

```
<?php
```

```
/******
```

```
 * 色の設定
```

```
 */
```

```
define("MISS_ANS_BACKGROUND", "salmon"); //間違った解答の背景色を設定する
```

```
define("MISS_ANS_MISSTITLE", "darkred"); //間違った解答の、間違いのタイトル背景色を設定する
```

```
define("MISS_ANS_MISSTABLE1", "pink"); //間違った解答の、間違いの位置と訂正例の背景色を設定する
```

```
define("MISS_ANS_MISSTABLE2", "white"); //間違った解答の、間違いの種類の背景色を設定する
```

```
define("CORRECT_ANS_BACKGROUND", "skyblue"); //正解の背景色を設定する
```

```
/******
```

```
 * 添削済み解答結果を5件ごとに表示するボタンを作成する
```

```
 */
```

```
function make_buttons_to_view_five_answers($anslog_list) {
```

```
    $size_of_anslog_list = count($anslog_list);
```

```
    echo "<form action='' method='POST'>";
```

```
    $i = 0;
```

```
    while ($i < $size_of_anslog_list) {
```

```
        $start_point = $i +1;
```

```
        echo "<input type='submit' name='start_point' value='". $start_point. "～".
```

```
($start_point +4). "'>";
```

```
        $i = $i +5;
```

```
    }
```

```
    echo " </form> ";
```

```
}
```

```
/******
```

```
 * 全解答結果リストから、
```

```
 * 与えられた解答番号範囲の解答を表示する
```

```
 */
```

```
function print_anslog_list_by_answer_number($anslog_list, $start_point, $length) {
```

```
    $start_point = $start_point -1;
```

```

        $anslog_list_to_print = array_slice($anslog_list, $start_point, $length);
        print_anslog_list($anslog_list_to_print);
    }

    /*****
    * 添削結果一覧を表示する
    */
    function print_anslog_list($anslog_list) {
        // -注釈 -
        //php ではグローバル変数は関数内で宣言しなければならないため、ここでも宣言
        //http://php.s3.to/man/language.variables.scope.html
        foreach ($anslog_list as $key => $value) {
            if (answer_is_corrected($value) == true) {
                $anslog_table_bgcolor = select_table_background($value);
                echo          "<table          width='75%'          border=1
bgcolor='$anslog_table_bgcolor'><tr><td>問題：";
                echo "$value >problem";
                echo "</td></tr>\n";
                echo "<tr><td>あなたの解答：";
                echo "$value >answer";
                echo "<tr><td>\n";
                if (answer_is_correct($value) == false) {
                    foreach ($value >mistakes as $misskey => $miss) {
                        echo          "<table          width='100%'><tr><td
bgcolor='darkred' colspan=3><font color='white'>あなたの解答：";
                        echo
make_markedproblem_from_problem_and_number_list($value >answer, $miss >location);
                        echo "</font></tr></td>\n";

                        echo "<tr><td bgcolor='pink'>";
                        echo "$miss >location";
                        echo "</td>\n";

                        echo "<td bgcolor='white'>";
                        echo "$miss >type";
                        echo "：$miss >correction<br>";

```

```

        echo "</td>\n";

        echo "<td bgcolor='pink'>";
        echo "$miss ->answer_comment";
        echo "</td></tr></table>\n";
        echo "<br>\n";
    }
} else {
    echo "正解です<br>\n";
}

echo "<tr><td>この問題の正解例：";
echo "$value ->correctans";
echo "<tr><td>\n";

echo "<tr><td>解答時間：";
echo "$value ->anstime";
echo "<tr><td>\n";

echo "</table>";
echo "<br>\n";
} else {
    echo "<table width='75%' border=1 bgcolor='#999999'><tr><td>問題：";

    echo "$value ->problem";
    echo "</td></tr>\n";
    echo "<tr><td>あなたの解答：";
    echo "$value ->answer";
    echo "<tr><td>\n";
    echo "<tr><td>解答時間：";
    echo "$value ->anstime";
    echo "<tr><td>\n";

    echo "</table>";
    echo "<br>\n";
}

```

```

    }
}

/*****
 * 解答に間違いがあるかどうかによってテーブルの背景色を変更する
 */
function select_table_background($anslog) {
    global $anslog_table_bgcolor;

    //間違いがあるかどうか調べて
    if (answer_is_correct($anslog) == false) { //間違いがあれば背景色を「間違い色」にする
        $anslog_table_bgcolor = MISS_ANS_BACKGROUND;
    } else {
        $anslog_table_bgcolor = CORRECT_ANS_BACKGROUND; //間違いがなければ背景色を
        「正解色」にする
    }
    return $anslog_table_bgcolor;
}

/*****
 * 表示する
 */

//make_anslog_list();
//make_buttons_to_view_five_answers($anslog_list);
//print_anslog_list($anslog_list);
//echo"<hr>";
//print_anslog_list_by_answer_number($anslog_list,0,3);

/*****
 * カンマ区切りの数字 (3,4,5) をばらばら(3)(4)(5)にする
 */
function explode_comma($comma_text) {
    $numbers = explode(",", $comma_text);
    return $numbers;
}

```

```

/*****
 * マークされた文字の位置の配列(3)(4)(5)と
 * 文字列（えおあいう。）から、マークされた文字が赤太字になっている文字列を作成する
 */
function make_markedproblem_from_problem_and_numbers($problem, $numbers) {
    $size_of_problem = mb_strlen($problem); //問題文の文字数
    $marked_problem = "";
    $letter_pointer = 0;
    while ($letter_pointer < $size_of_problem) {
        if (letter_ismarked($letter_pointer, $numbers) == false) {
            //マークされていない文字は、そのまま返す文字列に格納する
            $letter = mb_substr($problem, $letter_pointer, 1);
            $marked_problem = $marked_problem.$letter;
        } else {
            //マークされている文字は、赤太字にして返す文字列に格納する
            $letter = mb_substr($problem, $letter_pointer, 1);
            $letter = set_letter_to_bold_letter($letter);
            $marked_problem = $marked_problem.$letter;
        }
        $letter_pointer ++;
    }
    return $marked_problem;
}

/*****
 * マークされた文字のリスト（3,4,5）と問題文から
 * マークされた文字が赤太字になってい問題文を作成する
 */
function make_markedproblem_from_problem_and_number_list($problem, $number_list) {
    if ($number_list == "NOT_SET") {
        $markedproblem = "間違いの位置は登録されていません";
    } else {
        $numbers = explode_comma($number_list);
        $markedproblem = make_markedproblem_from_problem_and_numbers($problem,
$numbers);

```



```
$encoding).$replacement.mb_substr($string, $start + $length, mb_strlen($string, $encoding),
$encoding);
```

```
}
```

```
}
```

```
}
```

```
}
```

```
/******
```

```
 * 解答が正解かどうか調べる
```

```
*/
```

```
function answer_is_correct($answer) {
    $first_correction = $answer->mistakes[0];
    if (isset($first_correction->location)) {
        return false;
    } else {
        return true;
    }
}
```

```
/******
```

```
 * 解答が添削済みかどうか調べる
```

```
*/
```

```
function answer_is_corrected($answer) {
    if (isset($answer->correctans)) {
        return true;
    } else {
        return false;
    }
}
```

```
/******
```

```
 * 問題文の◎番目の文字がマークされているかどうか調べる
```

```
*/
```

```
function letter_ismarked($letter_pointer, $marked_numbers) {
    $ismarked = false;
    foreach ($marked_numbers as $marked_number) {
```

```
        if ($marked_number == $letter_pointer) {  
            $ismarked = true;  
        }  
    }  
    return $ismarked;  
}  
?>
```

=====


```
<?php
global $con;
global $enc;

function normalize($str) {
    $str = strip_tags($str);
    $str = str_replace(" ", "", $str);
    $str = str_replace(", ", "", $str);
    $str = str_replace("'", "", $str);
    $str = str_replace("\'", "", $str);
    $str = htmlspecialchars($str);
    return $str;
}
?>
```

=====

```
<?php
```

```
/**
```

```
 * 添削内容を扱う機能郡
```

```
 */
```

```
require_once ('DBAccessor.php');
```

```
/**
```

```
 * 問題 ID と回答文から解答 ID のリストを取得する
```

```
 */
```

```
function getAnswerfromQuestionIDandAnswer($answer, $question_id) {
```

```
    // 回答の内容が同じものを取得する
```

```
    $query = "SELECT answer_id FROM answer
```

```
                WHERE answer = '$answer'
```

```
                AND question_id = $question_id";
```

```
    connect();
```

```
    $findSameResult = mysql_query($query);
```

```
    close();
```

```
    $sameAnswerIDs = null;
```

```
    $i = 0;
```

```
    if ($findSameResult) {
```

```
        while ($row = mysql_fetch_object($findSameResult)) {
```

```
            $sameAnswerIDs[$i] = $row->answer_id;
```

```
            $i ++;
```

```
        }
```

```
    } else {
```

```
        return;
```

```
    }
```

```
    return $sameAnswerIDs;
```

```
}
```

```

/**
 * 添削を追加する
 * ※添削の種類が正解だった場合、添削の対象範囲に null を入れる作業を
 * 画面側にやって貰うことになるかも
 * @$answer_id : 添削対象の回答 ID
 * @$correction_kind_id : 添削の種類 ID
 * @$correction_spot : 添削の対象範囲
 * @$answer_comment : 回答に対するコメント
 */
/*
if(addCorrection(5, 2, '1,3,5', 'ほげほげな回答ですね')) {
    echo("添削が追加されたよ");
}
*/
function addCorrection($answer_id, $correction_kind_id, $correction_spot, $answer_comment) {
    // 添削を追加する
    $query = "INSERT INTO
                                correction(correction_kind_id, answer_id, answer_comment,
correction_spot)
                                VALUES($correction_kind_id, $answer_id, '$answer_comment',
'$correction_spot');"

    // "添削済み"ラベルをつける
    $query2 = "UPDATE answer SET
                                answer.is_corrected = 1
                                WHERE answer.answer_id = $answer_id";

    // 添削を追加する
    connect();
    $result = mysql_query($query);
    $result2 = mysql_query($query2);
    if (!$result || !$result2) {
        close();
        return null;
    }
}

```

```

// 重複している回答を添削する
correctDuplications($answer_id);

close();

return $result;
}

/**
 * 重複している回答を添削する
 */
function correctDuplications($answer_id) {
    // 先に編集した回答本文を取得する
    $getAnswer = "SELECT answer.answer, answer.question_id FROM answer
                  WHERE answer.answer_id = $answer_id LIMIT 1;";

    $getAnswerResult = mysql_query($getAnswer);
    $answerObject = mysql_fetch_object($getAnswerResult);
    $answer = $answerObject->answer;
    $question_id = $answerObject->question_id;

    // 回答の内容が同じものがあれば、回答 ID を取得する
    $findSame = "SELECT answer.answer_id FROM answer
                WHERE answer.answer = $answer
                AND answer.question_id = $question_id;";
    $findSameResult = mysql_query($findSame);

    //回答の内容が同じものがあるかどうか調べて
    if ($findSameResult) {
        $sameAnswersID = null;
        while ($row = mysql_fetch_object($findSameResult)) {
            $sameAnswersID[] = $row->answer_id;
        }
    } else {
        return;
    }
}

```

```

    }

    $getCorrection = "SELECT * FROM correction
                        WHERE correction.answer_id = $answer_id;";
    $getCorrectionResult = mysql_query($getCorrection);
    $correction = null;
    while ($row = mysql_fetch_object($getCorrectionResult) != null) {
        $correction->correction_kind_id = $row->correction_kind_id;
        $correction->answer_comment = $row->answer_comment;
        $correction->correction_spot = $row->correction_spot;
    }

    for ($i = 0; $i < count($sameAnswersID); $i++) {
        $correctDuplications = "INSERT INTO correction(correction_kind_id, answer_id,
answer_comment, correction_spot)

                                VALUES($correction->correction_kind_id,                                $sameAnswersID[$i],
$correction->answer_comment, $correction->correction_spot);";

        $correctDuplicationsResult = mysql_query($correctDuplications);

        if (!$correctDuplicationsResult) {
            echo ("重複している回答の添削編集に失敗しました");
            return;
        }
    }
}

/**
 * addCorrection の close() 無し版
 */
function addCorrectionAccessless($answer_id, $correction_kind_id, $correction_spot,
$answer_comment) {
    $query = "INSERT INTO

                                correction(correction_kind_id, answer_id, answer_comment,
correction_spot)

```

```
VALUES($correction_kind_id, $answer_id, '$answer_comment',
'$correction_spot');";
```

```
$query2 = "UPDATE answer SET
            answer.is_corrected = 1
            WHERE answer.answer_id = $answer_id";
```

```
// 添削を追加する
```

```
connect();
```

```
$result = mysql_query($query);
```

```
$result2 = mysql_query($query2);
```

```
if ($result && $result2) {
    return $result;
}
```

```
}
```

```
/**
```

```
 * 未添削な回答群にある全く同じ回答の ID を取得する
```

```
 */
```

```
/*
```

```
if(($foo = isInUncorrected('ほげほげですよ', 2)) != null) {
```

```
    echo("未添削な回答群に全く同じ回答があったよ");
```

```
    echo($foo);
```

```
} else {
```

```
    echo("未添削な回答群には同じ回答はなかったよ");
```

```
}
```

```
*/
```

```
function isInUncorrected($answer, $question_id) {
```

```
    $query = "SELECT answer_id FROM answer WHERE
```

```
                answer.answer = '$answer' AND answer.question_id =
```

```
$question_id
```

```
                AND answer.is_corrected = FALSE LIMIT 1;";
```

```
connect();
```

```
$result = mysql_query($query);
```

```
    if (!$result || mysql_num_rows($result) <= 0) {  
        return null;  
    }  
  
    $row = mysql_fetch_object($result);  
  
    return $row->answer_id;  
}  
?>
```

=====

```
<?php
```

```
/**
```

```
 * AnswerData.php
```

```
 * 学生の回答データを扱う機能
```

```
 * author 川口
```

```
 */
```

```
require_once ('DBAccessor.php');
```

```
/*
```

```
 * public methods*
```

```
 */
```

```
/**
```

```
 * 学生が回答した問題の履歴と回答の履歴を取得する
```

```
 * @$login : ログイン名
```

```
 */
```

```
function getAnswerHistory($login) {
```

```
    $query = "SELECT
```

```
                answer.answer_id, answer, answer_time,
```

```
is_corrected,
```

```
                question, correct_example
```

```
FROM
```

```
                answer, question
```

```
WHERE answer.question_id = question.question_id
```

```
AND answer.student_login_name = '$login';
```

```
    // 学生が回答した問題の履歴と回答の履歴を取得する
```

```
    connect();
```

```
    $result = mysql_query($query);
```

```
    close();
```



```

$i = 0;
$answer = null;
while ($row = mysql_fetch_object($result)) {
    if ($row->is_corrected == 1) {
        // 添削済みだった場合
        $answer[$i] ->answer_id = $row ->answer_id; //解答 ID
        $answer[$i] ->answer = $row ->answer; //学生の解答
        $answer[$i] ->anstime = $row ->answer_time; //学生の解答時間
        $answer[$i] ->problem = $row ->question; //問題文
        $answer[$i] ->correctans = $row ->correct_example; //正解例
        $mistake = getCorrection($row ->answer_id);
        $answer[$i] ->mistakes = $mistake;
    } else {
        $answer[$i] ->answer_id = $row ->answer_id; //解答 ID
        $answer[$i] ->answer = $row ->answer; //学生の解答
        $answer[$i] ->anstime = $row ->answer_time; //学生の解答時間
        $answer[$i] ->problem = $row ->question; //問題文
    }

    $i ++;
}
return $answer;
}

/**
 * 次に添削される回答を取得する
 */
function getNextToBeCorrected() {
    $query = "SELECT answer FROM answer
                                                    WHERE answer.is_corrected =
FALSE LIMIT 1;";

    // 次に添削される解答を取得する
    connect();
    $result = mysql_query($query);

```

```

        close();

        if (mysql_num_rows($result) == 1) {
            $row = mysql_fetch_object($result);
            $nextAnswer = $row->answer;
        } else
            $nextAnswer = null;

        return $nextAnswer;
    }

    /**
     * 添削済みの回答のみ取得する
     */
    function getCorrectedAnswer($question_id) {
        $query = "SELECT answer FROM answer
                  WHERE answer.question_id = $question_id
                  AND answer.is_corrected = TRUE;";

        connect();
        $result = mysql_query($query);
        close();

        $i = 0;
        $correctedAnswer[] = null;
        while ($row = mysql_fetch_object($result)) {
            $correctedAnswer[$i] = $row->answer;

            $i ++;
        }

        return $correctedAnswer;
    }

    /**
     * 回答を追加する

```

```

* @$answer : 回答文
* @$question_id : 問題 ID
* @$login : ログイン名
*/
function addAnswer($answer, $question_id, $login) {
    global $time;

    $query = "INSERT INTO answer(question_id, student_login_name, answer, answer_time,
is_corrected)
                                VALUES($question_id, '$login', '$answer', '$time',
FALSE)";

    // 回答を追加する
    connect();
    $result = mysql_query($query);

    // 追加した回答の ID を取得する
    global $db;
    $answer_id = mysql_insert_id($db);

    // 重複している回答を自動添削する
    require_once ('CorrectionData.php');
    if (($existingAnswer_id = isInCorrected($answer_id, $answer, $question_id)) != null) {
        // 添削済みの回答郡の中に、全く同じ回答が存在した場合
        // 存在した回答の添削を取得する
        $existingCorrectionIDs = getCorrectionConnectless($existingAnswer_id);

        if ($existingCorrectionIDs != null) {
            // 同じ内容の回答に添削がついていた場合
            // 追加された回答へ同じ添削をつける
            foreach ($existingCorrectionIDs as $value) {
                foreach ($value as $wvalue) {
                    addCorrectionToMirror($wvalue, $answer_id);
                }
            }
        }
    }
}

```

```

    }
    close();

    return $result;
}

/**
 * 添削済みな回答の中に、同じ回答がないか調べる
 */
function isInCorrected($answer_id, $answer, $question_id) {
    $query = "SELECT answer_id FROM answer
              WHERE answer.question_id = $question_id
              AND answer.answer = '$answer'
              AND answer.is_corrected = TRUE
              AND answer.answer_id != $answer_id LIMIT 1;";

    $result = mysql_query($query);

    // 問題に対する初の回答であった場合
    if (!$result || mysql_num_rows($result) <= 0) {
        return null;
    }

    $row = mysql_fetch_object($result);

    return $row->answer_id;
}

/**
 * 添削済みな回答の中に、同じ回答がないか調べて、
 * 同じ解答の解答 ID をすべて返す
 */
function isInCorrected_getAnswerIds($answer, $question_id) {
    $query = "SELECT answer_id FROM answer
              WHERE answer.question_id = $question_id
              AND answer.answer = '$answer'

```

```
AND answer.is_corrected = TRUE;";
```

```
connect();
$result = mysql_query($query);
close();
$i = 0;
$answer_ids[] = null;
while ($row = mysql_fetch_object($result)) {
    $answer_ids[$i] = $row->answer_id;
    $i++;
}
return $answer_ids;
}
```

```
/******
```

```
 * private methods*
```

```
******/
```

```
/**
```

```
 * 学生の回答に対する添削を取得する
```

```
 * @$answer_id : 回答 ID
```

```
 */
```

```
function getCorrection($answer_id) {
```

```
    $query = "SELECT
```

```
                                answer_comment, correction_spot,
```

```
                                correction_kind,
```

```
correction_kind_comment,correction_kind.correction_kind_id
```

```
FROM
```

```
                                correction,
```

```
                                correction_kind,
```

```
correction_object
```

```
WHERE      correction.correction_kind_id      =
```

```
correction_kind.correction_kind_id
```

```
AND correction_object.answer_id = '$answer_id'
```

```
AND      correction_object.correction_id      =
```

```
correction.correction_id;";
```

```
// 学生の回答に対する添削を取得する
```

```

connect();
$result = mysql_query($query);
close();
if (!$result) {
    return;
}

$i = 0;
$correction = null;
while ($row = mysql_fetch_object($result)) {
    $correction[$i] ->answer_comment = $row ->answer_comment;
    $correction[$i] ->type = $row ->correction_kind;
    if ($row ->correction_kind_id != 1) {
        // 回答が正解ではなかった場合
        $correction[$i] ->location = $row ->correction_spot;
        $correction[$i] ->correction = $row ->correction_kind_comment;
    }

    $i ++;
}

return $correction;
}

function getCorrectionConnectless($answer_id) {
    $query = "SELECT
                                correction_id
                                FROM
                                correction_object
                                WHERE      correction_object.answer_id      =
'$answer_id'";

    // 学生の回答に対する添削を取得する
    $result = mysql_query($query);

    if (!$result) {

```

```
        return;
    }

    $correction = null;
    $i = 0;
    while ($row = mysql_fetch_object($result)) {
        $correction[$i] =>correction_id = $row->correction_id;

        $i ++;
    }

    return $correction;
}
?>
```

=====

```
<?php
```

```
/**
```

```
 * 学生の回答データを扱う機能郡
```

```
 */
```

```
require_once ('DBAccessor.php');
```

```
/**
```

```
 * 回答の情報を取得するときのフォーマット
```

```
 */
```

```
class Answer {
```

```
    var $answer_id;
```

```
    var $answer;
```

```
    var $anstime;
```

```
    var $problem;
```

```
    var $correctans;
```

```
    var $mistakes;
```

```
}
```

```
/**
```

```
 * 添削の情報を取得するときのフォーマット
```

```
 */
```

```
class Correction {
```

```
    var $right_answer;
```

```
    var $answer_comment;
```

```
    var $correction_range;
```

```
    var $location;
```

```
    var $type;
```

```
    var $correction;
```

```
}
```

```
/**
```



```

* 学生が回答した問題の履歴と回答の履歴を取得する
* @$login : ログイン名
*/
/*
$getAHTest = getAnswerHistory('s00000aa');
foreach($getAHTest as $key => $value) {
    //echo($value->correction->answer_comment."<br>");

    foreach($value as $swalue) {
        echo($swalue."<br>");
    }

    echo("<br>");
}
*/
function getAnswerHistory($login) {
    $query = "SELECT
                answer_id, answer, answer_time, is_corrected,
                question, correct_example
            FROM
                answer, question
            WHERE answer.question_id = question.question_id
            AND answer.student_login_name = '".$login."'";

    // 学生が回答した問題の履歴と回答の履歴を取得する
    connect();
    $result = mysql_query($query);
    close();

    $i = 0;
    $answer[] = new Answer();
    while ($row = mysql_fetch_object($result)) {
        if ($row->is_corrected == 1) {
            // 添削済みだった場合
            $answer[$i]->answer_id = $row->answer_id; //解答 ID
            $answer[$i]->answer = $row->answer; //学生の解答

```

```

        $answer[$i] ->anstime = $row ->answer_time; //学生の解答時間
        $answer[$i] ->problem = $row ->question; //問題文
        $answer[$i] ->correctans = $row ->correct_example; //正解例
        $mistake = getCorrection($row ->answer_id);
        $answer[$i] ->mistakes = $mistake;
    }

    $i ++;
}

return $answer;
}

/**
 * 添削を取得する
 * @$answer_id : 回答 ID
 */
/*
$getCorTest = getCorrection(3);
foreach($getCorTest as $key => $value) {
    foreach($value as $wvalue) {
        echo($wvalue."<br>");
    }

    echo("<br>");
}
*/
function getCorrection($answer_id) {
    $query = "SELECT
                answer_comment, correction_spot,
                correction_kind, correction_kind_comment
            FROM
                correction, correction_kind
            WHERE
                correction.correction_kind_id =
correction_kind.correction_kind_id
            AND correction.answer_id = '".$answer_id."'";

