* 出現情報

→現状登録されているポケモン出現情報を表示する．

通常の使用方法については上記のとおりである．次に，動作について説明する．

1. 動作

アプリケーションの内部処理について説明する．

1. 初めは，appear.jspによって作成されるページが表示される．画面上の「並び替え」「登録」「削除」のいずれかをクリックすることで，処理が異なる．それぞれのボタンが押された際の処理について以下に記述する．

以下，appear.jsp

<%@ page language="java" contentType="text/html; charset=UTF-8"

 pageEncoding="UTF-8"%>

<%@ page import="db.Appear, java.util.List" %>

<%

List<Appear> appearList = (List<Appear>)request.getAttribute("list");

%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>ポケモン出現DB</title>

</head>

<body>

<h1> <a href="AppearServlet">ポケモン出現DB</a></h1>

<hr>

<form action="AppearServlet" method="POST">

<input type="radio" name="item" value="ID" checked="checked">ID

<input type="radio" name="item" value="番号">番号

<input type="radio" name="item" value="名前">名前

<br>

<input type="radio" name="order" value="asc" checked="checked">昇順

<input type="radio" name="order" value="desc">降順

<br>

<input type="submit" name="submit" value="並び替え">

<hr>

番号<input type="text" name="newnumber">

市コード<input type="text" name="newshicode">

<br>

時間の指定を行わない場合は，以下の入力不要

<br>

年<input type="text" name="newyear">

月<input type="text" name="newmonth">

日<input type="text" name="newdate">

<br>

時間<input type="text" name="newhour">

分<input type="text" name="newminute">

秒<input type="text" name="newsecond">

<input type="submit" name="submit" value="登録">

<br>

<a href="http://www13.plala.or.jp/bigdata/municipal\_code\_1.html" target="\_blank">全国の市コード一覧</a>

<hr>

ID<input type="text" name="deleteid">

<input type="submit" name="submit" value="削除">

</form>

<hr>

<a href="AppearServlet?shimei=習志野市">習志野市</a>

<a href="AppearServlet?shimei=船橋市">船橋市</a>

<hr>

<% if (appearList != null) { %>

出現情報

<table border="1">

<tr><th>ID</th><th>番号</th><th>名前</th><th>タイプ</th><th>県名</th><th>市名</th><th>日付</th><th>時刻</th></tr>

<% for (Appear appear: appearList) { %>

<tr>

<td><%=appear.getId() %></td>

<td><%=appear.getNumber() %></td>

<td><%=appear.getName() %></td>

<td><%=appear.getType() %></td>

<td><%=appear.getKen() %></td>

<td><%=appear.getShi() %></td>

<td><%=appear.getDate() %></td>

<td><%=appear.getTime() %></td>

</tr>

<% } %>

</table>

<% } %>

</body>

</html>

1. 並び替え

最初に，AppearServletクラス(以下Servletクラス)のdoGetメソッドが実行される．このメソッドでは，ボタンが押された際に，入力されている情報を取得する．

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

        request.setCharacterEncoding("UTF-8");

        String item = request.getParameter("item");

        String order = request.getParameter("order");

        String submit = request.getParameter("submit");

        String newnumber = request.getParameter("newnumber");

        String newshicode = request.getParameter("newshicode");

        String deleteid = request.getParameter("deleteid");

        String newyear = request.getParameter("newyear");

        String newmonth = request.getParameter("newmonth");

        String newdate = request.getParameter("newdate");

        String newhour = request.getParameter("newhour");

        String newminute = request.getParameter("newminute");

        String newsecond = request.getParameter("newsecond");

        String shimei = request.getParameter("shimei");

        System.out.printf("\n%s:%s:%s:\n",item,order,submit);

        System.out.printf("%s:%s:\n",newnumber,newshicode);

        System.out.printf("%s:%s:\n",deleteid,shimei);

並び替えボタンが押された際は，Servletクラス内のselectAllメソッドが実行される．selectAllメソッドではAppearDAO(以下DAOクラス)インスタンスを生成する．コンストラクタが実行され，DAOクラスのfindAllメソッドが実行される．なお，findAllメソッドに渡す値として，item,orderがある．itemは「ID,番号,名前」の中から選択されたもの，orderは