```

```

// 学生の回答に対する添削を取得する
connect();
$result = mysql_query($query);
close();

$i = 0;
$correction[] = new Correction();
while ($row = mysql_fetch_object($result)) {
    $correction[$i] ->answer_comment = $row ->answer_comment;
    $correction[$i] ->type = $row ->correction_kind;
    if ($row ->correction_kind != 1) {
        // 回答が正解ではなかった場合
        $correction[$i] ->location = $row ->correction_spot;
        $correction[$i] ->correction = $row ->correction_kind_comment;
    }

    $i ++;
}

return $correction;
}

/**
 * getCorrection の connect/close 無し版
 */
function getCorrectionAccessless($answer_id) {
    $query = "SELECT
                                answer_comment,                correction_spot,
correction_kind.correction_kind_id,
                                correction_kind, correction_kind_comment
                                FROM
                                correction, correction_kind
                                WHERE correction.correction_kind_id =
correction_kind.correction_kind_id
                                AND correction.answer_id = '".$answer_id."'";

```

```

// 学生の回答に対する添削を取得する
connect();
$result = mysql_query($query);

$i = 0;
$correction[] = new Correction();
while ($row = mysql_fetch_object($result)) {
    $correction[$i] ->answer_comment = $row ->answer_comment;
    $correction[$i] ->correction_kind_id = $row ->correction_kind_id;
    $correction[$i] ->type = $row ->correction_kind;
    if ($row ->correction_kind != 1) {
        // 回答が正解ではなかった場合
        $correction[$i] ->location = $row ->correction_spot;
        $correction[$i] ->correction = $row ->correction_kind_comment;
    }

    $i ++;
}

return $correction;
}

/**
 * 次に添削される回答を取得する
 */
/*
//echo(getNextToBeCorrected());
function getNextToBeCorrected() {
    $query = "SELECT answer, question_id FROM answer
              WHERE answer.is_corrected = FALSE LIMIT 1;";

    // 次に添削される解答を取得する
    connect();
    while(true) {
        $result = mysql_query($query);

```

```

        if(mysql_num_rows($result) == 1) {
            $row = mysql_fetch_object($result);
            $nextAnswer = $row->answer;
            $question_id = $row->question_id;
        } else {
            close();
            $nextAnswer = null;
        }
        echo("#####");
        // 拾ってきた回答が既に添削されている回答と同じ場合
        if(($existingCorrection_id = isInCorrected($nextAnswer, $question_id)) !=
null) {

            echo("hogehege");
            // 同じ回答の添削を拾ってきて、自動的に添削する
            $existingCorrections =
getCorrectionAccessless($existingCorrection_id);
            if(existingCorrections != null) {
                foreach($existingCorrections as $value) {
                    addCorrectionAccessless($question_id,
$value->correction_kind_id, $value->location, $value->answer_comment);
                }
            }

            continue;
        }

        break;
    }

    close();

    return $nextAnswer;
}
*/

/**

```

```

* 添削済みの回答のみ取得する
*/
/*
$getCrAns = getCorrectedAnswer(1);
foreach($getCrAns as $value) {
    echo($value."<br>");
}
*/
function getCorrectedAnswer($question_id) {
    $query = "SELECT answer FROM answer
              WHERE answer.question_id = $question_id
              AND answer.is_corrected = TRUE;";

    connect();
    $result = mysql_query($query);
    close();

    $i = 0;
    $correctedAnswer[] = null;
    while ($row = mysql_fetch_object($result)) {
        $correctedAnswer[$i] = $row->answer;

        $i ++;
    }

    return $correctedAnswer;
}

/**
* 回答を追加する
* @$answer : 回答文
* @$question_id : 問題 ID
* @$login : ログイン名
*/
// $addAnswerTest = addAnswer('よくわかりません(^ω^)', '3', 's22222cc');
// if($addAnswerTest) echo("回答が追加されたよ");

```

```

//      if(addAnswer('ぶぎゃーなんじゃね?', 3, 's00000aa')) echo("回答が追加されました");

function addAnswer($answer, $question_id, $login) {
    global $time;

    $query = "INSERT INTO
                                                    answer(question_id, student_login_name, answer,
answer_time, is_corrected)
                                                    VALUES($question_id,    '$login',    '$answer',
'$time', FALSE);";

    connect();
    // 回答を追加する
    $result = mysql_query($query);

    // 重複している回答を自動添削する
    require_once ('CorrectionData.php');
    if (($existingAnswer_id = isInCorrected($answer, $question_id)) != null) {
        // 添削済みの回答郡の中に、全く同じ回答が存在した場合
        // 存在した回答の添削を取得する
        $existingCorrections = getCorrectionAccessless($existingAnswer_id);

        if ($existingCorrections != null) {
            // 被っている回答の数を調べる
            $getExistingNum = "SELECT answer_id FROM answer
                                                    WHERE    answer.answer    =
'$answer' AND answer.student_login_name = '$login'";
            $getExistingNumResult = mysql_query($getExistingNum);
            $existingNum = mysql_num_rows($getExistingNumResult);
            $existingNumPre = $existingNum -1;

            // たった今追加された回答の ID を取得（たった今追加された回答は、回
            答が一致しログイン名も一致する回答の中の最後尾のものとして処理している・・・可能性はごく極めて
            低いとはいえ、バグは起こりうる）
            $getThisId = "SELECT answer_id FROM answer

```

```

WHERE answer.answer =
'$answer' AND answer.student_login_name = '$login' LIMIT $existingNumPre, $existingNum;";
    $getThisIdResult = mysql_query($getThisId);
    if ($getThisIdResult) {
        $row = mysql_fetch_object($getThisIdResult);
    }

    // たった今追加された回答へ、存在した回答に対する添削と同様の添削
を埋め込む

    foreach ($existingCorrections as $value) {
        addCorrectionAccessless($row->answer_id,
$value->correction_kind_id, $value->location, $value->answer_comment);
    }
}
}
close();

return $result;
}

/**
 * 今までの回答の中に同じ回答が含まれていないか調べる
 */
/*
if(($hoge = isInCorrected('あいいうえおあ', 2)) != null) {
    echo("既に添削済みな回答郡に存在している回答だよ!");
    echo($hoge);
} else {
    echo("添削済みの回答郡の中では初めての回答だったよ");
}
*/

function isInCorrected($answer, $question_id) {
    $query = "SELECT answer_id FROM answer WHERE
answer.question_id = $question_id AND
answer.answer = '$answer' AND answer.is_corrected = TRUE LIMIT 1;";

```



```
connect();
$result = mysql_query($query);

// 問題に対する初の回答であった場合
if (!$result || mysql_num_rows($result) <= 0) {
    return null;
}

$row = mysql_fetch_object($result);

return $row->answer_id;
}
?>
```

=====

```
<?php
```

```
/**
```

```
 * CorrectionData.php
```

```
 * 添削内容を扱う機能郡
```

```
 * author 川口
```

```
 *
```

```
 * addCorrection に global $db があるので、後々修正する、かも。
```

```
 */
```

```
require_once ('DBAccessor.php');
```

```
/*
```

```
 * public methods*
```

```
 */
```

```
/**
```

```
 * 問題 ID と回答文から解答 ID のリストを取得する
```

```
 */
```

```
function getAnswerfromQuestionIDandAnswer($answer, $question_id) {
```

```
    // 回答の内容が同じものを取得する
```

```
    $query = "SELECT answer_id FROM answer
```

```
                WHERE answer.answer = '$answer'
```

```
                AND answer.question_id = $question_id";
```

```
    connect();
```

```
    $findSameResult = mysql_query($query);
```

```
    close();
```

```
    $sameAnswerIDs = null;
```

```
    $i = 0;
```

```
    if ($findSameResult) {
```

```
        while ($row = mysql_fetch_object($findSameResult)) {
```

```

        $sameAnswerIDs[$i] = $row->answer_id;
        $i ++;
    }
} else {
    return;
}
return $sameAnswerIDs;
}

/**
 * 未添削な回答に添削する
 * @$answer_id : 添削対象の回答 ID
 * @$correction_kind_id : 添削の種類 ID
 * @$correction_spot : 添削の対象範囲
 * @$answer_comment : 回答に対するコメント
 */
function addCorrection($answer_id, $correction_kind_id, $correction_spot, $answer_comment) {
    // 添削を追加する
    $query = "INSERT INTO
                                correction(correction_kind_id,          answer_comment,
correction_spot)
                                VALUES($correction_kind_id,          '$answer_comment',
'$correction_spot');"
    connect();
    begin();
    $result = mysql_query($query);
    global $db;
    $correction_id = mysql_insert_id($db);
    if (!$result) {
        echo("添削の追加に失敗しました");
        rollback();
        close();
        return null;
    }

    // 添削オブジェクトを追加する

```

```

    $addCorrectionObject =
    "INSERT INTO correction_object(correction_id, answer_id)
    VALUES($correction_id, $answer_id);";
    $addObjectResult = mysql_query($addCorrectionObject);
    if (!$addObjectResult) {
        echo("添削オブジェクトの追加に失敗しました");
        rollback();
        close();
        return null;
    }

    // "添削済み"ラベルをつける
    $query2 = "UPDATE answer SET
                answer.is_corrected = 1
                WHERE answer.answer_id = $answer_id";
    $result2 = mysql_query($query2);
    if (!$result2) {
        echo("answer の添削済みラベルの更新に失敗しました");
        rollback();
        close();
        return null;
    }

    // 重複した回答を添削する
    correctDuplications($correction_id, $answer_id);

    commit();
    close();

    return $result;
}

/**
 * 添削済みな回答に添削をする
 */
function addRecorrection($answer_id, $correction_kind_id, $correction_spot, $answer_comment) {

```

```

// 添削を追加する
$query = "INSERT INTO
                                correction(correction_kind_id,
correction_spot)                                answer_comment,
                                VALUES($correction_kind_id,
'$correction_spot');";                                '$answer_comment',

connect();
begin();
$result = mysql_query($query);
global $db;
$correction_id = mysql_insert_id($db);
if (!$result) {
    echo("添削の追加に失敗しました");
    rollback();
    close();
    return null;
}

if($correction_kind_id == 1) {
// 添削内容が正解だった場合
    // 添削リンクを初期化する
    resetCorrectionObject($answer_id);
}

// 添削オブジェクトを追加する
$addCorrectionObject =
"INSERT INTO correction_object(correction_id, answer_id)
VALUES($correction_id, $answer_id);";
$addObjectResult = mysql_query($addCorrectionObject);
if (!$addObjectResult) {
    echo("添削オブジェクトの追加に失敗しました");
    rollback();
    close();
    return null;
}

```

```

// 重複した回答を添削する
correctDuplications($correction_id, $answer_id);

commit();
close();
}

/**
 * 内容が同じ回答が添削済みだった回答に添削する
 */
function addCorrectionToMirror($correction_id, $answer_id) {
    // 添削リンクを追加する
    $addCorrectionObject =
        "INSERT INTO correction_object(correction_id, answer_id)
        VALUES($correction_id, $answer_id)";
    begin();
    $addObjectResult = mysql_query($addCorrectionObject);
    if (!$addObjectResult) {
        echo("添削オブジェクトの追加に失敗しました");
        rollback();
        return null;
    }

    // "添削済み"ラベルをつける
    $query2 = "UPDATE answer SET
                answer.is_corrected = 1
                WHERE answer.answer_id = $answer_id";
    $result2 = mysql_query($query2);
    if (!$result2) {
        echo("answer の添削済みラベルの更新に失敗しました");
        rollback();
        close();
        return null;
    }

    commit();
}

```

```
}
```

```
/**
```

```
 * 未添削な回答群にある全く同じ回答の ID を取得する
```

```
*/
```

```
function isInUncorrected($answer, $question_id) {
```

```
    $query = "SELECT answer_id FROM answer WHERE
```

```
        answer.answer = '$answer' AND answer.question_id = $question_id
```

```
        AND answer.is_corrected = FALSE LIMIT 1;";
```

```
    connect();
```

```
    $result = mysql_query($query);
```

```
    if (!$result || mysql_num_rows($result) <= 0) {
```

```
        return null;
```

```
    }
```

```
    $row = mysql_fetch_object($result);
```

```
    return $row->answer_id;
```

```
}
```

```
/**
```

```
 * 重複している回答を添削する
```

```
*/
```

```
function correctDuplications($correction_id, $answer_id) {
```

```
    // 添削をしている回答本文を取得する
```

```
    $getAnswer = "SELECT answer.answer, answer.question_id FROM answer
```

```
        WHERE answer.answer_id = $answer_id LIMIT 1;";
```

```
    $getAnswerResult = mysql_query($getAnswer);
```

```
    $answerObject = mysql_fetch_object($getAnswerResult);
```

```
    $answer = $answerObject->answer;
```

```
    $question_id = $answerObject->question_id;
```

```
    // 回答の内容が同じものがあれば、回答 ID を取得する
```

```

$findSame = "SELECT answer.answer_id FROM answer
              WHERE answer.answer = '$answer'
              AND answer.question_id = $question_id
              AND answer.answer_id != $answer_id;";
$findSameResult = mysql_query($findSame);

if (mysql_num_rows($findSameResult) > 0) {
    // 内容が同じ回答があった場合
    $sameAnswersID = null;
    while ($row = mysql_fetch_object($findSameResult)) {
        $sameAnswersID[] = $row->answer_id;
    }
} else {
    // 内容が同じ回答がなかった場合
    return;
}

foreach ($sameAnswersID as $value) {
    $correctDuplications = "INSERT INTO correction_object(correction_id,
answer_id)
                                VALUES($correction_id,
$value);";

    $correctDuplicationsResult = mysql_query($correctDuplications);

    if (!$correctDuplicationsResult) {
        echo ("重複している回答の添削編集に失敗しました");
        return;
    }
}

}

/*****
* private methods*
*****/

```



```

function resetCorrectionObject($answer_id) {
    $query =
        "SELECT correction_id FROM correction_object
         WHERE correction_object.answer_id = $answer_id;";

    $result = mysql_query($query);

    $correction = null;
    $i = 0;
    while($row = mysql_fetch_object($result)) {
        $correction[$i] = $row->correction_id;

        $i++;
    }

    if(count($correction) < 1) return;

    foreach($correction as $value) {
        echo("リセット対象添削 ID:".$value."<br>");
        deleteCorrection($value);
    }
}

```

```

function deleteCorrection($correction_id) {
    $resetCorrection =
        "DELETE FROM correction_object
         WHERE correction_object.correction_id = $correction_id;";
    echo($resetCorrection);
    $result = mysql_query($resetCorrection);
    if(!$result) {
        echo("添削オブジェクトのリセットに失敗しました");
        return;
    }
}

```

?>

=====

```
<?php
```

```
/**
```

```
 * CorrectionKindData.php
```

```
 * 添削の種類を扱う機能郡
```

```
 * author 川口
```

```
 */
```

```
require_once ('DBAccessor.php');
```

```
/*
```

```
 * public methods*
```

```
 */
```

```
/**
```

```
 * 添削の種類を追加する
```

```
 * @$correction_kind : 添削の種類
```

```
 * @$correction_kind_comment : 添削の種類のコメント
```

```
 */
```

```
function addCorrectionKind($correction_kind, $correction_kind_comment) {
```

```
    $query = "INSERT INTO
```

```
                correction_kind(correction_kind, correction_kind_comment)
```

```
                VALUES('$correction_kind', '$correction_kind_comment');";
```

```
    connect();
```

```
    $result = mysql_query($query);
```

```
    close();
```

```
    return $result;
```

```
}
```

```
?>
```

```
=====
```

```
<?php
```

```
/**
```

```
 * DBAccessor.php
```

```
 * DB との接続/切断と SQL 文の発行を行う機能
```

```
 * author 川口
```

```
 * 2007/1/10 松澤 修正
```

```
 *
```

```
 * $db アクセス時の同期処理を確認すること
```

```
 */
```

```
//MySQL の設定
```

```
define('SERVER', 'localhost'); // サーバー名
```

```
define('USER', 'jun'); // ユーザー名
```

```
define('PASS', 'shige'); // パスワード
```

```
define('DBNAME', 'jun'); // DB 名
```

```
define('ENCODE', 'utf8'); // エンコード
```

```
$time = date('Y-m-d H:i:s'); // 日時の取得変数
```

```
$db = null; // DB 接続
```

```
/**
```

```
 * DB に接続する
```

```
 */
```

```
function connect() {
```

```
    global $db;
```

```
    // DB へ接続を試みる
```

```
    $db = mysql_connect(SERVER, USER, PASS);
```

```
    // DB を選択する
```

```
    mysql_select_db(DBNAME, $db);
```

```
        // エンコードを指定する
        mysql_query("SET NAMES".ENCODE);
    }
}
```

```
/**
 * DB から切断する
 */
function close() {
    global $db;

    // DB から切断する
    mysql_close($db);
}
```

```
/**
 * BEGIN クエリを発行する
 */
function begin() {
    mysql_query("BEGIN");
}
```

```
/**
 * COMMIT クエリを発行する
 */
function commit() {
    mysql_query("COMMIT");
}
```

```
/**
 * ROLLBACK クエリを発行する
 */
function rollback() {
    mysql_query("ROLLBACK");
}
?>
```

=====

```
<?php
```

```
/**
```

```
 * 添削に必要なデータを扱う機能部
```

```
 * MegumiAraki 2006/12/20
```

```
 */
```

```
require_once ('DBAccessor.php');
```

```
/******
```

```
 * public methods*
```

```
******/
```

```
/**
```

```
 * 未添削の中で一番最初の回答を取得する
```

```
 */
```

```
function getUnCorrectedAnswerSet() {
```

```
    $query = "select * from question,answer where answer.is_corrected=FALSE and  
answer.question_id=question.question_id ORDER BY answer.answer LIMIT 1;";
```

```
    // 未添削の解答、解答 ID,問題、問題 ID を取得する
```

```
    connect();
```

```
    while (true) {
```

```
        $result = mysql_query($query);
```

```
        if (!$result || mysql_num_rows($result) <= 0) {
```

```
            close();
```

```
            return null;
```

```
        }
```

```
        $answer_set = null;
```

```
        while ($row = mysql_fetch_object($result)) {
```

```

        $answer_set->answer_id = $row->answer_id;
        $answer_set->answer = $row->answer;
        $answer_set->question_id = $row->question_id;
        $answer_set->question = $row->question;
    }

    // 重複した回答を自動添削する
    require_once ('AnswerData.php');
    require_once ('CorrectionData.php');
    if (($correctedAnswerID = isInCorrected($answer_set->answer_id,
$answer_set->answer, $answer_set->question_id)) != null) {
        // 添削済みの中に同じ回答が存在した場合
        // 同じ回答の添削を拾ってきて、自動的に添削する
        $existingCorrections =
getCorrectionConnectless($correctedAnswerID);
        foreach ($existingCorrections as $value) {
            foreach($value as $value) {
                addCorrectionToMirror($value,
$answer_set->answer_id);
            }
        }

        continue;
    }

    break;
}

close();

return $answer_set;
}

function deleteCorrectionLink($correction_id) {
    $deleteLink =
"DELETE FROM correction_object

```



```

        WHERE correction_object.correction_id = $correction_id";

    connect();
    $deleteLinkResult = mysql_query($deleteLink);
    close();
    if(!$deleteLinkResult) {
        echo("添削のリンク削除に失敗しました");
        return;
    }
}

/**
 * 問題 ID から解答をすべて取得する (問題 ID)
 */
function getAllAnswersfromQuestionID($question_id) {
    $query = "SELECT answer,answer_id FROM answer WHERE answer.question_id = $question_id
and answer.is_corrected=1";
    connect();
    $result = mysql_query($query);
    close();
    $i = 0;
    $answers = null;
    while ($row = mysql_fetch_object($result)) {
        $answers[$i] >answer = $row >answer;
        $answers[$i] >answer_id = $row >answer_id;
        $i ++;
    }

    return $answers;
}

/**
 * 解答 ID から添削をすべて取得する (解答 ID)
 */
function getAllCorrectionfromAnswerID($answer_id) {
    $query =

```

```
"SELECT * FROM answer, question, correction, correction_object, correction_kind
WHERE answer.answer_id = $answer_id
AND answer.question_id = question.question_id
AND correction_object.answer_id = $answer_id
AND correction_object.correction_id = correction.correction_id
AND correction.correction_kind_id = correction_kind.correction_kind_id";
```

```
connect();
$result = mysql_query($query);
close();
if(!$result) {
    echo("この回答は未添削です");
    return null;
}
```

```
// 取得したデータを切り分けて扱える様にする
```

```
$i = 0;
$correction_set = null;
while ($row = mysql_fetch_object($result)) {
    $correction_set[$i] =>answer = $row->answer;
    $correction_set[$i] =>answer_id = $row->answer_id;
    $correction_set[$i] =>correction_id = $row->correction_id;
    $correction_set[$i] =>correction_kind = $row->correction_kind;
    $correction_set[$i] =>correction_kind_id = $row->correction_kind_id;
    $correction_set[$i] =>answer_comment = $row->answer_comment;
    $correction_set[$i] =>correction_spot = $row->correction_spot;
    $correction_set[$i] =>question = $row->question;
    $correction_set[$i] =>question_id = $row->question_id;

    $i ++;
}
```

```
return $correction_set;
```

```
}
```

```
/**
```

```

* 全ての添削の種類を取得する
*/
function getCorrectionKinds() {
    $query = "SELECT *
                FROM correction_kind
                WHERE correction_kind.correction_kind_id > 1
                ORDER BY correction_kind;";

    connect();
    $result = mysql_query($query);
    close();

    $i = 0;
    $correction_kinds = null;
    while ($row = mysql_fetch_object($result)) {
        $correction_kinds[$i] >correction_kind_id = $row >correction_kind_id;
        $correction_kinds[$i] >correction_kind = $row >correction_kind;
        $correction_kinds[$i] >correction_kind_comment =
    $row >correction_kind_comment;
        $i ++;
    }

    return $correction_kinds;
}
?>

```

=====

```

<?php
/**
 * QuestionData.php
 * 問題周りのデータを扱う機能郡
 * author 川口
 */

require_once ('DBAccessor.php');

/*****
 * public methods*
 *****/
/**
 * 出題出来る問題を取得する
 */
function getCandidateQuestion() {
    $query = "SELECT question, question_id, student_visible FROM question
              WHERE question.admin_visible = 1";

    connect();
    $result = mysql_query($query);
    close();

    $i = 0;
    $candidate = null;
    while ($row = mysql_fetch_object($result)) {
        $candidate[$i] =>question = $row->question;
        $candidate[$i] =>question_id = $row->question_id;
        $candidate[$i] =>student_visible = $row->student_visible;

        $i ++;
    }
}

```

```

        return $candidate;
    }

    /**
     * 出題されている問題を取得する
     */
    function getSelectedQuestion() {
        $query = "SELECT question, question_id FROM question
                    WHERE question.student_visible = TRUE
                    ORDER BY question.question_order;";

        connect();
        $result = mysql_query($query);
        close();

        $i = 0;
        $selected[] = null;
        while (mysql_num_rows($result) > 0 && $row = mysql_fetch_object($result)) {
            $selected[$i] =>question = $row->question;
            $selected[$i] =>question_id = $row->question_id;

            $i ++;
        }

        if ($i > 0)
            return $selected;
        else
            return null;
    }

    /**
     * 問題を追加する
     * デフォルトで admin からは可視、学生からは不可視となる
     * @$question : 問題文
     * @$correct_example : 解答例

```

```

*/
function addQuestion($question, $correct_example) {
    $begin = "BEGIN";
    $countGetter = "select count(*) from question;";
    $count = null;
    $query = "INSERT INTO
                                question(question,          correct_example,
question_order, admin_visible, student_visible)
                                VALUES('$question', '$correct_example', 7, TRUE,
FALSE);"
    $commit = "COMMIT";

    connect();
    $beginResult = mysql_query($begin);
    $countGetterResult = mysql_query($countGetter);
    $count = $countGetterResult;
    $result = mysql_query($query);
    $commitResult = mysql_query($commit);
    close();

    return $result;
}

```

```

/**
 * admin から不可視にする
 * @$question_id : 問題 ID
 */
function adminInvisible($question_id) {
    $query = "UPDATE question SET
                                question.admin_visible = FALSE,
                                question.student_visible = FALSE
                                WHERE question.question_id = $question_id;";

    connect();
    $result = mysql_query($query);
    close();
}

```

```

        return $result;
    }

    /**
     * 学生が問題を解ける様にする（回答画面から見える様にする）
     * ※admin から不可視で学生から可視にも出来るが、画面仕様上それは出来ない、かも
     */
    function studentVisible($question_id) {
        $query = "UPDATE question SET
                    question.student_visible = TRUE
                    WHERE question.question_id = $question_id;";

        connect();
        $result = mysql_query($query);
        close();

        return $result;
    }

    /**
     * 学生が問題を解けない様にする（回答画面から見えない様にする）
     * @$question : 問題 ID
     */
    function studentInvisible($question_id) {
        $query = "UPDATE question SET
                    question.student_visible = FALSE
                    WHERE question.question_id = $question_id;";

        connect();
        $result = mysql_query($query);
        close();

        return $result;
    }

```

```

/**
 * 問題の順番を入れ替える
 * @$question_id1 : 並べ替えたい問題その1
 * @$question_id2 : 並べ替えたい問題その2
 */
function swapOrder($tobeSwaped_former, $tobeSwaped_latter) {
    // 順番を交換する片方のデータの現在の順番を取得する SQL 文
    $getFormerOrder = "SELECT question_order FROM question
                        WHERE question.question_id = $tobeSwaped_former LIMIT 1;";

    // 順番を交換するもう片方のデータの現在の順番を取得する SQL 文
    $getLatterOrder = "SELECT question_order FROM question
                        WHERE question.question_id = $tobeSwaped_latter LIMIT 1;";

    connect();
    $gotFormerOrder = mysql_query($getFormerOrder);
    $gotLatterOrder = mysql_query($getLatterOrder);

    while ($gotFormerOrder && $row = mysql_fetch_object($gotFormerOrder)) {
        $formerOrder = $row->question_order;
    }
    while ($gotLatterOrder && $row = mysql_fetch_object($gotLatterOrder)) {
        $latterOrder = $row->question_order;
    }

    if ($getFormerOrder && $getLatterOrder) {
        $updateFormer = "UPDATE question SET question.question_order = $latterOrder
                        WHERE          question.question_id          =
$tobeSwaped_former;";

        $updateLatter = "UPDATE question SET question.question_order = $formerOrder
                        WHERE          question.question_id          =
$tobeSwaped_latter;";

        $resultFormer = mysql_query($updateFormer);
        $resultLatter = mysql_query($updateLatter);
    }
}

```



```

        close();

        return $resultFormer && $resultLatter;
    } else {
        close();
        return false;
    }
}

/**
 * エクセル出力用のデータを取得する
 */
function getExcelSources() {
    $query = "SELECT
                                question, correct_example, login_name, clazz, answer,
correction_kind,correction_kind_comment,answer_comment
                                FROM
                                question,      student,      answer,      correction,
correction_kind,correction_object
                                WHERE answer.question_id = question.question_id
                                AND answer.student_login_name = student.login_name
                                AND answer.answer_id = correction_object.answer_id
AND correction_object.correction_id = correction.correction_id
                                AND      correction.correction_kind_id      =
correction_kind.correction_kind_id
                                ORDER BY question.question";

    connect();
    $result = mysql_query($query);
    close();

    $i = 0;
    $excel[] = null;
    while ($row = mysql_fetch_object($result)) {
        $excel[$i] =>problem = $row->question; // 問題
    }
}

```

```

        $excel[$i] >correctans = $row >correct_example; // 正解例
        $excel[$i] >student = $row >login_name; // ログイン名
        $excel[$i] >classname = $row >clazz; // クラス名
        $excel[$i] >answer = $row >answer; // 学生の回答
        $excel[$i] >mistake = $row >correction_kind; // 間違いの種類
        $excel[$i] >correction_kind_comment = $row >correction_kind_comment; //間違い
        に対するコメント

        $excel[$i] >answer_comment = $row >answer_comment; //解答に対するコメント

        $i ++;
    }

    return $excel;
}

/**
 * エクセル出力用のデータを問題指定で取得する
 * @$question_id : 問題 ID
 */
function getExcelSource($question_id) {
    $query = "SELECT
                question, correct_example, login_name, clazz, answer,
        correction_kind
            FROM
                question, student, answer, correction, correction_kind
        WHERE question.question_id = $question_id
        AND answer.question_id = question.question_id
        AND answer.student_login_name = student.login_name
        AND answer.answer_id = correction.answer_id
        AND                correction.correction_kind_id                =
        correction_kind.correction_kind_id";

    connect();
    $result = mysql_query($query);
    close();
}

```

```
$i = 0;
$excel[] = new Excel();
while ($row = mysql_fetch_object($result)) {
    $excel[$i] >question = $row >question; // 問題
    $excel[$i] >correct_example = $row >correct_example; // 正解例
    $excel[$i] >login_name = $row >login_name; // ログイン名
    $excel[$i] >clazz = $row >clazz; // クラス名
    $excel[$i] >answer = $row >answer; // 学生の回答
    $excel[$i] >correction_kind = $row >correction_kind; // 間違いの種類

    $i ++;
}

return $excel;
}
?>
```

=====

```
<?php
```

```
/**
```

```
 * StudentData.php
```

```
 * 学生個々の情報を扱う機能郡
```

```
 * author 川口
```

```
 */
```

```
require_once ('DBAccessor.php');
```

```
define('UNIT', 1); // (行の数が) 一つ
```

```
/******
```

```
 * public methods*
```

```
******/
```

```
/**
```

```
 * ログイン名からパスワードを取得する
```

```
 * @$login : ログイン名
```

```
 */
```

```
function getPassword($login) {
```

```
    $query = "SELECT passwd FROM student WHERE
```

```
                student.login_name = '$login';";
```

```
    // パスワードを取得する
```

```
    connect();
```

```
    $result = mysql_query($query);
```

```
    close();
```

```
    if (mysql_num_rows($result) == UNIT) {
```

```
        $row = mysql_fetch_object($result);
```

```
        $pass = $row->passwd;
```

```
    } else
```

```

        $pass = null;

        return $pass;
    }

    /**
     * ログイン名から氏名を取得する
     * @$login : ログイン名
     */
    function getName($login) {
        $query = "SELECT name FROM student WHERE login_name = '$login'";

        // 氏名を取得する
        connect();
        $result = mysql_query($query);
        close();

        if (mysql_num_rows($result) == UNIT) {
            $row = mysql_fetch_object($result);
            $name = $row->name;
        } else
            $name = null;

        return $name;
    }

    /**
     * 学生の登録情報を取得する
     * ※一応これで登録情報は何でも取って来れる
     * @$login : ログイン名
     */
    function getStudentRegister($login) {
        $query = "SELECT * FROM student WHERE login_name = '$login'";

        // 学生の登録情報を取得する
        connect();

```

```

        $result = mysql_query($query);
        close();

        if (mysql_num_rows($result) == UNIT)
            $info = mysql_fetch_object($result);
        else
            $info = null;

        return $info;
    }

    /**
     * 学生を新規登録する
     * @$login : ログイン名
     * @$pass : パスワード
     * @$name : 氏名
     * @$class : クラス名
     */
    function addStudent($login, $pass, $name, $class) {
        global $time;

        $query = "INSERT INTO student(
                                login_name, passwd, name, clazz, register_time)
VALUES(
                                '$login',    '$pass',    '$name',    '$class',
                                '$time');"

        //echo($query);
        // 新規登録する
        connect();
        $result = mysql_query($query);
        close();

        return $result;
    }

    /**

```

```

* 学生の情報を更新する
* @$login : ログイン名
* @$pass : 新規パスワード
* @$name : 新規名前
* @$class : 新規クラス名
*/
function updateStudent($login, $newPass, $newName, $newClass) {
    $query = "UPDATE student SET
                                student.passwd = '$newPass',
                                student.name = '$newName',
                                student.clazz = '$newClass'
                                WHERE student.login_name = '$login' LIMIT 1;";

    // 学生の情報を更新する
    connect();
    $result = mysql_query($query);
    close();

    return $result;
}
?>

```

```
=====
```

```
CREATE TABLE answer (  
    answer_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    question_id INTEGER UNSIGNED NOT NULL,  
    student_login_name VARCHAR(20) NOT NULL,  
    answer TEXT NOT NULL,  
    answer_time DATETIME NOT NULL,  
    is_corrected BOOL NOT NULL,  
    PRIMARY KEY(answer_id),  
    INDEX answer_FKIndex1(student_login_name),  
    INDEX answer_FKIndex2(question_id)  
);
```

```
CREATE TABLE correction (  
    correction_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    correction_kind_id INTEGER UNSIGNED NOT NULL,  
    answer_id INTEGER UNSIGNED NOT NULL,  
    answer_comment TEXT NULL,  
    correction_spot TEXT NULL,  
    PRIMARY KEY(correction_id),  
    INDEX correction_FKIndex1(answer_id),  
    INDEX correction_FKIndex2(correction_kind_id)  
);
```

```
CREATE TABLE correction_kind (  
    correction_kind_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    correction_kind TEXT NOT NULL,  
    correction_kind_comment TEXT NULL,  
    PRIMARY KEY(correction_kind_id)  
);
```

```
CREATE TABLE question (  
    question_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    question TEXT NOT NULL,
```



```
correct_example TEXT NULL,  
question_order INTEGER NOT NULL,  
admin_visible BOOL NOT NULL,  
student_visible BOOL NOT NULL,  
PRIMARY KEY(question_id)  
);
```

```
CREATE TABLE student (  
    login_name VARCHAR(20),  
    passwd VARCHAR(20) NOT NULL,  
    name VARCHAR(45) NOT NULL,  
    clazz VARCHAR(45) NOT NULL,  
    register_time DATETIME NOT NULL,  
    PRIMARY KEY(login_name)  
);
```

```
INSERT INTO correction_kind(correction_kind, correction_kind_comment)  
VALUES('正解です', '正解です');  
INSERT INTO question(question, correct_example, admin_visible, student_visible)  
VALUES('!!この問題は削除して下さい', '!!この問題は削除して下さい', 1, 0);  
=====
```

```
DROP      TABLE      answer,      correction,      correction_kind,      question,  
student;=====
```

```
<?php
```

```
/**
```

```
 * InitializeDatabase.php
```

```
 * データベースを初期化する
```

```
 * Author 荒木
```

```
 * 修正 1 回目 川口
```

```
 * Created on 2007/01/10
```

```
 * Updated on 2007/01/12
```

```
 *
```

```
 * !! 注意 ! これを実行するとデータベースを初期化します。
```

```
 */
```

```
require_once ('../DBAccessors/DBAccessor.php');
```

```
echo "データベースを初期化します。(すでに入っているデータはすべて削除します) <BR>";
```

```
initializeDatabase();
```

```
echo "データベースを初期化しました。";
```

```
/**
```

```
 * データベースを初期化する
```

```
 */
```

```
function initializeDatabase() {
```

```
    executeQuery("DROP TABLE answer, correction, correction_kind, correction_object,  
question, student");
```

```
    executeQuery("CREATE TABLE answer (  
answer_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
question_id INTEGER UNSIGNED NOT NULL,  
student_login_name VARCHAR(20) NOT NULL,  
answer TEXT NOT NULL,  
answer_time DATETIME NOT NULL,
```

```
is_corrected BOOL NOT NULL,  
PRIMARY KEY(answer_id),  
INDEX answer_FKIndex1(student_login_name),  
INDEX answer_FKIndex2(question_id)  
)");
```

```
executeQuery("CREATE TABLE correction (  
correction_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
correction_kind_id INTEGER UNSIGNED NOT NULL,  
answer_comment TEXT NULL,  
correction_spot TEXT NULL,  
PRIMARY KEY(correction_id),  
INDEX correction_FKIndex2(correction_kind_id)  
)");
```

```
executeQuery("CREATE TABLE correction_kind (  
correction_kind_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
correction_kind TEXT NOT NULL,  
correction_kind_comment TEXT NULL,  
PRIMARY KEY(correction_kind_id)  
)");
```

```
executeQuery("CREATE TABLE correction_object (  
correction_object_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
correction_id INTEGER UNSIGNED NOT NULL,  
answer_id INTEGER UNSIGNED NOT NULL,  
PRIMARY KEY(correction_object_id, correction_id, answer_id),  
INDEX correction_object_FKIndex1(correction_id),  
INDEX correction_object_FKIndex2(answer_id)  
)");
```

```
executeQuery("CREATE TABLE question (  
question_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
question TEXT NOT NULL,  
correct_example TEXT NULL,  
admin_visible BOOL NOT NULL,
```

```

        student_visible BOOL NOT NULL,
        question_order INTEGER UNSIGNED NULL,
        PRIMARY KEY(question_id)
    );

```

```

executeQuery("CREATE TABLE student (
    login_name VARCHAR(20),
    passwd VARCHAR(20) NOT NULL,
    name VARCHAR(45) NOT NULL,
    clazz VARCHAR(45) NOT NULL,
    register_time DATETIME NOT NULL,
    PRIMARY KEY(login_name)
);

```

```

executeQuery("INSERT INTO correction_kind(correction_kind, correction_kind_comment)
VALUES('正解です', '正解です')");

```

```

executeQuery("INSERT INTO question(question, correct_example, admin_visible,
student_visible)
VALUES('この問題は番兵です。(シリアルナンバー問題の対応)', 'この問題は番兵
です.', 0, 0)");
}

```

```

function executeQuery($query) {
    connect();
    $result = mysql_query($query);
    close();
    echo $result;
    echo "<BR>";
}

```

?>

=====

```
?CREATE TABLE student(  
  login_name VARCHAR(20) primary key NOT NULL,  
  password VARCHAR(20) NOT NULL,  
  name VARCHAR(45) NOT NULL,  
  class VARCHAR(45) NOT NULL,  
  register_time DATETIME NOT NULL  
) default character set utf8;
```

```
CREATE TABLE answer(  
  answer_id INT auto_increment primary key NOT NULL,  
  answer TEXT NOT NULL,  
  question_id INT NOT NULL,  
  login_name VARCHAR(20) NOT NULL,  
  answer_time DATETIME NOT NULL  
) default character set utf8;
```

```
CREATE TABLE question(  
  question_id INT auto_increment primary key NOT NULL,  
  question TEXT NOT NULL,  
  right_example TEXT NOT NULL,  
  teacher_visible BOOLEAN NOT NULL,  
  student_visible BOOLEAN NOT NULL  
) default character set utf8;
```

```
CREATE TABLE correction(  
  correction_id INT auto_increment primary key NOT NULL,  
  right_answer BOOLEAN NOT NULL,  
  answer_comment TEXT,  
  range_start INT,  
  range_end INT,  
  answer_id INT NOT NULL,  
  kind_id INT  
) default character set utf8;
```

```
CREATE TABLE correction_kind(  
kind_id INT auto_increment primary key NOT NULL,  
correction_kind TEXT NOT NULL,  
kind_comment TEXT NOT NULL  
) default character set utf8;
```

```
INSERT INTO student(login_name, password, name, class, register_time) VALUES('t04263mk',  
'hoge hoge', '川口 将司', 'ベーシック A', '2006-11-26 12:34:56');  
INSERT INTO answer(answer_time, answer, login_name, question_id) VALUES('2006-11-26 23:45:12',  
'I'm walking in the park.', 't04263mk', 1);  
INSERT INTO question(question, right_example, student_visible, teacher_visible) VALUES('わたし  
は街を歩いています。', 'I'm walking in the town.', FALSE, TRUE);  
INSERT INTO correction(right_answer, answer_comment, range_start, range_end, answer_id, kind_id)  
VALUES(0, '何を言っているんだチミは', 2, 4, 1, 1);  
INSERT INTO correction_kind(correction_kind, kind_comment) VALUES('前置詞の位置が違います', '死ねばいいのに');
```

```
INSERT INTO student(login_name, password, name, class, register_time) VALUES('t04000yk',  
'yuuuuuta', '黒木 雄太', 'ベーシック B', '2006-11-26 15:36:50');  
INSERT INTO answer(answer_time, answer, login_name, question_id) VALUES('2006-11-26 23:45:12',  
'Hes walking in the park.', 't04000yk', 2);  
INSERT INTO question(question, right_example, student_visible, teacher_visible) VALUES('私は中  
国人です。', 'I'm Chinese.', TRUE, TRUE);  
INSERT INTO correction(right_answer, answer_comment, range_start, range_end, answer_id, kind_id)  
VALUES(0, 'おまえはアホか', 1, 5, 2, 2);  
INSERT INTO correction_kind(correction_kind, kind_comment) VALUES('漢字が日本語です', '氏ね');
```

```
INSERT INTO answer(answer_time, answer, login_name, question_id) VALUES('2006-12-24 23:59:59',  
'How are you?', 't04263mk', 3);  
INSERT INTO answer(answer_time, answer, login_name, question_id) VALUES('2007-01-31 23:55:12',  
'Go home.', 't04263mk', 6);  
INSERT INTO answer(answer_time, answer, login_name, question_id) VALUES('2009-01-31 20:55:12',  
'This is a hoge hoge sentence.', 't04263mk', 2);
```

```
INSERT INTO answer(answer_time, answer, login_name, question_id) VALUES('2007-01-31 23:55:12',  
'Yeah!', 't04263mk', 2);
```

```
INSERT INTO correction(right_answer, answer_comment, range_start, range_end, answer_id, kind_id)  
VALUES(1, '正解だ!', null, null, 6, null);
```

```
SELECT * FROM answer, question, correction, correction_kind  
        WHERE answer.question_id = question.question_id  
        AND answer.answer_id = correction.answer_id  
        AND correction.kind_id = correction_kind.kind_id  
        AND          answer.login_name          =  
't04263mk';=====
```



```
CREATE TABLE answer (  
    answer_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    question_id INTEGER UNSIGNED NOT NULL,  
    student_login_name VARCHAR(20) NOT NULL,  
    answer TEXT NOT NULL,  
    answer_time DATETIME NOT NULL,  
    is_corrected BOOL NOT NULL,  
    PRIMARY KEY(answer_id),  
    INDEX answer_FKIndex1(student_login_name),  
    INDEX answer_FKIndex2(question_id)  
);
```

```
CREATE TABLE correction (  
    correction_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    correction_kind_id INTEGER UNSIGNED NOT NULL,  
    answer_id INTEGER UNSIGNED NOT NULL,  
    answer_comment TEXT NULL,  
    correction_spot TEXT NULL,  
    PRIMARY KEY(correction_id),  
    INDEX correction_FKIndex1(answer_id),  
    INDEX correction_FKIndex2(correction_kind_id)  
);
```

```
CREATE TABLE correction_kind (  
    correction_kind_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    correction_kind TEXT NOT NULL,  
    correction_kind_comment TEXT NULL,  
    PRIMARY KEY(correction_kind_id)  
);
```

```
CREATE TABLE question (  
    question_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    question TEXT NOT NULL,
```

```
correct_example TEXT NULL,  
admin_visible BOOL NOT NULL,  
student_visible BOOL NOT NULL,  
PRIMARY KEY(question_id)  
);
```

```
CREATE TABLE student (  
    login_name VARCHAR(20) NOT NULL,  
    passwd VARCHAR(20) NOT NULL,  
    name VARCHAR(45) NOT NULL,  
    class VARCHAR(45) NOT NULL,  
    register_time DATETIME NOT NULL,  
    PRIMARY KEY(login_name)  
);
```