「昇順,降順」の中から選択されたものである．

void selectAll(HttpServletRequest request,HttpServletResponse response,String item,String order)throws ServletException{

        AppearDAO appearDAO = new AppearDAO();

        List<Appear> list = appearDAO.findAll(item,order);

        request.setAttribute("list",list);

    }

public List<Appear> findAll(String item,String order){

        List<Appear> list = new ArrayList<>();

        String url = "jdbc:h2:tcp://localhost/./s2132023";

        Connection conn = null;

        try {

            conn = DriverManager.getConnection(url,"user","pass");

            String sql = "SELECT ID,APPEAR.番号,POKEMON.名前,TYPE.タイプ,ken.県名,shi.市名,日付,時刻 FROM APPEAR LEFT JOIN POKEMON ON APPEAR.番号 = POKEMON.番号 LEFT JOIN SHI ON APPEAR.市コード = SHI.市コード LEFT JOIN KEN ON SHI.県コード = KEN.県コード LEFT JOIN TYPE ON APPEAR.番号 = TYPE.番号 ORDER BY " + item + " " + order;

            PreparedStatement pre = conn.prepareStatement(sql);

            ResultSet rs = pre.executeQuery();

            while(rs.next()) {

                int id = rs.getInt("ID");

                int number = rs.getInt("番号");

                String name = rs.getString("名前");

                String type = rs.getString("タイプ");

                String ken = rs.getString("県名");

                String shi = rs.getString("市名");

                Date date = rs.getDate("日付");

                Time time = rs.getTime("時刻");

                Appear a = new Appear(id, number, name, type, ken, shi, date, time);

                list.add(a);

            }

        }catch(SQLException e) {

            e.printStackTrace();

        }finally {

            if(conn != null) {

                try {

                    conn.close();

                }catch(SQLException e) {

                    e.printStackTrace();

                    return null;

                }

            }

        }

        return list;

    }

1. 登録

最初に，ServletクラスのdoGetメソッドが実行される．このメソッドでは，ボタンが押された際に，入力されている情報を取得する．登録ボタンが押された際は，Servletクラス内のinsertメソッドが実行される．insertメソッドではDAOクラスのインスタンスを生成する．コンストラクタが実行され，DAOクラスのinsertメソッドが実行される．

void insert(String newnumber,String newshicode) {

        AppearDAO appearDAO = new AppearDAO();

        try {

            int num = Integer.parseInt(newnumber);

            int code = Integer.parseInt(newshicode);

            appearDAO.insert(num,code);

        }catch(NumberFormatException e) {

            System.out.println("不正な番号または市コードが入力されました"+e.getMessage());

        }

    }

void insert(String newnumber,String newshicode,String newyear,String newmonth,String newdate,String newhour,String newminute,String newsecond) {

        AppearDAO appearDAO = new AppearDAO();

        try {

            int num = Integer.parseInt(newnumber);

            int code = Integer.parseInt(newshicode);

            int year = Integer.parseInt(newyear);

            int month = Integer.parseInt(newmonth);

            int date = Integer.parseInt(newdate);

            int hour = Integer.parseInt(newhour);

            int minute = Integer.parseInt(newminute);

            int second = Integer.parseInt(newsecond);

            appearDAO.insert(num,code,year,month,date,hour,minute,second);

        }catch(NumberFormatException e) {

            System.out.println("不正な番号または市コードが入力されました"+e.getMessage());

        }

    }

また，insertメソッドは2種類存在し，入力された情報に応じて使い分けている．

* + 引数が２つのinsertメソッド

public boolean insert(int number, int shicode) {

        String url = "jdbc:h2:tcp://localhost/./s2132023";

        Connection conn = null;

        try {

            conn = DriverManager.getConnection(url,"user","pass");

            String sql = "INSERT INTO appear(番号,市コード) VALUES(?,?)";

            PreparedStatement pre = conn.prepareStatement(sql);

            pre.setInt(1, number);

            pre.setInt(2, shicode);

            int result = pre.executeUpdate();

            if(result == 1) return true;