=====

```
CREATE TABLE answer (  
    answer_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    question_id INTEGER UNSIGNED NOT NULL,  
    student_login_name VARCHAR(20) NOT NULL,  
    answer TEXT NOT NULL,  
    answer_time DATETIME NOT NULL,  
    is_corrected BOOL NOT NULL,  
    PRIMARY KEY(answer_id),  
    INDEX answer_FKIndex1(student_login_name),  
    INDEX answer_FKIndex2(question_id)  
);
```

```
CREATE TABLE correction (  
    correction_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    correction_kind_id INTEGER UNSIGNED NOT NULL,  
    answer_comment TEXT NULL,  
    correction_spot TEXT NULL,  
    PRIMARY KEY(correction_id),  
    INDEX correction_FKIndex2(correction_kind_id)  
);
```

```
CREATE TABLE correction_kind (  
    correction_kind_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    correction_kind TEXT NOT NULL,  
    correction_kind_comment TEXT NULL,  
    PRIMARY KEY(correction_kind_id)  
);
```

```
CREATE TABLE correction_object (  
    correction_object_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,  
    correction_id INTEGER UNSIGNED NOT NULL,  
    answer_id INTEGER UNSIGNED NOT NULL,  
    PRIMARY KEY(correction_object_id, correction_id, answer_id),
```

```
INDEX correction_object_FKIndex1(correction_id),
INDEX correction_object_FKIndex2(answer_id)
);
```

```
CREATE TABLE question (
question_id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
question TEXT NOT NULL,
correct_example TEXT NULL,
admin_visible BOOL NOT NULL,
student_visible BOOL NOT NULL,
question_order INTEGER UNSIGNED NULL,
PRIMARY KEY(question_id)
);
```

```
CREATE TABLE student (
login_name VARCHAR(20) NOT NULL AUTO_INCREMENT,
passwd VARCHAR(20) NOT NULL,
name VARCHAR(45) NOT NULL,
clazz VARCHAR(45) NOT NULL,
register_time DATETIME NOT NULL,
PRIMARY KEY(login_name)
);
```

```
INSERT INTO correction_kind(correction_kind, correction_kind_comment)
```

```
<<<<<<< junjun_DB3.sql
```

```
VALUES('.正解です', '正解です');
```

```
INSERT INTO question(question, correct_example, admin_visible, student_visible)
```

```
VALUES('!!この問題は削除して下さい', '!!この問題は削除して下さい', 1, 0);
```

```
=====
```

```
VALUES('.正解です', '正解です');
```

```
>>>>>>> 1.3
```

```
=====
```

```
CREATE TABLE problem(  
  problem_id INTEGER auto_increment primary key NOT NULL,  
  question VARCHAR(255),  
  deleteflag INTEGER NOT NULL,  
  date VARCHAR(255)  
)default character set utf8;
```

```
CREATE TABLE unclassifiedanswer(  
  problem_id INTEGER NOT NULL,  
  answer_id INTEGER auto_increment primary key NOT NULL,  
  answer VARCHAR(255),  
  date VARCHAR(255)  
)default character set utf8;
```

```
CREATE TABLE correctanswer(  
  problem_id INTEGER NOT NULL,  
  answer_id INTEGER auto_increment primary key NOT NULL,  
  answer VARCHAR(255),  
  date VARCHAR(255)  
)default character set utf8;
```

```
CREATE TABLE incorrectanswer(  
  problem_id INTEGER NOT NULL,  
  answer_id INTEGER auto_increment primary key NOT NULL,  
  answer VARCHAR(255),  
  category_id INTEGER NOT NULL,  
  comment VARCHAR(255),  
  range_start VARCHAR(255),  
  range_end VARCHAR(255),  
  date VARCHAR(255)  
)default character set utf8;
```

```
CREATE TABLE category(  
  category_id INTEGER auto_increment primary key NOT NULL,  
  category_name VARCHAR(255),  
  date VARCHAR(255)  
)default character set utf8;
```

```
category_id INTEGER auto_increment primary key NOT NULL,  
category VARCHAR(255),  
content VARCHAR(255),  
date VARCHAR(255)  
)default character set utf8;
```

```
INSERT INTO problem(question,deleteflag,date) VALUES('わたしは街を歩いています。',0,'2006 年 06  
月 10 日');
```

```
INSERT INTO problem(question,deleteflag,date) VALUES('わたしは中国語の先生です。',0,'2006 年 06  
月 10 日');
```

```
INSERT INTO problem(question,deleteflag,date) VALUES('あなたのお父さんの名前はなんですか？  
,0,'2006 年 06 月 10 日');
```

```
INSERT INTO problem(question,deleteflag,date) VALUES('あなたは学生ですか？',0,'2006 年 06 月 10  
日');
```

```
INSERT INTO problem(question,deleteflag,date) VALUES('昨日はなにをしていましたか？',0,'2006 年  
06 月 10 日');=====
```

```
<html lang="ja">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>ユーザ新規登録画面</title>
</head>
<BODY>
<center>

<h1>
新規登録画面
</h1>
<table>
<form action="AddNewUserCheck.php" method="post">
  <tr>
    <td>ログイン名</td>
    <td><input type="text" style="ime-mode: disabled;" name="login" size="30"></td>
  </tr>
  <tr>
    <td>名前</td>
    <td><input type="text" name="name" size="30"></td>
  </tr>
  <tr>
    <td>履修クラス</td>
    <td><select name="class">
<option value="A">A</option>
<option value="B">B</option>
<option value="C">C</option>
<option value="D">D</option>
</select></td>
  </tr>
<tr>
```

```
<td>パスワード</td>
<td><input type="password" name="password" size="30"></td>
</tr>

<tr>
<td>もう一度</td>
<td><input type="password" name="againPassword" size="30"></td>
</tr>

</table>
<input type="reset" value="リセット">

<input type="submit" value="登録">

</form>
<form>
<input type="button" onClick="location.href='login.php'" value="戻る">
</form>
</center>

</body>
</html>
```

=====


```
<?php
```

```
/*
```

```
 * Created on 2006/11/26
```

```
 *
```

```
 * To change the template for this generated file go to
```

```
 * Window - Preferences - PHPeclipse - PHP - Code Templates
```

```
 */
```

```
require_once ('../DBAccessors/StudentData.php');
```

```
$emptyChecker = false;
```

```
$passwordChecker = false;
```

```
if (isset ($_POST["login"])) {
```

```
    $newLogin = $_POST["login"];
```

```
    $newName = $_POST["name"];
```

```
    $newClass = $_POST["class"];
```

```
    $newPassword = $_POST["password"];
```

```
    $againPassword = $_POST["againPassword"];
```

```
    if (empty ($newLogin) || empty ($newName) || empty ($newClass) || empty ($newPassword)
```

```
|| empty ($againPassword)) {
```

```
        $emptyChecker = true;
```

```
    } else {
```

```
        if ($newPassword != $againPassword) {
```

```
            $passwordChecker = true;
```

```
        }
```

```
    }
```

```
}
```

```
?>
```

```
<html lang="ja">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>登録確認画面</title>
```

```
</head>
```

```
<BODY>
```

```
<?php
```

```
if ($emptyChecker) {
```

```
?>
```

```
<center>
```

```
<h1>エラー</h1>
```

```
<strong>
```

```
<p>
```

```
  入力が不完全です。すべての項目を入力してください。
```

```
</p>
```

```
</strong>
```

```
<form>
```

```
<input type="button" onClick="location.href='AddNewUser.php'"value="戻る">
```

```
</form>
```

```
</center>
```

```
<?php
```

```
} else
```

```
    if ($passwordChecker) {
```

```
?>
```

```
<center>
```

```
<h1>エラー</h1>
```

```
<strong>
```

```
<p>
```

```
  パスワードをもう一度入力しなおしてください。
```

```
</p>
```

```
</strong>
```

```
<form>
```

```
<input type="button" onClick="location.href='AddNewUser.php'"value="戻る">
```

```
</form>
```

```
</center>
```

```
<?php
```

```
    } else {
```

```
        if (isset ($_POST["login"])) {
```

```
?>
```

```
<center>
```

```
<h1>登録確認画面</h1>
```

```
<strong>
```

```
    <table>
```

```
        <tr>
```

```
            <td>
```

```
                <p>ログイン名</p>
```

```
            </td>
```

```
            <td>
```

```
                <?php print $newLogin;?>
```

```
            </td>
```

```
        </tr>
```

```
        <tr>
```

```
            <td>
```

```
                <p>名前</p>
```

```
            </td>
```

```
            <td>
```

```
                <?php print $newName;?>
```

```
            </td>
```

```
        </tr>
```

```
        <tr>
```

```
            <td>
```

```
                <p>履修クラス</p>
```

```
            </td>
```

```
            <td>
```

```
                <?php print $newClass;?>
```

```
            </td>
```

```
        </tr>
```

```
        <tr>
```

```
            <td>
```

```

        <p>パスワード</p>
    </td>
    <td>
        <?php

            $passwordVolume = strlen($newPassword);
            for ($i = 0; $i < $passwordVolume; $i ++) {
                print "**";
            }

?>

    </td>
</tr>
<tr>
    <td>
        <form action="AddNewUserCompleted.php" method="post">
            <input type="hidden" name="newLogin" value ="<?=$_POST["login"]?>">
            <input type="hidden" name="newPassword" value ="<?=$_POST["password"]?>">
            <input type="hidden" name="newName" value ="<?=$_POST["name"]?>">
            <input type="hidden" name="newClass" value ="<?=$_POST["class"]?>">
            <input
                                                                    type="submit"
onClick="location.href='AddNewUserCompleted.php'"value="登録">
        </form>
    </td>
    <td>
        <form>
            <input type="button" onClick="location.href='AddNewUser.php'" value="キャンセル">
        </form>
    </td>
</tr>
</table>
</strong>
</center>

```

```
<?php
```

```
    } else {
```

```
?>
```

```
        <center>
```

```
        <h1>エラー</h1>
```

```
        <strong>
```

```
        <p>
```

```
        ログイン画面からやり直してください。
```

```
        </p>
```

```
        </strong>
```

```
        <form>
```

```
        <input type="button" onClick="location.href='Login.php'" value="戻
```

```
        る">
```

```
        </form>
```

```
        </center>
```

```
<?php
```

```
    }
```

```
}
```

```
?>
```

```
</body>
```

```
</html>=====
```

```
<?php
```

```
/*
```

```
 * Created on 2006/11/27
```

```
 *
```

```
 * To change the template for this generated file go to
```

```
 * Window - Preferences - PHPeclipse - PHP - Code Templates
```

```
 */
```

```
require_once ('../DBAccessors/StudentData.php');
```

```
?>
```

```
<html lang="ja">
```

```
  <head>
```

```
    <meta http-equiv="Content-Type" content="text/html"; charset="UTF-8">
```

```
    <title>データ送信</title>
```

```
  </head>
```

```
<BODY>
```

```
<center>
```

```
<?
```

```
if (isset ($_POST["newLogin"]) && isset ($_POST["newPassword"]) && isset ($_POST["newName"]) &&
isset ($_POST["newClass"])) {
```

```
    if (addStudent($_POST["newLogin"], $_POST["newPassword"], $_POST["newName"],
$_POST["newClass"])) {
```

```
?>
```

```
<h1>送信しました。</h1>
```

```
<?
```

```
    } else {
```

```
?>
```

```
<h1>エラー：データを登録できませんでした。</h1>
```

```
<?
}

} else {
?>
    <h1>最初からやり直してください。</h1>
<?}?>
<form>
    <input type="button" onClick="location.href='Login.php'"value="TOP にもどる
">
</form>
</center>
</body>
</html>=====
```

```

<?php
require_once ('../DBAccessors/QuestionData.php');
//生徒が解くことの出来る問題の配列を取得する
$visibleQuestions = getSelectedQuestion();
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>中国語研究室 SystemJUN ver.STUDENT</title>
</head>

<body>
<?php

if (!isset ($_POST['questionNumber'])) {
    print "<h1>問題を選択してください。 </h1>";
    print "<form action='AnswerSelectQuestion.php'>
        <input type='submit' value='問題解答画面に戻る'>
    </form>";

} else {
    print "<h1 align='center'>問題</h1><br><br>";
    print "<h3>次の日本語を中国語に訳しなさい。 </h3>";
    $questionNumber = $_POST["questionNumber"];
    $question = $visibleQuestions[$questionNumber] ->question;
    print "<h2> 「".$question."」 </h2>";
?>

<font color=red>
<h5>、。などは全角で入力してください</h5>
<h5>かっこは「」を用いてください</h5>
<h5>アルファベットは半角で入力してください</h5>

```


<form method="post" action="AnswerCheck.php">

<input type="text" name="answer" size="50">

<input type="hidden" name="question" value="<?= \$question ?>">

<input type="hidden" name="questionNumber" value="<?= \$questionNumber?>">

<input type="submit" value="決定">

</form>

<form action="AnswerSelectQuestion.php">

<input type="submit" value="戻る">

</form>

<?php

}

?>

</body>

</html>=====

```
<?php
```

```
//セッションを開始
```

```
session_start();
```

```
require_once ('../DBAccessors/AnswerData.php');
```

```
require_once ('../DBAccessors/QuestionData.php');
```

```
//生徒が解くことの出来る問題の配列を取得する
```

```
$visibleQuestions = getSelectedQuestion();
```

```
/*
```

```
echo $questionNumber;
```

```
echo $visibleQuestions[$questionNumber] ->question_id;
```

```
echo $_POST["answer"];
```

```
echo $_SESSION["currentLogin"]
```

```
*/
```

```
?>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/ >
```

```
<title>中国語研究室 SystemJUN ver.STUDENT</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
if (isset ($_SESSION["currentLogin"])) {
```

```
    if (isset ($_POST["answer"]) && isset ($_POST["questionNumber"])) {
```

```
        $questionNumber = $_POST["questionNumber"];
```

```
        //新しい回答を追加する
```

```
        if
```

```
            (addAnswer($_POST["answer"],
```

```
$visibleQuestions[$questionNumber] ->question_id, $_SESSION["currentLogin"])) {
```

```
?>
```

```
<h2 align="center">回答を送信しました</h2>
```

```

<?}else{?>
    <h2 align="center">エラー：回答を送信できませんでした。</h2>
<?}?>
<center>
<form action="AnswerSelectQuestion.php">
<input type="submit" value="問題解答画面に戻る">
</form>
</center>
<?

        } else {
?>

        <center>
        <h2>最初からやり直してください。</h2>
        <form action="AnswerSelectQuestion.php">
        <input type="submit" value="問題解答画面に戻る">
        </form>
        </center>
        <?

        }
        } else {
?>
<center>
<h2>ログイン画面からやり直してください</h2>
<form action="Login.php">
<input type="submit" value="TOP に戻る">
</form>
</center>

<?}?>

</body>
</html>=====

```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>中国語研究室 SystemJUN ver.STUDENT</title>
</head>

<body>

<?php
if (!empty ($_POST["answer"])) {
    $answer = $_POST["answer"];
    $answer = mb_convert_kana($answer, "AKNrS", "UTF-8");

    $question = $_POST["question"];
    $questionNumber = $_POST["questionNumber"];

    print "<h1>".$answer."</h1><br>";
    print "これでいいですか?";
?>
<form method="post" action="AnswerAddComplete.php">
<input type="hidden" name="questionNumber" value="<?= $questionNumber?>">
<input type="hidden" name="answer" value="<?= $answer ?>"><br>
<input type="submit" value="決定">
</form>

<?

} else {
?>
<h1>回答を入力してください</h1>
<?
```

```
        if (isset ($_POST["questionNumber"])) {
            $question = $_POST["question"];
            $questionNumber = $_POST["questionNumber"];
        }
    }
?>
<form method="post" action="Answer.php">
<input type="hidden" name="questionNumber" value="<?= $questionNumber ?>">
<input type="submit" value="戻る">
</form>

</body>
</html>=====
```

```

?<?php
require_once ('../DBAccessors/QuestionData.php');

//生徒が解くことの出来る問題の配列を取得する
$visibleQuestions = getSelectedQuestion();
//生徒が解くことの出来る問題数を取得する
$questionVolume = count($visibleQuestions);
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>中国語研究室 SystemJUN ver.STUDENT</title>
</head>
<body>

<h1 align="center">中国語 JUNSystem ver.Student</h1>
<h3 align="center">順番に問題を解いていってください。</h3>
<h4 align="center">この問題は和文中訳です。日本語の文章を中国語で書いてもらいます。</h4>
<h4 align="center">中国語の入力の仕方はマニュアルを参考にしてください。</h4>
<br>
<br>
<table align="center">
<tr>
<td>
<form method="post" action="Answer.php">
<?

for ($i = 0; $i < $questionVolume; $i++) {
    $question = $visibleQuestions[$i] ->question;
?>
<input type="radio" name="questionNumber" value="<?=$i?>">
<?echo $question;?>

```


<?}?>

<input type='submit' value='問題に解答する。'>

</form>

<form>

<input type="button" onClick="location.href='StudentMain.php'" value=" TOP に戻る ">

</form>

</td>

</tr>

</table>

</body>

</html>=====

```
<?php

/*
 * Created on 2006/12/07
 *
 * To change the template for this generated file go to
 * Window - Preferences - PHPeclipse - PHP - Code Templates
 */
session_start(); //セッションを開始する。
?>
<html lang="ja">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>ユーザ情報変更画面</title>
</head>
<BODY>
<center>
<h1>ユーザ情報変更画面</h1>
<?

if (isset ($_SESSION["currentLogin"])) {
?>
<h3>変更したい情報のみ記入してください。</h3>
<form action="ChangeUserInfoCheck.php" method="post">
<table>
<tr>
<td>ログイン名</td>
<td><p><?print $_SESSION["currentLogin"]?></p></td>
</tr>
<tr>
<td>名前</td>
<td><input type="text" name="name" size="30"></td>
</tr>
```



```

<tr>
  <td>履修クラス</td>
  <td><select name="class">
<option value=""></option>
<option value="A">A</option>
<option value="B">B</option>
<option value="C">C</option>
<option value="D">D</option>
</select></td>
</tr>
<tr>
  <td>パスワード</td>
  <td><input type="password" name="password" size="30"></td>
</tr>

<tr>
  <td>もう一度</td>
  <td><input type="password" name="againPassword" size="30"></td>
</tr>

</table>
<input type="reset" value="リセット">

<input type="submit" value="登録">

</form>
<form>
<input type="button" onClick="location.href='StudentMain.php'" value="戻る">
<?}else{?>
  <center>
    <h3>ログイン画面からやり直してください。</h3>
    <form>
      <input type="button" onClick="location.href='Login.php'" value="ログイン画面へ戻る">
    </form>
  </center>
<?}}?>

```

</form>

</center>

</body>

</html>=====

```
<?php
```

```
/*
 * Created on 2006/12/07
 *
 * To change the template for this generated file go to
 * Window - Preferences - PHPeclipse - PHP - Code Templates
 */
session_start(); //セッションを開始する
require_once ('../DBAccessors/StudentData.php');
$passwordChecker = true; //パスワードが正しく入力されているかどうかを調べる変数
//パスワードが入力されていたら正しい入力かどうか調べる
if (!empty ($_POST["password"])) {
    if (empty ($_POST["againPassword"])) {
        $passwordChecker = false;
    } else {
        //入力された二つのパスワードが違うとき
        if ($_POST["password"] != $_POST["againPassword"]) {
            $passwordChecker = false;
        }
    }
}
?>
<html lang="ja">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>変更確認画面</title>
</head>
<BODY>
<center>
<h1>変更確認画面</h1>
<?>
```

```

if (isset ($_SESSION["currentLogin"])) {
    $currentUserInformation = getStudentRegister($_SESSION["currentLogin"]);
    //パスワードが正しい入力かどうか
    if ($passwordChecker) {
?>

        <table>

        <tr>
        <td>ログイン名</td>
        <td><p><?print $_SESSION["currentLogin"]?></p></td>
        </tr>
        <?if(!empty($_POST["name"])){?>
        <tr>
        <td>名前</td>
        <td><?print $_POST["name"]?></td>
        </tr>
    <?

        }
        if (!empty ($_POST["class"])) {
?>

            <tr>
            <td>履修クラス</td>
            <td><?print $_POST["class"]?></td>
            </tr>
        <?

            }
            if (!empty ($_POST["password"])) {
?>

                <tr>

```

```

        <td>パスワード</td>
        <td><?

//パスワードは*で表示させる。
$passwordVolume = strlen($_POST["password"]);
for ($i = 0; $i < $passwordVolume; $i++) {
    print "*";
}
?>    </td>
    </tr>

<?}?>

</table>
<form action="ChangeUserInformationCompleted.php" method="post">
    <input          type="hidden"          name="newLogin"          value
="<?=$_SESSION["currentLogin"]?>">
    <?if(!empty($_POST["password"])){?>
    <input type="hidden" name="newPassword" value ="<?=$_POST["password"]?>">
    <?}else{?>
        <input          type="hidden"          name="newPassword"          value
="<?=$_currentUserInformation->passwd?>">
        <?}?>
    <?if(!empty($_POST["name"])){?>
    <input type="hidden" name="newName" value ="<?=$_POST["name"]?>">
    <?}else{?>
        <input          type="hidden"          name="newName"          value
="<?=$_currentUserInformation->name?>">
        <?}?>
    <?if(!empty($_POST["class"])){?>
    <input type="hidden" name="newClass" value ="<?=$_POST["class"]?>">
    <?}else{?>
        <input          type="hidden"          name="newClass"          value
="<?=$_currentUserInformation->clazz?>">
        <?}?>

```

```

        <input                                     type="submit"
onClick="location.href='ChangeUserInfoCompleted.php'" value="登録">
    </form>

    <form>
        <input  type="button"  onClick="location.href='ChangeUserInfo.php'"
value="キャンセル">
    </form>
</center>
<?

    } else {
?>

    <center>
    <h3>パスワードの入力が不正です。</h3>
    <form>
        <input type="button" onClick="location.href='ChangeUserInfoEnterPassword.php'"
value="戻る">
    </form>
    </center>
<?

    }
} else {
?>

    <center>
    <h3>ログイン画面からやり直してください。</h3>
    <form>
        <input type="button" onClick="location.href='Login.php'" value="ログイン画面へ戻る">
    </form>
    </center>

<?}?>
</body>
</html>=====

```

```
<?php
```

```
/*
```

```
 * Created on 2006/12/07
```

```
 *
```

```
 * To change the template for this generated file go to
```

```
 * Window - Preferences - PHPeclipse - PHP - Code Templates
```

```
 */
```

```
session_start(); //セッションを開始する
```

```
require_once ('../DBAccessors/StudentData.php');
```

```
?>
```

```
<html lang="ja">
```

```
  <head>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
    <title>データ送信</title>
```

```
  </head>
```

```
<BODY>
```

```
<center>
```

```
<?
```

```
if (isset ($_POST["newLogin"]) && isset ($_POST["newPassword"]) && isset ($_POST["newName"]) &&
isset ($_POST["newClass"])) {
```

```
    if (updateStudent($_POST["newLogin"], $_POST["newPassword"], $_POST["newName"],
$_POST["newClass"])) {
```

```
?>
```

```
        <h1>送信しました。 </h1>
```

```
    <?>else{?>
```

```
        <h1>エラー：データを登録できませんでした。 </h1>
```

```
    <?
```

```
    }
```

```
} else {
```

```
?>
```

```
<h1>最初からやり直してください。</h1>
```

```
<?}?>
```

```
<form>
```

```
<input type="button" onClick="location.href='StudentMain.php'"value="TOP に
```

```
もどる">
```

```
</form>
```

```
</center>
```

```
</body>
```

```
</html>
```

=====


```
<?php
```

```
/*
```

```
 * Created on 2006/12/07
```

```
 *
```

```
 * To change the template for this generated file go to
```

```
 * Window - Preferences - PHPeclipse - PHP - Code Templates
```

```
 */
```

```
session_start(); //セッションを開始する。
```

```
require_once ('../DBAccessors/StudentData.php');
```

```
$passwordErrorChecker = false;
```

```
if (isset ($_SESSION["currentLogin"])) {
```

```
    if (isset ($_POST["password"])) {
```

```
        if ($_POST["password"] == getPassword($_SESSION["currentLogin"])) {
```

```
            header("Location: ChangeUserInformation.php");
```

```
        } else {
```

```
            $passwordErrorChecker = true;
```

```
        }
```

```
    }
```

```
}
```

```
?>
```

```
<html lang="ja">
```

```
<head>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>ユーザ情報変更画面</title>
```

```
</head>
```

```
<BODY>
```

```
<center>
```

```
<h1>ユーザ情報変更画面</h1>
```

```
<strong>
```

```
<?php
```

```
if ($passwordErrorChecker) {
```

```
?>
```

```
<p>
```

```
パスワードが違います。
```

```
</p>
```

```
<?php
```

```
}
```

```
?>
```

```
</strong>
```

```
<form action="ChangeUserInfoEnterPassword.php" method="post">
```

```
<strong>
```

```
古いパスワード <input type="password" name="password" size="30">
```

```
</strong>
```

```
<br>
```

```
<br>
```

```
<input type="submit" name="changeButton" value="送信">
```

```
<br>
```

```
<br>
```

```
<input type="button" onClick="location.href='StudentMain.php'" value="戻る">
```

```
</form>
```

```
</center>
```

```
</body>
```

```
</html>
```

```
=====
```

```
<?php
```

```
//セッションを開始
```

```
session_start();
```

```
require_once ('../DBAccessors/StudentData.php');
```

```
function user_Checker($loginNumber, $password) {
```

```
    //$testLoginNumber = "t04322yk";
```

```
    //$testLoginPassword = "testdayo";
```

```
    $adminChecker = false;
```

```
    //if ($loginNumber == $testLoginNumber && $password == $testLoginPassword) {
```

```
    if (getPassword($loginNumber) != null) {
```

```
        if ($password == getPassword($loginNumber)) {
```

```
            $adminChecker = true;
```

```
            return $adminChecker;
```

```
        } else {
```

```
            return $adminChecker;
```

```
        }
```

```
    } else {
```

```
        return $adminChecker;
```

```
    }
```

```
}
```

```
$emptyChecker = false; //フォームに完全に入力されたかどうか調べる変数
```

```
if (isset ($_POST["login"])) {
```

```
    $currentLogin = $_POST["login"];
```

```
    $currentPassword = $_POST["password"];
```

```
    //html タグを無効にする
```

```
    $currentLogin = htmlspecialchars($currentLogin);
```

```
    $currentPassword = htmlspecialchars($currentPassword);
```

```
        if (empty ($currentLogin) || empty ($currentPassword)) {
            $emptyChecker = true;
        } else {
            if (user_Checker($currentLogin, $currentPassword)) {
                $_SESSION["currentLogin"] = $currentLogin;
                header("Location: StudentMain.php");
            } else {
                session_destroy();
                header("Location: LoginError.php");
            }
        }
    }
}
?>
<html lang="ja">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>ログイン画面</title>
</head>

<BODY>
<center>

<?php

if ($emptyChecker) {
?>
<strong>
<center>
<p>
    入力が不完全です。ログイン名とパスワードを入力してください。
</p>
</center>
</strong>
```

```
<?php
```

```
}
```

```
?>
```

```
<h1>ログイン画面</h1>
```

```
<form action="Login.php" method="post">
```

```
<table>
```

```
<tr>
```

```
<td>ログイン名</td>
```

```
<td><input type="text" style="ime-mode: disabled;" name="login" size="30"></td>
```

```
</tr>
```

```
<tr>
```

```
<td>パスワード</td>
```

```
<td><input type="password" name="password" size="30"></td>
```

```
</tr>
```

```
</table>
```

```
<br>
```

```
<br>
```

```
<input type="submit" name="loginButton" value=" login ">
```

```
</form>
```

```
<form>
```

```
<input type="button" onClick="location.href='AddNewUser.php'" value=" 新規登録 ">
```

```
</form>
```

```
</center>
```

```
</body>
```

```
</html>
```

```
=====
```

```
<?php
```

```
/*
```

```
 * Created on 2006/11/26
```

```
 *
```

```
 * To change the template for this generated file go to
```

```
 * Window - Preferences - PHPeclipse - PHP - Code Templates
```

```
 */
```

```
?>
```

```
<html lang="ja">
```

```
  <head>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
    <title>失敗</title>
```

```
  </head>
```

```
  <BODY>
```

```
    <center>
```

```
      <h2>ログイン名またはパスワードが違います。</h2>
```

```
      <form>
```

```
        <input type="button" onClick="location.href='Login.php'" value=" TOP にもどる ">
```

```
      </form>
```

```
    </center>
```

```
  </body>
```

```
</html>=====
```

```
<?php
```

```
/*
```

```
 * Created on 2006/11/27
```

```
 *
```

```
 * To change the template for this generated file go to
```

```
 * Window - Preferences - PHPeclipse - PHP - Code Templates
```

```
 */
```

```
session_start();
```

```
session_destroy();
```

```
?>
```

```
<html lang="ja">
```

```
  <head>
```

```
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
    <title>ログアウト</title>
```

```
  </head>
```

```
<BODY>
```

```
<center>
```

```
<h1>
```

```
ログアウトしました。
```

```
</h1>
```

```
<form>
```

```
<input type="button" onClick="location.href='login.php'" value=" TOP にもどる ">
```

```
</form>
```

```
</center>
```

```
</body>
```

```
</html>
```

```
=====
```

```
<?php
```

```
/*
```

```
 * Created on 2006/11/26
```

```
 *
```

```
 * To change the template for this generated file go to
```

```
 * Window - Preferences - PHPeclipse - PHP - Code Templates
```

```
 */
```

```
session_start(); //セッションを開始する。
```

```
require_once ('../DBAccessors/StudentData.php');
```

```
?>
```

```
<html lang="ja">
```

```
<head>
```

```
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
  <title>生徒メイン画面</title>
```

```
</head>
```

```
<BODY>
```

```
<?php
```

```
if (isset ($_SESSION["currentLogin"])) {
```

```
?>
```

```
<p>
```

```
  ようこそ<?php
```

```
    print getName($_SESSION["currentLogin"]);
```

```
?>さん。
```

```
</p>
```

```
<center>
```

```
<h1>生徒メイン画面</h1>
```



```

<table>
    <tr><td>
        <form>
            <input
                type="button"
                onClick="location.href='AnswerSelectQuestion.php'" value=" 問題を解く  ">
            </form>
        </td></tr>
        <tr><td>
            <form>
                <input
                    type="button"
                    onClick="location.href='ViewAnswerLog.php'" value=" 履歴を見る  ">
                </form>
            </td></tr>
        <tr><td>
            <form>
                <input type="button" onClick="location.href='logout.php'"
                value=" ログアウト  ">
            </form>
        </td></tr>
        <tr><td>
            <form>
                <input
                    type="button"
                    onClick="location.href='ChangeUserInformationEnterPassword.php'" value=" ユーザ情報変更  ">
                </form>
            </td></tr>

```

```

</table>

```

```

</center>

```

```

<?php

```

```

} else {

```

```

?>

```

```
<strong>
<center>
<h2>
ログインしていません。トップページからログインしてください。
</h2>
<form>
<input type="button" onClick="location.href='Login.php'" value=" TOP にもどる ">
</form>
</center>
</strong>
    <?php

}
?>
</body>
</html>=====
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```
<title>中国語研究室 SystemJUN ver.Shige</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
session_start(); //セッションを開始する。
```

```
require_once ("../DBAccessors/AnswerData.php");
```

```
require_once ("../Common/AnswerViewCreateTools.php");
```

```
$login = $_SESSION["currentLogin"];
```

```
echo "ログイン名：";
```

```
echo $_SESSION["currentLogin"];
```

```
echo "<BR>";
```

```
$anslog_list = getAnswerHistory($login);
```

```
if ($anslog_list) {
```

```
    make_buttons_to_view_five_answers($anslog_list);
```

```
    if (isset ($_POST['start_point'])) {
```

```
        print_anslog_list_by_answer_number($anslog_list, $_POST['start_point'], 5);
```

```
    } else {
```

```
        print "表示されているボタンを押してください。";
```

```
    }
```

```
} else {
```

```
    print "まだ添削された回答がありません。";
```

```
}
```

```
?>
```

</body>

</html>=====

```
<?
```

```
//セッション開始
```

```
session_start();
```

```
?>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>中国語研究室 SystemJUN ver.Shige</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
$correction_state = 0; //添削の状態
```

```
define("NEW_CORRECTION", "1");
```

```
define("ADD_CORRECTION", "2");
```

```
define("OVERWRITE_CORRECTION", "3");
```

```
$view_state = 0; //画面の状態
```

```
define("FROM_MENU_VIEW", "1");
```

```
define("FROM_DONE_VIEW", "2");
```

```
define("FROM_EDIT_VIEW", "3");
```

```
define("FROM_THIS_VIEW", "4");
```

```
//日付を取得する
```

```
$today = date('Y年m月d日');
```

```
//正解か不正解かを取得する
```

```
$correction_kind_id = $_POST['category'];
```

```
//echo $correction_kind_id;
```

```
if ($correction_kind_id == 1) {
```

```

        $correction_spot = NULL;
    } else {
        if (count($_POST["letters"]) > 0) {
            $correction_spot = join(",", $_POST["letters"]);
        } else {
            $correction_spot = "NOT_SET";
        }
    }
}
$answer_comment = $_POST['comment'];

include ("../Common/HTMLNormalizer.php");
require_once ("../DBAccessors/CorrectionData.php");
require_once ("../DBAccessors/MarkingData.php");
/*****
$answer_comment = normalize($answer_comment);
*****/
//セッションから変数を登録
$problem = $_SESSION['PROBLEM'];
$problem_id = $_SESSION['PROBLEM_ID'];
$answer = $_SESSION['ANSWER'];
$answer_id = $_SESSION['ANSWER_ID'];
$correction_state = $_SESSION['CORRECTION_STATE'];
$answer_set=$_SESSION['ANSWER_SET'];

$answer_ids = getAnswerfromQuestionIDandAnswer($answer, $problem_id);

if($correction_state==OVERWRITE_CORRECTION){
    //DB から添削データを消去する
    $correction_id=$answer_set->correction_id;
    echo("削除する ID:".$correction_id."<br>");
    deleteCorrectionLink($correction_id);
    addReCorrection($answer_id, $correction_kind_id, $correction_spot, $answer_comment);
}elseif($correction_state==ADD_CORRECTION){
    addReCorrection($answer_id, $correction_kind_id, $correction_spot, $answer_comment);
}else{
    addCorrection($answer_id, $correction_kind_id, $correction_spot, $answer_comment);
}

```

```
}
```

```
?>
```

```
<h2 align="center">添削が追加されました。</h2>
```

```
<table align="center">
```

```
<tr><td>
```

```
<form method='post' action="Correction_Judge.php">
```

```
<input type="submit" name= "same_answer" value="同じ回答をさらに添削する">
```

```
</form>
```

```
</td></tr>
```

```
<tr><td>
```

```
<form action="Correction_Judge.php">
```

```
<input type="submit" value="次の回答を添削する">
```

```
</form>
```

```
</td></tr>
```

```
<tr><td>
```

```
<form action="TeacherTop.php">
```

```
<input type="submit" value="TOP ページに戻る">
```

```
</form>
```

```
</td></tr>
```

```
</table>
```

```
</body>
```

```
</html>=====
```

```

<html><head><title>統計データ</title></head><body><table><tr>
<td>問題</td>
<td>学生番号</td>
<td>学生のクラス</td>
<td>学生の解答</td>
<td>間違いの種類</td>
<td>間違いに対するコメント</td>
<td>解答に対するコメント</td>
<td>訂正例</td>
</tr>
<tr>
<td>あなたの家はどこですか？</td>
<td>s00000aa</td>
<td>classA</td>
<td>I wish so.</td>
<td>文法（１）</td>
<td>前置詞の位置が違います</td>
<td>目的語の前置詞の位置が違います</td>
<td>?的家在?里？</td>
</tr>
<tr>
<td>あなたの家はどこですか？</td>
<td>s00000aa</td>
<td>classA</td>
<td>I wish so.</td>
<td>単語（１）</td>
<td>日本語の漢字を使っています</td>
<td>中国語をやり直して下さい</td>
<td>?的家在?里？</td>
</tr>
<tr>
<td>あなたの家はどこですか？</td>
<td>s11111bb</td>
<td>classA</td>
<td>He goes school.</td>
<td>文法（１）</td>

```


<td>前置詞の位置が違います</td>	
<td>副詞の前置詞の位置が違います</td>	
<td>?的家在?里?</td>	
</tr>	<tr>
<td>あなたの家はどこですか?</td>	
<td>s11111bb</td>	
<td>classA</td>	
<td>He goes school.</td>	
<td>単語（1）</td>	
<td>日本語の漢字を使っています</td>	
<td>もうほげほげですね</td>	
<td>?的家在?里?</td>	
</tr>	<tr>
<td>あなたの家はどこですか?</td>	
<td>s00000aa</td>	
<td>classA</td>	
<td>AAAAAAAAAA.</td>	
<td>単語（2）</td>	
<td>カタカナでピインを書いています</td>	
<td>Aばかりではつつこみどころがありません</td>	
<td>?的家在?里?</td>	
</tr>	<tr>
<td>あなたの家はどこですか?</td>	
<td>s00000aa</td>	
<td>classA</td>	
<td>BBBBBBBBBB.</td>	
<td>単語（1）</td>	
<td>日本語の漢字を使っています</td>	
<td>Bばかりではわかりません</td>	
<td>?的家在?里?</td>	
</tr>	<tr>
<td>あなたの家はどこですか?</td>	
<td>s00000aa</td>	
<td>classA</td>	
<td>CCCCCCCCCC.</td>	
<td>単語（1）</td>	

<td>日本語の漢字を使っています</td>
<td></td>
<td>?的家在?里?</td>
</tr> <tr>
<td>あなたの家はどこですか?</td>
<td>s00000aa</td>
<td>classA</td>
<td>DDDDDDDDDD.</td>
<td>単語（1）</td>
<td>日本語の漢字を使っています</td>
<td></td>
<td>?的家在?里?</td>
</tr> <tr>
<td>あなたの家はどこですか?</td>
<td>s11111bb</td>
<td>classA</td>
<td>EEEEEEEEEE.</td>
<td>単語（1）</td>
<td>日本語の漢字を使っています</td>
<td></td>
<td>?的家在?里?</td>
</tr> <tr>
<td>あなたの家はどこですか?</td>
<td>s11111bb</td>
<td>classA</td>
<td>FFFFFFFFFF.</td>
<td>単語（1）</td>
<td>日本語の漢字を使っています</td>
<td></td>
<td>?的家在?里?</td>
</tr> <tr>
<td>あなたの家はどこですか?</td>
<td>s11111bb</td>
<td>classA</td>
<td>GGGGGGGGGG.</td>
<td>単語（1）</td>

<td>日本語の漢字を使っています</td>
<td></td>
<td>?的家在?里?</td>
</tr> <tr>
<td>あなたの家はどこですか?</td>
<td>suika</td>
<td>B</td>
<td>私の家はどこでしょう??</td>
<td>単語(1)</td>
<td>日本語の漢字を使っています</td>
<td>日本語で質問に答えないでください。</td>
<td>?的家在?里?</td>
</tr> <tr>
<td>あなたの家はどこですか?</td>
<td>suika</td>
<td>B</td>
<td>どこでもいいでしょ</td>
<td>単語(1)</td>
<td>日本語の漢字を使っています</td>
<td>間違ってるんだけどどことはいえませんね</td>
<td>?的家在?里?</td>
</tr> <tr>
<td>あなたの家はどこですか?</td>
<td>suika</td>
<td>B</td>
<td>どこかな~</td>
<td>種類テスト1</td>
<td>コメントテスト1</td>
<td>1文字目と2文字目</td>
<td>?的家在?里?</td>
</tr> <tr>
<td>あなたの家はどこですか?</td>
<td>suika</td>
<td>B</td>
<td>ああああああああああ</td>
<td>正解</td>

<td>正解です。</td>	
<td>2－5</td>	
<td>?的家在?里?</td>	
</tr>	<tr>
<td>おはようございます。1</td>	
<td>suika</td>	
<td>B</td>	
<td>おはよう</td>	
<td>正解</td>	
<td>正解です。</td>	
<td>おはよう正解です</td>	
<td>ございますおはよう1</td>	
</tr>	<tr>
<td>おはようございます。1</td>	
<td>ringo</td>	
<td>A</td>	
<td>おはよう</td>	
<td>単語（1）</td>	
<td>日本語の漢字を使っています</td>	
<td>おはよう正解です2回目</td>	
<td>ございますおはよう1</td>	
</tr>	<tr>
<td>おはようございます。1</td>	
<td>ringo</td>	
<td>A</td>	
<td>おはよう</td>	
<td>正解</td>	
<td>正解です。</td>	
<td>おはよう正解です</td>	
<td>ございますおはよう1</td>	
</tr>	<tr>
<td>おはようございます2</td>	
<td>suika</td>	
<td>B</td>	
<td>おはおは2</td>	
<td>正解</td>	

<td>正解です。</td>
<td></td>
<td>ございますおはよう 2</td>
</tr> <tr>
<td>おはようございます 2</td>
<td>ringo</td>
<td>A</td>
<td>おはおっは 2</td>
<td>単語（1）</td>
<td>日本語の漢字を使っています</td>
<td>おはおっはこめんと</td>
<td>ございますおはよう 2</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s00000aa</td>
<td>classA</td>
<td>I am Hoge.</td>
<td>正解</td>
<td>正解です。</td>
<td>模範解答です</td>
<td>?好，初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s00000aa</td>
<td>classA</td>
<td>I like dogs.</td>
<td>文法（1）</td>
<td>前置詞の位置が違います</td>
<td>カタカナを使わないで下さい</td>
<td>?好，初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s11111bb</td>
<td>classA</td>
<td>She loves me.</td>
<td>単語（1）</td>

<td>日本語の漢字を使っています</td>
<td></td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s11111bb</td>
<td>classA</td>
<td>She loves me.</td>
<td>単語（2）</td>
<td>カタカナでピインを書いています</td>
<td>添削テスト1です</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s11111bb</td>
<td>classA</td>
<td>I dislike humans.</td>
<td>単語（1）</td>
<td>日本語の漢字を使っています</td>
<td>てんさくてすと3 4 4</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s11111bb</td>
<td>classA</td>
<td>I dislike humans.</td>
<td>単語（1）</td>
<td>日本語の漢字を使っています</td>
<td>てんさくてすと3 4 4</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s11111bb</td>
<td>classA</td>
<td>I dislike humans.</td>
<td>単語（1）</td>

<td>日本語の漢字を使っています</td>
<td>てんさくてすと3344</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s11111bb</td>
<td>classA</td>
<td>I dislike humans.</td>
<td>単語（2）</td>
<td>カタカナでピインを書いています</td>
<td>添削の追加テストふろむ完了画面です。</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s11111bb</td>
<td>classA</td>
<td>I dislike humans.</td>
<td>文法（2）</td>
<td>助詞が抜けています</td>
<td>添削のやりなおしてすと2です。</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s00000aa</td>
<td>classA</td>
<td>I am Hoge.</td>
<td>単語（2）</td>
<td>カタカナでピインを書いています</td>
<td>添削テスト2</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s00000aa</td>
<td>classA</td>
<td>I like dogs.</td>
<td>文法（2）</td>

<td>助詞が抜けています</td>
<td>添削テストの2 3時2 6分版です</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s00000aa</td>
<td>classA</td>
<td>I like dogs.</td>
<td>種類テスト 2</td>
<td>コメントテスト 2</td>
<td>添削テストの追加版 2 3時2 6分です</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s00000aa</td>
<td>classA</td>
<td>I wish so.</td>
<td>単語（1）</td>
<td>日本語の漢字を使っています</td>
<td>こめんと</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s11111bb</td>
<td>classA</td>
<td>He goes school.</td>
<td>単語（1）</td>
<td>日本語の漢字を使っています</td>
<td>間違いの位置を入力しない</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>s11111bb</td>
<td>classA</td>
<td>She loves me.</td>
<td>単語（1）</td>

<td>日本語の漢字を使っています</td>
<td></td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>suika</td>
<td>B</td>
<td>こんにしは、ははは。</td>
<td>単語（1）</td>
<td>日本語の漢字を使っています</td>
<td>論外です。</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>suika</td>
<td>B</td>
<td>はじめまして～よろしく</td>
<td>単語（1）</td>
<td>日本語の漢字を使っています</td>
<td>3 = 5 もじめ</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>suika</td>
<td>B</td>
<td>きここここ</td>
<td>文法（3）</td>
<td>動詞がありません</td>
<td>にににに</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。</td>
<td>suika</td>
<td>B</td>
<td>たすけて～</td>
<td>種類テスト 1</td>

<td>種類コメント 2 3 時 2 7 </td>
<td>3 </td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは、はじめまして。 </td>
<td>suika</td>
<td>B</td>
<td>どこもまちがいにしないでね</td>
<td>文法（1） </td>
<td>前置詞の位置が違います</td>
<td>どこもまちがいにしません</td>
<td>?好, 初次?面</td>
</tr> <tr>
<td>こんにちは 3 </td>
<td>suika</td>
<td>B</td>
<td>こんにちは</td>
<td>正解</td>
<td>正解です。 </td>
<td>こんにちは 1 </td>
<td>にちはこん 3 </td>
</tr> <tr>
<td>こんにちは 3 </td>
<td>ringo</td>
<td>A</td>
<td>こんにちは</td>
<td>正解</td>
<td>正解です。 </td>
<td>こんにちは 2 </td>
<td>にちはこん 3 </td>
</tr> <tr>
<td>こんにちは 3 </td>
<td>ringo</td>
<td>A</td>
<td>こんにちは</td>
<td>正解</td>

<td>正解です。</td>	
<td>こんにちは 1</td>	
<td>にちはこん 3</td>	
</tr>	<tr>
<td>さようならまた明日 4</td>	
<td>suika</td>	
<td>B</td>	
<td>さよなら 4</td>	
<td>文法（1）</td>	
<td>前置詞の位置が違います</td>	
<td>さよなら 4</td>	
<td>さようならまた明日 4</td>	
</tr>	<tr>
<td>さようならまた明日 4</td>	
<td>ringo</td>	
<td>A</td>	
<td>ばいばいまたあした 4</td>	
<td>正解</td>	
<td>正解です。</td>	
<td>ばいばいまたあした 4</td>	
<td>さようならまた明日 4</td>	
</tr>	<tr>
<td>さようならまた明日 4</td>	
<td>ringo</td>	
<td>A</td>	
<td>さようなら 4</td>	
<td>単語（1）</td>	
<td>日本語の漢字を使っています</td>	
<td>添削 1</td>	
<td>さようならまた明日 4</td>	
</tr>	<tr>
<td>さようならまた明日 4</td>	
<td>ringo</td>	
<td>A</td>	
<td>さようなら 4</td>	
<td>単語（1）</td>	

<td>日本語の漢字を使っています</td>

<td>添削 1 </td>

<td>さようならまた明日 4 </td>

</tr> <tr>

<td>問題 1 </td>

<td>suika</td>

<td>B</td>

<td>解答 1 </td>

<td>文法（2） </td>

<td>助詞が抜けています</td>

<td>やってみよう</td>

<td>かいとうれい 1 </td>

</tr> <tr>

<td>問題 1 </td>

<td>suika</td>

<td>B</td>

<td>解答 1 </td>

<td>単語（1） </td>

<td>日本語の漢字を使っています</td>

<td>できたかな？</td>

<td>かいとうれい 1 </td>

</tr> <tr>

<td>問題 1 </td>

<td>suika</td>

<td>B</td>

<td>解答 1 </td>

<td>単語（1） </td>

<td>日本語の漢字を使っています</td>

<td>できできたかな？</td>

<td>かいとうれい 1 </td>

</tr> <tr>

<td>問題 1 </td>

<td>ringo</td>

<td>A</td>

<td>解答 1 </td>

<td>文法（2） </td>

<td>助詞が抜けています</td>	
<td>やってみよう</td>	
<td>かいとうれい1</td>	
</tr>	<tr>
<td>問題1</td>	
<td>ringo</td>	
<td>A</td>	
<td>解答1</td>	
<td>単語（1）</td>	
<td>日本語の漢字を使っています</td>	
<td>できたかな？</td>	
<td>かいとうれい1</td>	
</tr>	<tr>
<td>問題1</td>	
<td>ringo</td>	
<td>A</td>	
<td>解答1</td>	
<td>単語（1）</td>	
<td>日本語の漢字を使っています</td>	
<td>できできたかな？</td>	
<td>かいとうれい1</td>	
</tr>	<tr>
<td>問題1</td>	
<td>suika</td>	
<td>B</td>	
<td>解答1 1</td>	
<td>構文（1）</td>	
<td>違う構文を使っています</td>	
<td>添削第一回目のテストです。</td>	
<td>かいとうれい1</td>	
</tr>	<tr>
<td>問題1</td>	
<td>suika</td>	
<td>B</td>	
<td>解答1 1</td>	
<td>単語（1）</td>	

<td>日本語の漢字を使っています</td>	
<td>添削第二回目のテストです。</td>	
<td>かいとうれい 1</td>	
</tr>	<tr>
<td>問題 1</td>	
<td>suika</td>	
<td>B</td>	
<td>解答 1 1</td>	
<td>単語（1）</td>	
<td>日本語の漢字を使っています</td>	
<td>添削代 3 回目のテストです</td>	
<td>かいとうれい 1</td>	
</tr>	<tr>
<td>問題 1</td>	
<td>ringo</td>	
<td>A</td>	
<td>解答 1 1</td>	
<td>単語（1）</td>	
<td>日本語の漢字を使っています</td>	
<td>添削代 3 回目のテストです</td>	
<td>かいとうれい 1</td>	
</tr>	<tr>
<td>問題 1</td>	
<td>suikaboushi</td>	
<td>B</td>	
<td>解答のテストです。</td>	
<td>種類テスト 1</td>	
<td>コメントテスト 1</td>	
<td>添削のテストです。</td>	
<td>かいとうれい 1</td>	
</tr>	<tr>
<td>本当にありがとう 5</td>	
<td>suika</td>	
<td>B</td>	
<td>ありがとう 5</td>	
<td>構文（1）</td>	

<td>違う構文を使っています</td>
<td>ありがと 5</td>
<td>ありがとうほんとうに 5</td>
</tr> <tr>
<td>本当にありがとう 5</td>
<td>ringo</td>
<td>A</td>
<td>ありりりり 5</td>
<td>単語（1）</td>
<td>日本語の漢字を使っています</td>
<td>ありり 5</td>
<td>ありがとうほんとうに 5</td>
</tr> <tr>
<td>私の財布をみかけませんでしたか？</td>
<td>suika</td>
<td>B</td>
<td>あなたの財布はみかけていません</td>
<td>正解</td>
<td>正解です。</td>
<td>日本語で質問にきたえてはいけません</td>
<td>没看?我的?包??</td>
</tr> <tr>
<td>私の財布をみかけませんでしたか？</td>
<td>suika</td>
<td>B</td>
<td>みかけていません</td>
<td>単語（1）</td>
<td>日本語の漢字を使っています</td>
<td>ダメです</td>
<td>没看?我的?包??</td>
</tr> <tr>
<td>私の財布をみかけませんでしたか？</td>
<td>s00000aa</td>
<td>classA</td>
<td>ぶぎゃーなんじゃね？</td>
<td>単語（1）</td>

<td>日本語の漢字を使っています</td>	
<td>あああ</td>	
<td>没看?我的?包??</td>	
</tr>	<tr>
<td>私の財布をみかけませんでしたか?</td>	
<td>s00000aa</td>	
<td>classA</td>	
<td>ぶぎゃーなんじゃね?</td>	
<td>単語（１）</td>	
<td>日本語の漢字を使っています</td>	
<td>ぶギヤー添削</td>	
<td>没看?我的?包??</td>	
</tr>	<tr>
<td>私の財布をみかけませんでしたか?</td>	
<td>s00000aa</td>	
<td>classA</td>	
<td>ぶぎゃーなんじゃね?</td>	
<td>単語（１）</td>	
<td>日本語の漢字を使っています</td>	
<td>あああ</td>	
<td>没看?我的?包??</td>	
</tr>	<tr>
<td>私の財布をみかけませんでしたか?</td>	
<td>s00000aa</td>	
<td>classA</td>	
<td>ぶぎゃーなんじゃね?</td>	
<td>単語（１）</td>	
<td>日本語の漢字を使っています</td>	
<td>あああ</td>	
<td>没看?我的?包??</td>	
</tr>	<tr>
<td>私の財布をみかけませんでしたか?</td>	
<td>s00000aa</td>	
<td>classA</td>	
<td>ぶぎゃーなんじゃね?</td>	
<td>単語（１）</td>	

<td>日本語の漢字を使っています</td>	
<td>あああ</td>	
<td>没看?我的?包??</td>	
</tr>	<tr>
<td>私の財布をみかけませんでしたか?</td>	
<td>s00000aa</td>	
<td>classA</td>	
<td>ぶぎゃーなんじゃね?</td>	
<td>単語（1）</td>	
<td>日本語の漢字を使っています</td>	
<td>あああ</td>	
<td>没看?我的?包??</td>	
</tr>	<tr>
<td>私の財布をみかけませんでしたか?</td>	
<td>s00000aa</td>	
<td>classA</td>	
<td>ぶぎゃーなんじゃね?</td>	
<td>単語（1）</td>	
<td>日本語の漢字を使っています</td>	
<td>あああ</td>	
<td>没看?我的?包??</td>	
</tr>	<tr>
<td>私はおいしいお店を知っています。</td>	
<td>suika</td>	
<td>B</td>	
<td>私はおいしいお店を知りません</td>	
<td>正解</td>	
<td>正解です。</td>	
<td>だめだめです</td>	
<td>我知道好吃的店。</td>	
</tr>	<tr>
<td>私はおいしいお店を知っています。</td>	
<td>suika</td>	
<td>B</td>	
<td>知らないです</td>	
<td>正解</td>	

<td>正解です。</td>	
<td>だめですよ</td>	
<td>我知道好吃的店。</td>	
</tr>	<tr>
<td>私はおいしいお店を知っています。</td>	
<td>suika</td>	
<td>B</td>	
<td>oooooooooooooooooooooooo</td>	
<td>正解</td>	
<td>正解です。</td>	
<td>llllllll</td>	
<td>我知道好吃的店。</td>	
</tr>	<tr>
<td>私はおいしいお店を知っています。</td>	
<td>suika</td>	
<td>B</td>	
<td>k k k k k k k k k k k k k k</td>	
<td>単語（1）</td>	
<td>日本語の漢字を使っています</td>	
<td>3－6</td>	
<td>我知道好吃的店。</td>	
</tr>	<tr>
<td>私はおいしいお店を知っています。</td>	
<td>suika</td>	
<td>B</td>	
<td>わわわわわわあわわわあっわ</td>	
<td>単語（1）</td>	
<td>日本語の漢字を使っています</td>	
<td>1－4</td>	
<td>我知道好吃的店。</td>	
</tr>	<tr>
<td>私はおいしいお店を知っています。</td>	
<td>suika</td>	
<td>B</td>	
<td>これでさいごだ！！！！！！</td>	
<td>文法（1）</td>	

<td>前置詞の位置が違います</td>

<td>さいごだ！</td>

<td>我知道好吃的店。</td>

</tr></table></body></html>=====