        } catch (SQLException e) {

            e.printStackTrace();

        }finally {

            if(conn != null) {

                try {

                    conn.close();

                }catch(SQLException e) {

                    e.printStackTrace();

                }

            }

        }

        return false;

    }

上記のメソッドは，ポケモンの番号と市コードの

みを用いて登録するメソッドである．おもに登録

した時刻を指定しない場合に使用する．

* + 引数が８つのinsertメソッド

public boolean insert(int number, int shicode, int year, int month, int day, int hour, int minute, int second) {

        String url = "jdbc:h2:tcp://localhost/./s2132023";

        Connection conn = null;

        try {

            conn = DriverManager.getConnection(url,"user","pass");

            String sql = "INSERT INTO appear(番号,市コード,日付,時刻) VALUES(?,?,?,?)";

            PreparedStatement pre = conn.prepareStatement(sql);

            String datestr = String.format("%4d-%02d-%02d", year, month, day);

            Date date = Date.valueOf(datestr);

            String timestr = String.format("%02d:%02d:%02d", hour, minute, second);

            Time time = Time.valueOf(timestr);

            pre.setInt(1, number);

            pre.setInt(2, shicode);

            pre.setDate(3, date);

            pre.setTime(4, time);

            int result = pre.executeUpdate();

            if(result == 1)return true;

        }catch(SQLException e) {

            e.printStackTrace();

        }finally {

            if(conn != null) {

                try {

                    conn.close();

                }catch(SQLException e) {

                    e.printStackTrace();

                }

            }

        }

        return false;

    }

上記のメソッドは，ポケモンの番号と市コードに

加え，日付と時刻まで登録するメソッドである．

日付と時刻の情報が与えられた場合に使用され

る．

1. 削除

最初に，ServletクラスのdoGetメソッドが実行される．このメソッドでは，ボタンが押された際に，入力されている情報を取得する．登録ボタンが押された際は，Servletクラス内のdeleteメソッドが実行される．deleteメソッドではDAOクラスのインスタンスを生成する．コンストラクタが実行され，DAOクラスのdeleteメソッドが実行される．

void delete(String deleteid) {

        AppearDAO appearDAO = new AppearDAO();

         try {

             int id = Integer.parseInt(deleteid);

             appearDAO.delete(id);

         } catch (NumberFormatException e) {

             System.out.println("不正なIDが入力されました"+e.getMessage());

         }

    }

public boolean delete(int id) {

        String url = "jdbc:h2:tcp://localhost/./s2132023";

        Connection conn = null;

        try {

            conn = DriverManager.getConnection(url,"user","pass");

            String sql = "DELETE FROM appear WHERE ID = ?";

            PreparedStatement pre = conn.prepareStatement(sql);

            pre.setInt(1, id);

            int result = pre.executeUpdate();

            if(result == 1)return true;

        }catch(SQLException e) {

            e.printStackTrace();

        }finally {

            if(conn != null) {

                try {

                    conn.close();

                }catch(SQLException e) {

                    e.printStackTrace();

                }

            }

        }

        return false;

    }

1. アピールポイント

今回のプログラム作成において，自己流で追加した動作について記載する．

* 日付，時刻の登録

ポケモン登録を行う際に，日付と時刻を入力した場合のみ日付と時刻を登録内容に追加できるようにした．入力されなかった場合は，登録を行った日時で登録される仕様となっている．

* 全国の市コード確認

登録情報入力欄の下部に「全国の市コード一覧」を確認できるURLを配置している．これにより，任意の市コードをその場で確認することが可能となっている．

* 出現情報追加

出現情報表示欄に表示される内容に「タイプ」を追加している．タイプを２つ以上持つポケモンに関しては，タイプの数だけ行数が増えた状態で表示される．

1. 感想

今回の「ポケモン出現db」の作成に関して，多くの場面で「難しい」と感じた．javaファイルとhtmlファイルの結び付け，jspファイルの記述方法，新たな知識を多く学んだ．現状，卒業研究でもシステムの開発を行う予定であるため，夏季の長期休業を用いて再度修得に励もうと思う．