```

<html><head><title>統計データ</title></head><body><table><tr>
<td>問題</td>
<td>学生番号</td>
<td>学生のクラス</td>
<td>学生の解答</td>
<td>間違いの種類</td>
<td>間違いに対するコメント</td>
<td>解答に対するコメント</td>
<td>訂正例</td>
</tr>
<tr>
<td>問題 1</td>
<td>test2</td>
<td>A</td>
<td>回答 1</td>
<td>.正解です</td>
<td>正解です</td>
<td></td>
<td>回答 1</td>
</tr>
<tr>
<td>問題 1</td>
<td>test1</td>
<td>A</td>
<td>回答 1 2</td>
<td>.正解です</td>
<td>正解です</td>
<td>回答コメント 1</td>
<td>回答 1</td>
</tr>
<tr>
<td>問題 1</td>
<td>test1</td>
<td>A</td>
<td>回答 1</td>
<td>種類 1</td>

```

```
<td>種類コメントテスト 1</td>
<td>採点やりなおしコメント 1</td>
<td>回答 1</td>
</tr>          <tr>
<td>問題 1</td>
<td>test2</td>
<td>A</td>
<td>回答 1</td>
<td>種類 1</td>
<td>種類コメントテスト 1</td>
<td>採点やりなおしコメント 1</td>
<td>回答 1</td>
</tr>          <tr>
<td>問題 4</td>
<td>test1</td>
<td>A</td>
<td>回答 4</td>
<td>.正解です</td>
<td>正解です</td>
<td></td>
<td>回答 4</td>
</tr>          <tr>
<td>問題 4</td>
<td>test2</td>
<td>A</td>
<td>回答 4</td>
<td>.正解です</td>
<td>正解です</td>
<td></td>
<td>回答 4</td>
</tr>          <tr>
<td>問題 4</td>
<td>test2</td>
<td>A</td>
<td>回答 4</td>
<td>種類 1</td>
```

```
<td>種類コメントテスト 1</td>
<td>コメント 4</td>
<td>回答 4</td>
</tr>          <tr>
<td>問題 4</td>
<td>test1</td>
<td>A</td>
<td>回答 4</td>
<td>種類 2</td>
<td>種類 2 コメント</td>
<td>コメント追加 4</td>
<td>回答 4</td>
</tr>          <tr>
<td>問題 4</td>
<td>test2</td>
<td>A</td>
<td>回答 4</td>
<td>種類 2</td>
<td>種類 2 コメント</td>
<td>コメント追加 4</td>
<td>回答 4</td>
```

```
</tr></table></body></html>=====
```

```

<html><head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>統計データ</title></head><body><table><tr>
<td>問題</td>
<td>学生番号</td>
<td>学生のクラス</td>
<td>学生の解答</td>
<td>間違いの種類</td>
<td>間違いに対するコメント</td>
<td>解答に対するコメント</td>
<td>訂正例</td>
</tr>
<tr>
<td>問題 1</td>
<td>test1</td>
<td>A</td>
<td>回答 1</td>
<td>種類 1</td>
<td>同時添削テスト回答 1 2</td>
<td>添削修正コメント</td>
<td>回答 1</td>
</tr>
<tr>
<td>問題 1</td>
<td>test1</td>
<td>A</td>
<td>回答 1</td>
<td>チェックなし</td>
<td>チェックなしテスト追加（編集画面から）</td>
<td></td>
<td>回答 1</td>
</tr>
<tr>
<td>問題 1</td>
<td>test1</td>
<td>A</td>
<td>回答 1 2</td>

```

<td>種類 1 </td>	
<td>同時添削テスト回答 1 2 </td>	
<td>同時添削テスト回答 1 2 </td>	
<td>回答 1 </td>	
</tr>	<tr>
<td>問題 1 </td>	
<td>test2</td>	
<td>A</td>	
<td>回答 1 </td>	
<td>種類 1 </td>	
<td>同時添削テスト回答 1 2 </td>	
<td>添削修正コメント</td>	
<td>回答 1 </td>	
</tr>	<tr>
<td>問題 1 </td>	
<td>test2</td>	
<td>A</td>	
<td>回答 1 </td>	
<td>チェックなし</td>	
<td>チェックなしテスト追加（編集画面から） </td>	
<td></td>	
<td>回答 1 </td>	
</tr>	<tr>
<td>問題 1 </td>	
<td>test3</td>	
<td>B</td>	
<td>回答 1 </td>	
<td>種類 1 </td>	
<td>同時添削テスト回答 1 2 </td>	
<td>添削修正コメント</td>	
<td>回答 1 </td>	
</tr>	<tr>
<td>問題 1 </td>	
<td>test3</td>	
<td>B</td>	
<td>回答 1 </td>	


```
<td>チェックなし</td>
<td>チェックなしテスト追加（編集画面から）</td>
<td></td>
<td>回答 1</td>
</tr>          <tr>
<td>問題 4</td>
<td>test1</td>
<td>A</td>
<td>回答 4</td>
<td>.正解です</td>
<td>正解です</td>
<td></td>
<td>回答 4</td>
</tr>          <tr>
<td>問題 4</td>
<td>test2</td>
<td>A</td>
<td>回答 4</td>
<td>.正解です</td>
<td>正解です</td>
<td></td>
<td>回答 4</td>
```

```
</tr></table></body></html>=====
```

```
<?
```

```
//セッション開始
```

```
session_start();
```

```
?>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>中国語研究室 SystemJUN ver.Shige</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
$correction_state = 0; //添削の状態
```

```
define("NEW_CORRECTION", "1");
```

```
define("ADD_CORRECTION", "2");
```

```
define("OVERWRITE_CORRECTION", "3");
```

```
$view_state = 0; //画面の状態
```

```
define("FROM_MENU_VIEW", "1");
```

```
define("FROM_DONE_VIEW", "2");
```

```
define("FROM_EDIT_VIEW", "3");
```

```
define("FROM_THIS_VIEW", "4");
```

```
//日付を取得する
```

```
$today = date('Y年m月d日');
```

```
//正解か不正解かを取得する
```

```
$correction_kind_id = $_POST['category'];
```

```

//echo $correction_kind_id;
if ($correction_kind_id == 1) {
    $correction_spot = NULL;
} else {
    if (isset($_POST["letters"])) {
        if (count($_POST["letters"]) > 0) {
            $correction_spot = join(",", $_POST["letters"]);
        } else {
            $correction_spot = "NOT_SET";
        }
    } else {
        $correction_spot = "NOT_SET";
    }
}

$answer_comment = $_POST['comment'];

include ("../Common/HTMLNormalizer.php");
require_once ("../DBAccessors/CorrectionData.php");
require_once ("../DBAccessors/MarkingData.php");
/*****/
$answer_comment = normalize($answer_comment);
/*****/
//セッションから変数を登録
$problem = $_SESSION['PROBLEM'];
$problem_id = $_SESSION['PROBLEM_ID'];
$answer = $_SESSION['ANSWER'];
$answer_id = $_SESSION['ANSWER_ID'];
$correction_state = $_SESSION['CORRECTION_STATE'];
$answer_set = $_SESSION['ANSWER_SET'];

$answer_ids = getAnswerfromQuestionIDandAnswer($answer, $problem_id);

if ($correction_state == OVERWRITE_CORRECTION) {
    //DB から添削データを消去する
    $correction_id = $answer_set->correction_id;
    echo ("削除する ID:".$correction_id."<br>");
}

```

```

        deleteCorrectionLink($correction_id);
        addReCorrection($answer_id, $correction_kind_id, $correction_spot, $answer_comment);
    }
    elseif ($correction_state == ADD_CORRECTION) {
        addReCorrection($answer_id, $correction_kind_id, $correction_spot, $answer_comment);
    } else {
        addCorrection($answer_id, $correction_kind_id, $correction_spot, $answer_comment);
    }
    ?>
<h2 align="center">添削が追加されました。</h2>
<table align="center">
<tr><td>
<form method='post' action="Correction_Judge.php">
<input type="submit" name= "same_answer" value="同じ回答をさらに添削する">
</form>
</td></tr>
<tr><td>
<form method='post' action="Correction_Judge.php">
<input type="submit" name="new_correction" value="次の回答を添削する">
</form>
</td></tr>
<tr><td>
<form action="TeacherTop.php">
<input type="submit" value="TOP ページに戻る">
</form>
</td></tr>

</table>
</body>
</html>=====

```

```
<?
```

```
// セッションを開始。
session_start();
// セッションに変数を登録
$_SESSION['PROBLEM'];
$_SESSION['PROBLEM_ID'];
$_SESSION['ANSWER'];
$_SESSION['ANSWER_ID'];
$_SESSION['CORRECTION_STATE'];
$_SESSION['VIEW_STATE'];
?>
```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>
<body>
```

```
<?php
```

```
require_once ("../DBAccessors/MarkingData.php");
require_once ("../Common/HTMLNormalizer.php");
require_once ("../DBAccessors/CorrectionKindData.php");
require_once ("../DBAccessors/CorrectionData.php");
```

```
$correction_state = 0; //添削の状態
define("NEW_CORRECTION", "1");
define("ADD_CORRECTION", "2");
define("OVERWRITE_CORRECTION", "3");
```

```
$view_state = 0; //画面の状態
```

```

define("FROM_MENU_VIEW", "1");
define("FROM_DONE_VIEW", "2");
define("FROM_EDIT_VIEW", "3");
define("FROM_THIS_VIEW", "4");

/**
 * 添削の状態を取得する
 */
function getCorrectionAndViewState() {
    //CORRECTION_STATE の判定
    //状態セッションが空だったら、デフォルトの状態を入れる。
    if ($_SESSION['CORRECTION_STATE'] == null) {
        $_SESSION['CORRECTION_STATE'] = NEW_CORRECTION;
    }

    //状態遷移 hidden フラグが立っていたら、状態を遷移させる。
    if (isset ($_POST['same_answer'])) { //添削を追加する場合（添削完了画面より）
        $_SESSION['CORRECTION_STATE'] = ADD_CORRECTION;
    }
    elseif (isset ($_POST['answer_from_edit_marking'])) {
        if (isset ($_POST['redo'])) { //添削をやりなおす場合（添削編集画面より）

            $_SESSION['CORRECTION_STATE'] = OVERWRITE_CORRECTION;
        } else { //添削を追加する場合（添削編集画面より）
            $_SESSION['CORRECTION_STATE'] = ADD_CORRECTION;
        }
    }
}

//状態セッションから現在の状態を取得する。
$correction_state = $_SESSION['CORRECTION_STATE'];

//VIEW_STATE の判定(前バージョンとの整合性を保つために必要)
//状態セッションが空だったら、デフォルトの状態を入れる。
if ($_SESSION['VIEW_STATE'] == null) {
    $_SESSION['VIEW_STATE'] = FROM_MENU_VIEW;
}

```

```

//状態遷移 hidden フラグが立っていたら、状態を遷移させる。
if (isset ($_POST['same_answer'])) { //添削完了画面でフラグを渡された場合
    $_SESSION['VIEW_STATE'] = FROM_DONE_VIEW;
}
elseif (isset ($_POST['answer_from_edit_marking'])) { //添削編集画面でフラグを渡され
た場合
    if (isset ($_POST['redo'])) { //上書きの場合
        $_SESSION['VIEW_STATE'] = FROM_EDIT_VIEW;
    } else { //追加の場合
        $_SESSION['VIEW_STATE'] = FROM_EDIT_VIEW;
    }
}
elseif (isset ($_POST['add_category'])) {
    $_SESSION['VIEW_STATE'] = FROM_THIS_VIEW;
}

//状態セッションから現在の状態を取得する。
$view_state = $_SESSION['VIEW_STATE'];

$correction_view_state->correction_state = $correction_state;
$correction_view_state->view_state = $view_state;

return $correction_view_state;
}

```

```

$correction_view_state = getCorrectionAndViewState();
$view_state = $correction_view_state->view_state;
$correction_state = $correction_view_state->correction_state;

/**
 * 添削に必要な情報を取得する
 */
function getDataForCorrection($correction_state, $view_state) {
    $answer_set=$_SESSION['ANSWER_SET'];

```

```

        return $answer_set;
    }

    /**
     * 添削用データを DB から取得する
     */
    function getAnswerSetFromDB() {
        $answer_set = getUnCorrectedAnswerSet();
        return $answer_set;
    }

    /**
     * 添削用データを画面から取得する
     */
    function getAnswerSetFromView($view_state) {
        if ($view_state == FROM_EDIT_VIEW) { //編集画面から遷移した場合
            $answer_from_edit_marking = $_POST['answer_from_edit_marking'];
            $answer_id_from_edit_marking = $_POST['answer_id_from_edit_marking'];
            $question_from_edit_marking = $_POST['question_from_edit_marking'];
            $question_id_from_edit_marking = $_POST['question_id_from_edit_marking'];
            $correction_id_from_edit_marking =
$_POST['correction_id_from_edit_marking'];

            $answer_set->question = $question_from_edit_marking;
            $answer_set->question_id = $question_id_from_edit_marking;
            $answer_set->answer = $answer_from_edit_marking;
            $answer_set->answer_id = $answer_id_from_edit_marking;
            $answer_set->correction_id = $correction_id_from_edit_marking;

        } else
            if ($view_state == FROM_DONE_VIEW) { //完了画面から遷移した場合
                $answer_set->answer = $_SESSION['ANSWER'];
                $answer_set->answer_id = $_SESSION['ANSWER_ID'];
                $answer_set->question = $_SESSION['PROBLEM'];
                $answer_set->question_id = $_SESSION['PROBLEM_ID'];
            }
    }

```



```

        return $answer_set;
    }

/*****
 * 不正解の位置を入力するチェックボックスを表示する
 */
function make_checkboxed_answer($answer) {
    $size_of_answer = mb_strlen($answer); //問題文の文字数
    $checkbox_added_answer = "";
    $letter_pointer = 0;
    while ($letter_pointer < $size_of_answer) {
        $letter = mb_substr($answer, $letter_pointer, 1);
        $letter = $letter."<input type=\"checkbox\" name=\"letters[]\"
value=\"{$letter_pointer}\">\n";
        $checkbox_added_answer = $checkbox_added_answer.$letter;
        $letter_pointer ++;
    }
    print $checkbox_added_answer;
}

/*****
*画面遷移後の処理**
*****/

//日付を取得する
$today = date('Y年m月d日');
//添削と画面の状態を取得する
$correction_view_state = getCorrectionAndViewState();
//添削に必要な情報を取得する
$correction_state = $correction_view_state->correction_state;
$view_state = $correction_view_state->view_state;
$answer_set = getDataForCorrection($correction_state, $view_state);

$uneditedanswer = $answer_set->answer;
$uneditedanswerquestion = $answer_set->question;
$uneditedanswerquestionid = $answer_set->question_id;

```

?>

<h1 align="center">不正解の種類追加画面</h1>

<table align="center">

<form method='POST' action="Correction_Review.php">

<h4>不正解の種類を追加する</h4>

不正解の種類の名称を決めてください

例) 単語、文型、簡体字

種類名: <input type="text" name ="correction_kind" size="10">

その不正解をしたときに表示するコメントを決めてください

例) 単語の間違いです。

コメント: <input type="text" name ="correction_kind_comment" size="30">

<input type="submit" name ="add_category" value="追加する">

</form>

<?php

// 添削する問題と回答を保存する

\$_SESSION['PROBLEM'] = \$answer_set ->question;

\$_SESSION['PROBLEM_ID'] = \$answer_set ->question_id;

\$_SESSION['ANSWER'] = \$answer_set ->answer;

\$_SESSION['ANSWER_ID'] = \$answer_set ->answer_id;

if (\$uneditedanswer) {

 print "

 <hr>

 <h2>今添削している問題</h2>

 <h3>問題: \$uneditedanswerquestion</h3>

 <h3>回答: \$uneditedanswer</h3>


```
        ";
    } else {
        print "<h3>未処理の回答はありません</h3>";
        <form>
            <input      type="button"      onClick="location.href='TeacherTop.php'"
value="TOP ページに戻る">
        </form>
        ";
    }
}
?><br><br>
<form>
<input type="button" onClick="location.href='TeacherTop.php'" value="TOP ページに戻る">
</form>

</body>
</html>=====
```

<?

```
// セッションを開始。
session_start();
//// セッションに変数を登録
//$_SESSION['PROBLEM'];
//$_SESSION['PROBLEM_ID'];
//$_SESSION['ANSWER'];
//$_SESSION['ANSWER_ID'];
//$_SESSION['CORRECTION_ID'];
//$_SESSION['CORRECTION_STATE'];
//$_SESSION['VIEW_STATE'];
?>
```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>
<body>
```

<?php

```
require_once ("../DBAccessors/MarkingData.php");
require_once ("../Common/HTMLNormalizer.php");
require_once ("../DBAccessors/CorrectionKindData.php");
require_once ("../DBAccessors/CorrectionData.php");
```

```
$correction_state = 0; //添削の状態
define("NEW_CORRECTION", "1");
define("ADD_CORRECTION", "2");
```

```
define("OVERWRITE_CORRECTION", "3");
```

```
$view_state = 0; //画面の状態
```

```
define("FROM_MENU_VIEW", "1");
```

```
define("FROM_DONE_VIEW", "2");
```

```
define("FROM_EDIT_VIEW", "3");
```

```
define("FROM_THIS_VIEW", "4");
```

```
/**
```

```
 * 添削の状態を取得する
```

```
 */
```

```
function getCorrectionAndViewState() {
```

```
    //CORRECTION_STATE の判定
```

```
    //状態セッションが空だったら、デフォルトの状態を入れる。
```

```
    if (isset($_SESSION['CORRECTION_STATE']) == false) {
```

```
        $_SESSION['CORRECTION_STATE'] = NEW_CORRECTION;
```

```
    }
```

```
    //状態遷移 hidden フラグが立っていたら、状態を遷移させる。
```

```
    if (isset ($_POST['same_answer'])) { //添削を追加する場合（添削完了画面より）
```

```
        $_SESSION['CORRECTION_STATE'] = ADD_CORRECTION;
```

```
    }
```

```
    elseif (isset ($_POST['answer_from_edit_marking'])) {
```

```
        if (isset ($_POST['redo'])) { //添削をやりなおす場合（添削編集画面より）
```

```
            $_SESSION['CORRECTION_STATE'] = OVERWRITE_CORRECTION;
```

```
        } else { //添削を追加する場合（添削編集画面より）
```

```
            $_SESSION['CORRECTION_STATE'] = ADD_CORRECTION;
```

```
        }
```

```
    }elseif (isset($_POST['new_correction'])) {
```

```
        $_SESSION['CORRECTION_STATE'] = NEW_CORRECTION;
```

```
    }
```

```
    //状態セッションから現在の状態を取得する。
```

```
    $correction_state = $_SESSION['CORRECTION_STATE'];
```

```

//VIEW_STATE の判定(前バージョンとの整合性を保つために必要)
//状態セッションが空だったら、デフォルトの状態を入れる。
if (isset($_SESSION['VIEW_STATE']) == false) {
    $_SESSION['VIEW_STATE'] = FROM_MENU_VIEW;
}

//状態遷移 hidden フラグが立っていたら、状態を遷移させる。
if (isset ($_POST['same_answer'])) { //添削完了画面でフラグを渡された場合
    $_SESSION['VIEW_STATE'] = FROM_DONE_VIEW;
}
elseif (isset ($_POST['answer_from_edit_marking'])) { //添削編集画面でフラグを渡され
た場合
    if (isset ($_POST['redo'])) { //上書きの場合
        $_SESSION['VIEW_STATE'] = FROM_EDIT_VIEW;
    } else { //追加の場合
        $_SESSION['VIEW_STATE'] = FROM_EDIT_VIEW;
    }
}
elseif (isset ($_POST['add_category'])) {
    $_SESSION['VIEW_STATE'] = FROM_THIS_VIEW;
}elseif (isset($_POST['new_correction'])) {
    $_SESSION['VIEW_STATE'] = FROM_MENU_VIEW;
}

//状態セッションから現在の状態を取得する。
$view_state = $_SESSION['VIEW_STATE'];

$correction_view_state->correction_state = $correction_state;
$correction_view_state->view_state = $view_state;

return $correction_view_state;
}

$correction_view_state = getCorrectionAndViewState();
$view_state = $correction_view_state->view_state;
$correction_state = $correction_view_state->correction_state;

```

```

////画面と添削の状態をデバッグ用に表示する。
//echo $view_state;
//echo $correction_state;

/**
 * 添削に必要な情報を取得する
 */
function getDataForCorrection($correction_state, $view_state) {
    //添削の状態を調べて
    if ($correction_state == NEW_CORRECTION) { //添削がされていない場合は
        //DB からデータを読み込む
        $answer_set = getAnswerSetFromDB();
    } else
        if ($correction_state == OVERWRITE_CORRECTION) { //添削をやり直す場合は
            //画面からデータを読み込む
            $answer_set = getAnswerSetFromView($view_state);
            $answer_id = $answer_set >answer_id;
            $correction_id = $answer_set >correction_id;
//            //DB から添削データを消去する
//            deleteCorrection($correction_id, $answer_id);
        } else
            if ($correction_state == ADD_CORRECTION) { //添削を追加する場合は
                //画面からデータを読み込む
                $answer_set = getAnswerSetFromView($view_state);
            }
    return $answer_set;
}

/**
 * 添削用データを DB から取得する
 */
function getAnswerSetFromDB() {
    $answer_set = getUnCorrectedAnswerSet();
    return $answer_set;
}

```

```

/**
 * 添削用データを画面から取得する
 */
function getAnswerSetFromView($view_state) {
    $answer_set=null;
    if ($view_state == FROM_EDIT_VIEW) { //編集画面から遷移した場合
        $answer_from_edit_marking = $_POST['answer_from_edit_marking'];
        $answer_id_from_edit_marking = $_POST['answer_id_from_edit_marking'];
        $question_from_edit_marking = $_POST['question_from_edit_marking'];
        $question_id_from_edit_marking = $_POST['question_id_from_edit_marking'];
        $correction_id_from_edit_marking = $_POST['correction_id_from_edit_marking'];

        $answer_set->question = $question_from_edit_marking;
        $answer_set->question_id = $question_id_from_edit_marking;
        $answer_set->answer = $answer_from_edit_marking;
        $answer_set->answer_id = $answer_id_from_edit_marking;
        $answer_set->correction_id = $correction_id_from_edit_marking;

    } else
        if ($view_state == FROM_DONE_VIEW) { //完了画面から遷移した場合
            $answer_set->answer = $_SESSION['ANSWER'];
            $answer_set->answer_id = $_SESSION['ANSWER_ID'];
            $answer_set->question = $_SESSION['PROBLEM'];
            $answer_set->question_id = $_SESSION['PROBLEM_ID'];
        }
    return $answer_set;
}

/*****
*画面遷移後の処理**
*****/

//日付を取得する
$today = date('Y年m月d日');

```



```
//添削と画面の状態を取得する
$correction_view_state = getCorrectionAndViewState();
//添削に必要な情報を取得する
$correction_state = $correction_view_state ->correction_state;
$view_state = $correction_view_state ->view_state;
$answer_set = getDataForCorrection($correction_state, $view_state);
```

```
$uneditedanswer = $answer_set ->answer;
$uneditedanswerquestion = $answer_set ->question;
$uneditedanswerquestionid = $answer_set ->question_id;
?>
```

```
<h1 align="center">添削画面</h1><br><br>
<table align="center">
```

```
<?php
//添削する問題と回答を保存する
$_SESSION['ANSWER_SET']=$answer_set;
$_SESSION['PROBLEM'] = $answer_set ->question;
$_SESSION['PROBLEM_ID'] = $answer_set ->question_id;
$_SESSION['ANSWER'] = $answer_set ->answer;
$_SESSION['ANSWER_ID'] = $answer_set ->answer_id;
//if($answer_set ->correction_id){
//$_SESSION['CORRECTION_ID'] = $answer_set ->correction_id;
//}
```

```
$html =<<<EOD
```

```
<h4>まず、正解か不正解かを選んでください。</h4>
<form method="post" action="Correction_Completed.php" >
<input type="hidden" name="category" value="1">
<input type="hidden" name = "comment" value="">
<input type="submit" value="正解">
</form>
```

```
<form>
<input type="button" onClick="location.href='Correction_Review.php'" value="不正解">
```

```

</form>
<HR>
<form>
<input type="button" onClick="location.href='TeacherTop.php'" value="TOP ページに戻る">
</form>
EOD;

if ($uneditedanswer) {
    print "
        <h2>添削する回答</h2>
        <h3>問題文：$uneditedanswerquestion</h3>
        <h3>回答文：$uneditedanswer</h3>
        <br>
        ";
    print $html;
} else {
    print "<h3>未処理の回答はありません</h3>
        <form>
        <input type=\"button\"
onClick=\"location.href='TeacherTop.php'\" value=\"TOP ページに戻る\">
        </form>
        ";
}
?>

</body>
</html>=====

```

<?

```
// セッションを開始。
session_start();
// セッションに変数を登録
$_SESSION['PROBLEM'];
$_SESSION['PROBLEM_ID'];
$_SESSION['ANSWER'];
$_SESSION['ANSWER_ID'];
$_SESSION['CORRECTION_STATE'];
$_SESSION['VIEW_STATE'];
?>
```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>
<body>
```

<?php

```
require_once ("../DBAccessors/MarkingData.php");
require_once ("../Common/HTMLNormalizer.php");
require_once ("../DBAccessors/CorrectionKindData.php");
require_once ("../DBAccessors/CorrectionData.php");
```

```
$correction_state = 0; //添削の状態
define("NEW_CORRECTION", "1");
define("ADD_CORRECTION", "2");
define("OVERWRITE_CORRECTION", "3");
```

```
$view_state = 0; //画面の状態
```

```

define("FROM_MENU_VIEW", "1");
define("FROM_DONE_VIEW", "2");
define("FROM_EDIT_VIEW", "3");
define("FROM_THIS_VIEW", "4");

/**
 * 添削の状態を取得する
 */
function getCorrectionAndViewState() {
    //CORRECTION_STATE の判定
    //状態セッションが空だったら、デフォルトの状態を入れる。
    if ($_SESSION['CORRECTION_STATE'] == null) {
        $_SESSION['CORRECTION_STATE'] = NEW_CORRECTION;
    }

    //状態遷移 hidden フラグが立っていたら、状態を遷移させる。
    if (isset ($_POST['same_answer'])) { //添削を追加する場合（添削完了画面より）
        $_SESSION['CORRECTION_STATE'] = ADD_CORRECTION;
    }
    elseif (isset ($_POST['answer_from_edit_marking'])) {
        if (isset ($_POST['redo'])) { //添削をやりなおす場合（添削編集画面より）

            $_SESSION['CORRECTION_STATE'] = OVERWRITE_CORRECTION;
        } else { //添削を追加する場合（添削編集画面より）
            $_SESSION['CORRECTION_STATE'] = ADD_CORRECTION;
        }
    }
}

//状態セッションから現在の状態を取得する。
$correction_state = $_SESSION['CORRECTION_STATE'];

//VIEW_STATE の判定(前バージョンとの整合性を保つために必要)
//状態セッションが空だったら、デフォルトの状態を入れる。
if ($_SESSION['VIEW_STATE'] == null) {
    $_SESSION['VIEW_STATE'] = FROM_MENU_VIEW;
}

```

```

//状態遷移 hidden フラグが立っていたら、状態を遷移させる。
if (isset ($_POST['same_answer'])) { //添削完了画面でフラグを渡された場合
    $_SESSION['VIEW_STATE'] = FROM_DONE_VIEW;
}
elseif (isset ($_POST['answer_from_edit_marking'])) { //添削編集画面でフラグを渡され
た場合
    if (isset ($_POST['redo'])) { //上書きの場合
        $_SESSION['VIEW_STATE'] = FROM_EDIT_VIEW;
    } else { //追加の場合
        $_SESSION['VIEW_STATE'] = FROM_EDIT_VIEW;
    }
}
elseif (isset ($_POST['add_category'])) {
    $_SESSION['VIEW_STATE'] = FROM_THIS_VIEW;
}

//状態セッションから現在の状態を取得する。
$view_state = $_SESSION['VIEW_STATE'];

$correction_view_state->correction_state = $correction_state;
$correction_view_state->view_state = $view_state;

return $correction_view_state;
}

$correction_view_state = getCorrectionAndViewState();
$view_state = $correction_view_state->view_state;
$correction_state = $correction_view_state->correction_state;

////画面と添削の状態をデバッグ用に表示する。
//echo $view_state;
//echo $correction_state;

/**

```

```

* 添削に必要な情報を取得する
*/
function getDataForCorrection($correction_state, $view_state) {
    $answer_set=$_SESSION['ANSWER_SET'];
    return $answer_set;
}

/**
* 添削用データを DB から取得する
*/
function getAnswerSetFromDB() {
    $answer_set = getUnCorrectedAnswerSet();
    return $answer_set;
}

/**
* 添削用データを画面から取得する
*/

/*****
* 不正解の位置を入力するチェックボックスを表示する
*/
function make_checkboxed_answer($answer) {
    $size_of_answer = mb_strlen($answer); //問題文の文字数
    $checkbox_added_answer = "";
    $letter_pointer = 0;
    while ($letter_pointer < $size_of_answer) {
        $letter = mb_substr($answer, $letter_pointer, 1);
        $letter = $letter."<input type=\"checkbox\" name=\"letters[]\"
value=\"".$letter_pointer.">\n";
        $checkbox_added_answer = $checkbox_added_answer.$letter;
        $letter_pointer ++;
    }
    print $checkbox_added_answer;
}

```

```

/*****
*画面遷移後の処理**
*****/

//日付を取得する
$today = date('Y年 m 月 d 日');
//添削と画面の状態を取得する
$correction_view_state = getCorrectionAndViewState();
//添削に必要な情報を取得する
$correction_state = $correction_view_state ->correction_state;
$view_state = $correction_view_state ->view_state;
$answer_set = getDataForCorrection($correction_state, $view_state);

//添削の種類を追加するボタンが押されていたら
//添削の種類を追加する
if ($view_state == FROM_THIS_VIEW) {
    if (isset ($_POST['correction_kind']) && isset ($_POST['correction_kind_comment'])) {
        $correction_kind = $_POST['correction_kind'];
        $correction_kind_comment = $_POST['correction_kind_comment'];
        $correction_kind_comment = htmlspecialchars($correction_kind_comment);
        addCorrectionKind($correction_kind, $correction_kind_comment);
    }
}

$uneditedanswer = $answer_set ->answer;
$uneditedanswerquestion = $answer_set ->question;
$uneditedanswerquestionid = $answer_set ->question_id;
?>

<h1 align="center">不正解コメント入力画面</h1><br><br>
<table align="center">
<?php

//添削する問題と回答を保存する
$_SESSION['PROBLEM'] = $answer_set ->question;
$_SESSION['PROBLEM_ID'] = $answer_set ->question_id;

```

```

$_SESSION['ANSWER'] = $answer_set->answer;
$_SESSION['ANSWER_ID'] = $answer_set->answer_id;
if ($uneditedanswer) {
    print "
        <h2>添削する回答</h2>
        <h3>質問：$uneditedanswerquestion</h3>
        <h3>回答：$uneditedanswer</h3>
        <br>
        ";
} else {
    print "<h3>未処理の回答はありません</h3>
        <form>
            <input      type=\"button\"      onClick=\"location.href='TeacherTop.php'\"
value=\"TOP ページに戻る\">
        </form>
        ";
}
?>

```

<h4>不正解の種類と場所、回答に対するコメントを入力してください。</h4>

<form method="post" action="Correction_Completed.php" >

不正解の種類:

```

<select name = "category">
<?php

```

```

// (不正解) 種類 DB からデータを読み込む
$correction_kinds = getCorrectionKinds();
//(不正解)種類 DB から種類名を取り出し、配列に格納する
echo $correction_kinds;
foreach ($correction_kinds as $correction_kind) {
    $id = $correction_kind->correction_kind_id;
    $kind = $correction_kind->correction_kind;
    $contant = $correction_kind->correction_kind_comment;
    echo "<option value=\".$id.\">".$kind.":".$contant."</option><br>";
}

```


?>

</select>

<input type="button" onClick="location.href='Correction_CreateCategory.php'" value="不正解の種類を追加する">

不正解の場所：

<?php

make_checkboxed_answer(\$answer_set->answer);

?>

コメント：<input type="text" name = "comment" size="50">

<input type="submit" value="添削完了">

</form>

<HR>

添削編集画面を表示する

<form>

<input type="button" onClick="location.href='TeacherTop.php'" value="TOP ページに戻る">

</form>

</body>

</html>=====

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```
<title>中国語研究室 SystemJUN ver.Shige</title>
```

```
</head>
```

```
<body>
```

```
<?php
```

```
require_once ("../DBAccessors/QuestionData.php");
```

```
$checkedanswers_list = getExcelSources();
```

```
function create_html_data($checkedanswers_list) {
```

```
    $html_header =<<<EOK
```

```
<html><head><meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```
<title>統計データ</title></head><body>
```

```
EOK;
```

```
    $table_header =<<<EOD
```

```
<table><tr>
```

```
<td>問題</td>
```

```
<td>学生番号</td>
```

```
<td>学生のクラス</td>
```

```
<td>学生の解答</td>
```

```
<td>間違いの種類</td>
```

```
<td>間違いに対するコメント</td>
```

```
<td>解答に対するコメント</td>
```

```
<td>訂正例</td>
```

```
</tr>
```

```
EOD;
```

```
    $table_body = "";
```

```
    foreach ($checkedanswers_list as $checkedanswer) {
```

```
        $tr_body =<<<EOD
```

```

        <tr>
        <td>${checkedanswer }>problem</td>
        <td>${checkedanswer }>student</td>
        <td>${checkedanswer }>classname</td>
        <td>${checkedanswer }>answer</td>
        <td>${checkedanswer }>mistake</td>
        <td>${checkedanswer }>correction_kind_comment</td>
        <td>${checkedanswer }>answer_comment</td>
        <td>${checkedanswer }>correctans</td>
        </tr>

EOD;

        $table_body = $table_body.$tr_body;
    }
    $table_footer = "</table>";
    $html_footer = "</body></html>";

    $html_data = $html_header.$table_header.$table_body.$table_footer.$html_footer;
    return $html_data;
}

function create_csv_file($checkedanswers_list) {
    $download_day = date("Ymd");
    $csvfilename = $download_day."DATA.xls";
    echo "<a href='\".$csvfilename.\"'>\".$csvfilename.\"</a>";
    $html_data = create_html_data($checkedanswers_list);
    $html_data = mb_convert_encoding($html_data, "UTF-8");
    $fp = fopen($csvfilename, "w");
    fwrite($fp, $html_data);
    fclose($fp);
}

create_csv_file($checkedanswers_list);

echo "<form action='\" method='POST'><input type='submit' name='csvmaker' value='CSV ファイル
を作成する'>";
?>

</body></html>=====

```

```

?<html>
<head>
<title>中国語研究室 SystemJUN ver.Shige</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"></head>

<body>
<?php
require_once ("../DBAccessors/MarkingData.php");
require_once ("../Common/AnswerViewCreateTools.php");
define("CORRECTANS_ID", "1");
/*****
*画面遷移後の処理**
*****/

//選択された問題を取得する
$selected_question = $_POST['question'];
$selected_question_id = $_POST['question_id'];
?>

<h1 align="center">添削編集画面</h1><br><br>
問題：
<?php

//問題を表示する
print "$selected_question";
?>
<br><br><br>
<?php

/*****
****問題に対する解答をすべて表示し、それに対する添削を表示する****
*****/

//echo $selected_question_id;

```

```

$answers = getAllAnswersfromQuestionID($selected_question_id);
echo $answers;
$count = 1;
if ($answers) {
    foreach ($answers as $answer) {
        if (answerIsShown($answer->answer, $selected_question_id,
$answer->answer_id)) {
            //すでに表示済みだったらなにも表示しない
            return;

        } else {
            print "

            <h4>回答";

            print "$count";
            print " : ".$answer->answer."</h4>";
            echo $answer->answer_id;
            show_all_corrections_from_answer($answer->answer_id);
            $count ++;
        }
    }
} else {
    echo "まだ添削はありません";
}

```

```

function show_all_corrections_from_answer($answer_id) {
    $correction_sets = getAllCorrectionfromAnswerID($answer_id);
    foreach ($correction_sets as $correction_set) {
        $correction_kind = $correction_set->correction_kind;
        $correction_kind_id = $correction_set->correction_kind_id;
        $correction_spot = $correction_set->correction_spot;
        $answer = $correction_set->answer;
        $answer_id = $correction_set->answer_id;
        $answer_comment = $correction_set->answer_comment;
        $question = $correction_set->question;
        $question_id = $correction_set->question_id;
    }
}

```

```

$correction_id = $correction_set >correction_id;
print "添削$correction_id<br>

添削の種類：$correction_kind<br>";
show_correction_spot($correction_spot, $answer);
print "<br>

コメント：$answer_comment

<form      method      =      'post'
action=\"Correction_Judge.php\">

      <input      type=\"hidden\"
name=\"answer_from_edit_marking\" value=\"$answer\" />

      <input      type=\"hidden\"
name=\"answer_id_from_edit_marking\" value=\"$answer_id\" />

      <input      type=\"hidden\"
name=\"question_from_edit_marking\" value=\"$question\" />

      <input      type=\"hidden\"
name=\"question_id_from_edit_marking\" value=\"$question_id\" />

      <input      type=\"hidden\"
name=\"correction_id_from_edit_marking\" value=\"$correction_id\" />

      ";
      show_correction_add_redo_button($correction_kind_id);
  }
}

function show_correction_spot($correction_spot, $question) {
    if ($correction_spot) {
        $marked_answer = make_markedproblem_from_problem_and_number_list($question,
$correction_spot);

```

```

        print "間違いの位置 : ".$marked_answer;
    } else {
        print "正解です";
    }
}

function show_correction_add_redo_button($correction_kind_id) {
    if ($correction_kind_id == CORRECTANS_ID) {
        print "<input type=\"submit\" name='redo' value=\"この採点をしなおす\">

                                </form><br><br><br>";

    } else {
        print "<input type=\"submit\" name = \"redo\" value=\"この添削をしなおす\">

                                <input type=\"submit\" value=\"この回
答の添削を追加する\"></form><br><br><br>";

    }

}

/*****
 * 同じ解答がすでに表示されたかどうか調べる
 */
function answerIsShown($answer, $question_id, $answer_id) {
    require_once ('../DBAccessors/AnswerData.php');
    $answerids = isInCorrected_getAnswerIds($answer, $question_id);
    isInCorrected_getAnswerIds($answer, $question_id);
    if ($answerids) {
        $i = 0;
        foreach ($answerids as $answerid) {
            $answer_ids[$i] = $answerid;
            $i ++;
        }
    }

    //配列の一番はじめての要素だけ取り除く（1回だけ表示するため）
    $firstanswer_id = array_shift($answer_ids);

```

```

        if ($answer_ids != NULL) { //同じ解答のものが複数あれば
            foreach ($answer_ids as $answerid) {
                if ($answer_id == $answerid) {
                    //すでに表示済みならば true を返す
                    return true;
                } else {
                    //表示されていないならば false を返す
                    return false;
                }
            }
        } else {
            //同じ解答のものがなければ false を返す
            return false;
        }
    }
}
?>

```

```

<form action="TeacherTop.php">
<input type="submit" value="トップページに戻る">
</form><br><br>

```

```

</body>

```

```

</html>=====

```



```

?<html>
<head>
<title>中国語研究室 SystemJUN ver.Shige</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"></head>

<body>
<h1 align="center">添削編集画面</h1><br><br>
添削を編集したい問題を選択してください。<br>
その問題の添削を一覧表示します。<br><br><br>
<table align="center">
<?php
require_once ("../DBAccessors/QuestionData.php");
/*****
*画面遷移後の処理**
*****/
$questions = getCandidateQuestion();

foreach ($questions as $question) {
    $question_text = $question->question;
    $question_id = $question->question_id;
    print "<form method='POST' action=\"EditCorrectionCorrectionList.php\">
            <input type=\"submit\" name = \"question\"
value=\"\$question_text\">
            <input type =\"hidden\" name=\"question_id\"
value=\"\$question_id\"> </form><br><br>";
}
?>

</body>=====

```

?<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>中国語研究室 SystemJUN ver.Shige</title>

</head>

<body>

<h1 align="center">問題作成</h1>

<h4 align="center">問題文を入力してください。
(日本語で。例：こんにちは。) </h4>

<form action="EditQuestion_Create_Confirm.php" method="POST">

<table align="center">

<tr><td>

<input type="text" size="50" name="question">

<h4 align="center">回答例を入力してください。
(中国語で。例：?好。) </h4>

<input type="text" size="50" name="correctAnswer">

</td></tr>

</table>

<table align="center">

<tr><td>

<input type="submit" value="問題作成">

</form>

<form>

<input type="button" onClick="location.href='TeacherTop.php'" value="TOP ページに戻る">

</form>

</td></tr>

</body>

</html>=====

```

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>

<body>
<?php
require_once ('../DBAccessors/QuestionData.php');
if (isset ($_POST["question"]) && $_POST["correctAnswer"]) {
    $question = $_POST["question"];
    $correctAnswer = $_POST["correctAnswer"];

    //HTML タグを無効にする
    $question = htmlspecialchars($question);
    $correctAnswer = htmlspecialchars($correctAnswer);
    //特殊文字列を無効にする
    $question = mysql_escape_string($question);
    $correctAnswer = mysql_escape_string($correctAnswer);

    if ($question != "" && $correctAnswer != "") {
        //問題文をデータベースに追加する
        if (addQuestion($question, $correctAnswer)) {
            echo ("<h2 align='center'>問題<br>「".$question."（回答例：
".$correctAnswer."）」<br>を作成しました。</h2>");
        } else {
            echo ("<h2 align='center'>システムエラーにより、問題を作成できませ
んでした</h2>");
        }
    } else {
        echo ("<h2 align='center'>問題文または回答例が入力されていません</h2>");
    }
}
?>

```

```

<table align="center">
  <tr><td>
    <form>
      <input
        type="button"
        onClick="location.href='EditQuestion_Create.php'" value=" さらに問題を作る ">
    </form>
  </td></tr>
  <tr><td>
    <form>
      <input type="button" onClick="location.href='TeacherTop.php'"
        value=" TOP ページに戻る ">
    </form>
  </td></tr>
</table>
<?

} else {
?>
<center>
<h2>問題作成画面からやり直してください</h2>
<input type="button" onClick="location.href='EditQuestion_Create.php'" value="戻る">
</center>
<?

}
?>

</body>
</html>=====

```

```

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>

<body>
<?php
if (isset ($_POST["question"]) && isset ($_POST["correctAnswer"])) {
    $question = $_POST["question"];
    $correctAnswer = $_POST["correctAnswer"];

    //html を無効にする
    $question = htmlspecialchars($question);
    $correctAnswer = htmlspecialchars($correctAnswer);

    if ($question == "" || $correctAnswer == "") {
        //問題文が入力されていないとき
        if ($question == "") {
            echo ("<h3 align='center'>問題文を入力してください</h3>\n");
        }
        if ($correctAnswer == "") {
            echo ("<h3 align='center'>回答例を入力してください</h3>\n");
        }
        echo ("<table align='center'><tr><td>\n");
        echo ("<form><input type='button'
onClick=\"location.href='EditQuestion_Create.php'\" value=' 問題作成ページに戻る
'></form>\n");
        echo ("</td></tr></table>\n");

    } else {
?>

```


<h3 align="center">問題文：<?php echo(ereg_replace("\'", "'", \$question)); ?></h3>

<h3 align="center">回答例：<?php echo(ereg_replace("\'", "'", \$correctAnswer)); ?></h3>

<h3 align="center">上記の問題を本当に追加しますか？</h3>

<table align="center">

<tr><td>

<form action="EditQuestion_Create_Completed.php" method="post">

<input type="hidden" value="<?php echo(\$question); ?>" name="question">

<input type="hidden" value="<?php echo(\$correctAnswer); ?>" name="correctAnswer">

<input type="submit" value=" 追加する ">

</form>

</td>

<td>

<form>

<input type="button" onClick="location.href='EditQuestion_Create.php'" value=" キャンセル ">

</form>

</td>

</tr>

</table>

<?php

}

} else {

?>

<center>

<h2>問題作成画面からやり直してください。</h2>

<input type="button" onClick="location.href='EditQuestion_Create.php'" value="戻る">

</center>

<?}?>

</body>

</html>

=====


```

?<?php
require_once ('../DBAccessors/QuestionData.php');
//登録されている問題群を取得する
$candidateQuestions = getCandidateQuestion();
//登録されている問題群の数
$questionVolume = count($candidateQuestions);
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>
<body>
<h1 align="center">問題削除画面</h1><br><br>

<table align="center">

<tr><td>

<h4>削除したい問題を選択し、ボタンを選んでください</h4>

<form action="EditQuestion_Delete_Confirm.php" method="POST">
<?php

//削除用レイアウトの作成
if ($questionVolume == 0) {

?>

<h4>問題文が登録されていません</h4>

<?

} else {

```

```

        for ($i = 0; $i < $questionVolume; $i ++) {
?>
        <input type="checkbox" name="delete[]" value="<?=$i?>">
        <?print $i." : ".$candidateQuestions[$i] ->question;?>
        <br>
<?
        }
    }
?>
    <input type="submit" value="削除">
</form>
</td></tr>

</table>

<center>
<form>
<input type="button" onClick="location.href='TeacherTop.php'" value="TOP ページに戻る">
</form>
</center>

</body>

```

```
=====
```

```

<?php

/*
 * Created on 2006/12/11
 *
 * To change the template for this generated file go to
 * Window - Preferences - PHPeclipse - PHP - Code Templates
 */
require_once ('../DBAccessors/QuestionData.php');
//登録されている問題群を取得する
$candidateQuestions = getCandidateQuestion();
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>
<body><center>

<?

//チェックボックスで選択した問題を一間ずつ削除する。
if (isset ($_POST["deleteQuestionVolume"])) {
    //チェックされた問題数
    $deleteQuestionVolume = $_POST["deleteQuestionVolume"];
    //チェックされた問題の番号を取得する
    $checkedQuestions = $_POST["checkedQuestions"];
    //チェックされた問題を一間ずつ削除する
    for ($i = 0; $i < $deleteQuestionVolume; $i ++) {
        $deleteQuestion = $candidateQuestions[$checkedQuestions[$i]];
        $deleteQuestionId = $deleteQuestion->question_id;
        adminInvisible($deleteQuestionId);
    }
}
?>

```

<h1>

<?

//削除された問題を表示させる

for (\$j = 0; \$j < \$deleteQuestionVolume; \$j ++) {

print \$checkedQuestions[\$j]. "番";

if (\$j != \$deleteQuestionVolume -1) {

print "、";

}

}

?>の問題を削除しました。

</h1>

<?}else{?>

<h1>問題を選択してください</h1>

<?}?>

<form>

<input type="button" onClick="location.href='TeacherTop.php'" value="TOP ページに戻る">

</form>

</center>

</body>

</html>

=====

```
<?php
```

```
/*
 * Created on 2006/12/11
 *
 * To change the template for this generated file go to
 * Window - Preferences - PHPeclipse - PHP - Code Templates
 */
require_once ('../DBAccessors/QuestionData.php');
//登録されている問題群を取得する
$candidateQuestions = getCandidateQuestion();
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>
<body><center>
<?
```

```
//チェックボックスで選択した問題を一問ずつ削除候補とする。
```

```
if (isset ($_POST["delete"])) {
    //チェックされた問題数
    $deleteQuestionVolume = count($_POST["delete"]);
    //チェックされた問題の番号を取得する
    $checkedQuestions = $_POST["delete"];
?>
```

```
<h1>
```

```
<?
```

```

//削除候補の問題を表示させる
for ($j = 0; $j < $deleteQuestionVolume; $j ++) {

    print $checkedQuestions[$j]. "番";
    if ($j != $deleteQuestionVolume - 1) {
        print "、 ";
    }
}

?>の問題を削除しますか？
</h1>
<form action="EditQuestion_Delete_Completed.php" method="POST">
<input type="hidden" name="deleteQuestionVolume" value="<?=$deleteQuestionVolume?>">
<?for($i = 0;$i<$deleteQuestionVolume;$i++){?>
<input type="hidden" name="checkedQuestions[]" value="<?=$checkedQuestions[$i]?>">
<?}?>
<input type="submit" value="削除する">
</form>
<?}else{?>
    <h1>問題を選択してください</h1>

    <?}?>

<br>
<br>
<form>
<input type="button" onClick="location.href='EditQuestion_Delete.php'" value="問題削除画面に戻る">

</form>
</center>
</body>
</html>

```

=====

```

?<?php
require_once ('../DBAccessors/QuestionData.php');
//登録されている問題群を取得する
$candidateQuestions = getCandidateQuestion();
//登録されている問題群の数
$questionVolume = count($candidateQuestions);
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>
<body>
<h1 align="center">問題選択画面</h1><br><br>

<table align="center">
<tr><td>

<h4>生徒に解かせたい問題を選び、ボタンを押してください</h4>
<form action="EditQuestion_Select_Completed.php" method="POST">
<?php

if ($questionVolume == 0) {
?>

    <h4>問題文が登録されていません</h4>

    <?

} else {
    for ($i = 0; $i < $questionVolume; $i++) {
?>

```

```

        <input type="checkbox">

        //すでに生徒側から見えている問題はチェックが入る
        if ($candidateQuestions[$i] ->student_visible == 1) {
?>
            checked
            <?}?> name="visibleQuestions[]" value="<?=$i?>">
            <?print $i." : ".$candidateQuestions[$i] ->question;?>
            <br>
        <?

        }

    }
?>
<br>
<input type='submit' value='選択'>
</form>
</td></tr>
</table>

<center>
<form>
<input type="button" onClick="location.href='TeacherTop.php'" value="TOP ページに戻る">
</form>
</center>

</body>

```

```
=====
```



```

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>
<body>
<?php
require_once ('../DBAccessors/QuestionData.php');
$candidateQuestions = getCandidateQuestion();

if (!isset ($_POST["visibleQuestions"])) {
?>
    <center>
    <h3>問題を選択してください。</h3>
    <form>
    <input type="button" onClick="location.href='EditQuestion.php'" value="問題編集画面に
戻る">
    </form>
    </center>
    <?

} else {
    $checkedQuestions = $_POST["visibleQuestions"];
    $checkedVolume = count($checkedQuestions);
    //まずとりあえずすべての問題を生徒から見えなくする
    for ($h = 0; $h < count($candidateQuestions); $h ++) {
        $currentCandidateQuestion = $candidateQuestions[$h];
        $currentCandidateQuestionId = $currentCandidateQuestion->question_id;
        studentInvisible($currentCandidateQuestionId);
    }
?><table align="center">
    <?

```

```

for ($i = 0; $i < $checkedVolume; $i ++ ) {
    //今、まさに見えるようにする問題
    $currentQuestion = $candidateQuestions[$checkedQuestions[$i]];
    $currentQuestionId = $currentQuestion->question_id;
    studentVisible($currentQuestionId);
?>

```

```

<tr>
<td>
<?print $currentQuestion->question;?>
</td>
</tr>
<?
}
?>

```

```

<tr>
<td>
<h3>上記の問題を生徒が解けるように選択しました。</h3>
</td>
</tr>
<tr><td>
    <form>
        <input type="button" onClick="location.href='TeacherTop.php'"
value=" TOP ページに戻る ">
    </form>
</td></tr>
</table>
<?}?>
</body>
</html>=====

```

```

<?php
require_once ('../DBAccessors/QuestionData.php');
//登録されている問題群を取得する
$candidateQuestions = getCandidateQuestion();
//登録されている問題群の数
$questionVolume = count($candidateQuestions);
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>
<body>
<h1 align="center">問題順序入れ替え画面</h1><br><br>

<table align="center">

<tr><td>
<h4>入れ替えたい問題を2つ選択して、ボタンを押してください</h4>
<form action="EditQuestion_Swap_Completed.php" method="POST">
<?php

//入れ替え用レイアウトの作成
if ($questionVolume == 0) {
?>

<h4>問題文が登録されていません</h4>

<?

} else {
    for ($i = 0; $i < $questionVolume; $i++) {
?>

```

```
<input type="checkbox" name="swap[]" value="<?=$i?>">
<?print $i." : ".$candidateQuestions[$i] ->question;?>
<br>

<?

    }

}

?>
<input type="submit" value="入れ替える">
</form>
</td></tr>
</table>

<center>
<form>
<input type="button" onClick="location.href='TeacherTop.php'" value="TOP ページに戻る">
</form>
</center>

</body>
```

=====

```
<?php
```

```
/*
 * Created on 2006/12/11
 *
 * To change the template for this generated file go to
 * Window - Preferences - PHPeclipse - PHP - Code Templates
 */
require_once ('../DBAccessors/QuestionData.php');
//登録されている問題群を取得する
$candidateQuestions = getCandidateQuestion();
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>
<body><center>
<?
```

```
if (isset ($_POST["swap"])) {
    $swapNumbers = $_POST["swap"];
    $swapVolume = count($swapNumbers);
    //選択された問題が二つの場合
    if ($swapVolume == 2) {
        $firstQuestion = $candidateQuestions[$swapNumbers[0]];
        $lastQuestion = $candidateQuestions[$swapNumbers[1]];
        $firstQuestionId = $firstQuestion->question_id;
        $lastQuestionId = $lastQuestion->question_id;
        //echo $firstQuestionId;
        //echo $lastQuestionId;
    }
}
```

```
//問題を入れ替える。
swapOrder($firstQuestionId, $lastQuestionId);

?>

<h1>
<?print $swapNumbers[0]?>番と<?print $swapNumbers[1]?>番の問題を入れ替えました
</h1>
<?}else{?>
<h1>選択する問題は二つです。</h1>

<?

    }
} else {
?>
    <h1>問題を選択してください</h1>

    <?}?>

<form>
<input type="button" onClick="location.href='EditQuestion_Swap.php'" value="さらに問題の順序を
入れ替える">
<br>
<br>
<input type="button" onClick="location.href='TeacherTop.php'" value="TOP ページに戻る">

<form>
</center>
</body>
</html>
```

=====

```

<html>

  <head>

    <title>中国語研究室 SystemJUN （先生用画面） </title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">

  </head>

  <body>
    <h1 align="center">中国語 JUNSsystem </h1>
    <center><br>先生用メニュー<br><BR></center>
    <table align="center">
      <tr><td>
        ----- 問題の編集 -----

      </td></tr>
      <tr><td>
        <form>
          <input type="button"
onClick="location.href='EditQuestion_Create.php'" value=" 問題を作成する ">
        </form>
      </td></tr>
      <tr><td>
        <form>
          <input type="button"
onClick="location.href='EditQuestion_Delete.php'" value=" 問題を削除する ">
        </form>
      </td></tr>
      <tr><td>
        <form>
          <input type="button"
onClick="location.href='EditQuestion_Select.php'" value=" 出題する問題を選択する ">
        </form>
      </td></tr>
      <tr><td>
        <form>

```

```

                                <input                                type="button"
onClick="location.href='EditQuestion_Swap.php'" value=" 問題の順序を入れ替える ">
                                </form>
                                </td></tr>
                                <tr><td>
                                ----- 添削 -----
                                </td></tr>
                                <tr><td>
                                <form method="POST" action="Correction_Judge.php">
                                <input type="submit" name="new_correction" value=" 未添
削問題を添削する ">
                                </form>
                                </td></tr>
                                <tr><td>
                                <form>
                                <input                                type="button"
onClick="location.href='EditCorrectionQuestionList.php'" value=" 既添削を修正する ">
                                </form>
                                </td></tr>
                                <tr><td>
                                ----- 添削データ出力 -----
                                </td></tr>
                                <tr><td>
                                <form>
                                <input                                type="button"
onClick="location.href='DataOutput.php'" value=" エクセル出力 ">
                                </form>
                                </td></tr>
                                </table>
</body>
</html>

```

=====


```

INSERT INTO student(login_name, passwd, name, clazz, register_time)
    VALUES('s00000aa', 'aaa111', 'A1', 'classA', '1000 -10 -10 00:00');
INSERT INTO student(login_name, passwd, name, clazz, register_time)
    VALUES('s11111bb', 'bbb222', 'B2', 'classA', '1111 -11 -11 00:00');
INSERT INTO student(login_name, passwd, name, clazz, register_time)
    VALUES('s22222cc', 'ccc333', 'C3', 'classB', '2222 -12 -12 00:00');
INSERT INTO student(login_name, passwd, name, clazz, register_time)
    VALUES('t00000dd', 'ddd444', 'D4', 'classB', '1212 -12 -12 00:00');
INSERT INTO student(login_name, passwd, name, clazz, register_time)
    VALUES('t11111ee', 'eee555', 'E5', 'classC', '1313 -09 -10 00:00');
INSERT INTO student(login_name, passwd, name, clazz, register_time)
    VALUES('t22222ff', 'fff666', 'F6', 'classC', '2000 -01 -01 00:00');

```

```

INSERT INTO question(question, correct_example, question_order, admin_visible, student_visible)
    VALUES('aaaa', 'I am Japanese.', last_insert_id(), 1, 1);
INSERT INTO question(question, correct_example, question_order, admin_visible, student_visible)
    VALUES('bbbb', 'He is Korean.', last_insert_id(), 1, 0);
INSERT INTO question(question, correct_example, question_order, admin_visible, student_visible)
    VALUES('cccc', 'This is tasty.', last_insert_id(), 1, 1);
INSERT INTO question(question, correct_example, question_order, admin_visible, student_visible)
    VALUES('dddd', 'You are dangerous here.', last_insert_id(), 1, 1);
INSERT INTO question(question, correct_example, question_order, admin_visible, student_visible)
    VALUES('eeee', 'Go ahead.', last_insert_id(), 1, 0);
INSERT INTO question(question, correct_example, question_order, admin_visible, student_visible)
    VALUES('ffff', 'I see.', last_insert_id(), 1, 0);

```

```

INSERT INTO correction_kind(correction_kind, correction_kind_comment)
    VALUES('kind1', 'kind_comment01');
INSERT INTO correction_kind(correction_kind, correction_kind_comment)
    VALUES('kind2', 'kind_comment02');
INSERT INTO correction_kind(correction_kind, correction_kind_comment)
    VALUES('kind3', 'kind_comment03');
INSERT INTO correction_kind(correction_kind, correction_kind_comment)

```

```

VALUES('kind4', 'kind_comment04');
INSERT INTO correction_kind(correction_kind, correction_kind_comment)
VALUES('kind5', 'kind_comment05');
INSERT INTO correction_kind(correction_kind, correction_kind_comment)
VALUES('kind6', 'kind_comment06');

INSERT INTO correction(correction_kind_id, answer_id, answer_comment, correction_spot)
VALUES(1, 1, 'answer_comment01', null);
INSERT INTO correction(correction_kind_id, answer_id, answer_comment, correction_spot)
VALUES(2, 2, 'answer_comment02', null);
INSERT INTO correction(correction_kind_id, answer_id, answer_comment, correction_spot)
VALUES(2, 3, 'answer_comment03', '1,2,3,4,5,6');
INSERT INTO correction(correction_kind_id, answer_id, answer_comment, correction_spot)
VALUES(2, 4, 'answer_comment04', '1,4,5');
INSERT INTO correction(correction_kind_id, answer_id, answer_comment, correction_spot)
VALUES(3, 4, 'answer_comment05', '4,5,6');
INSERT INTO correction(correction_kind_id, answer_id, answer_comment, correction_spot)
VALUES(3, 3, 'answer_comment06', '4,5,6');

INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(1, 's00000aa', 'I am Hoge.', '2000-01-01 00:00', 1);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(1, 's00000aa', 'I like dogs.', '2001-02-01 00:00', 1);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(2, 's00000aa', 'I wish so.', '2002-03-01 00:00', 1);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(2, 's11111bb', 'He goes school.', '2003-04-01 00:00', 1);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(1, 's11111bb', 'She loves me.', '2004-05-01 00:00', 0);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(1, 's11111bb', 'I dislike humans.', '2005-06-01 00:00', 0);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(1, 's00000aa', 'I am Hoge.', '2000-01-01 00:00', 0);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(1, 's00000aa', 'I like dogs.', '2001-02-01 00:00', 0);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)

```

```

VALUES(1, 's00000aa', 'I wish so.', '2002-03-01 00:00', 0);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(1, 's11111bb', 'He goes school.', '2003-04-01 00:00', 0);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(1, 's11111bb', 'She loves me.', '2004-05-01 00:00', 0);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(2, 's00000aa', 'AAAAAAAAAA.', '2006-06-01 00:00', 0);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(2, 's00000aa', 'BBBBBBBBBB.', '2000-01-01 00:00', 0);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(2, 's00000aa', 'CCCCCCCCC.', '2001-02-01 00:00', 0);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(2, 's00000aa', 'DDDDDDDDDD.', '2002-03-01 00:00', 0);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(2, 's11111bb', 'EEEEEEEEEE.', '2003-04-01 00:00', 0);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(2, 's11111bb', 'FFFFFFFFFF.', '2004-05-01 00:00', 0);
INSERT INTO answer(question_id, student_login_name, answer, answer_time, is_corrected)
VALUES(2, 's11111bb', 'GGGGGGGGGG.', '2005-06-01 00:00', 0);

```

=====

```

<?php
/*
 * Created on 2006/11/26
 *
 * To change the template for this generated file go to
 * Window - Preferences - PHPeclipse - PHP - Code Templates
 */

class AnsLog{
    var $problem;
    var $answer;
    var $mistakes;
    var $correctans;
    var $anstime;

    function AnsLog(){
        $this->problem = "aaa ";
        $this->answer = "bbb ";
        $this->mistakes = "124 ";
        $this->correctans="corretans ";
        $this->anstime = "123455 ";
    }

    function set_AnsLogData($problem,$answer,$mistakes,$correctans,$anstime){
        $this->problem = $problem;
        $this->answer = $answer;
        $this->mistakes = $mistakes;
        $this->correctans=$correctans;
        $this->anstime = $anstime;
    }
}
?>
=====

```

```

<?php
require_once ('QuestionData.php');
//生徒が解くことの出来る問題の配列を取得する
$visibleQuestions = getSelectedQuestion();
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>中国語研究室 SystemJUN ver.STUDENT</title>
</head>

<body>
<?php

if ($_POST['questionNumber'] == '') {
    print "<h1>問題を選択してください。</h1>";
    print "<form action='AnswerQuestionTop.php'>
    <input type='submit' value='問題解答画面に戻る'>
    </form>";

} else {
    print "<h1 align='center'>問題</h1><br><br>";
    print "<h3>次の日本語を中国語に訳しなさい。</h3>";
    $questionNumber = $_POST["questionNumber"];
    $question = $visibleQuestions[$questionNumber] ->question;
    print "<h2>「" . $question . "」</h2>";
?>

<font color=red>
<h5>、。などは全角で入力してください</h5>
<h5>かつこは「」を用いてください</h5>
<h5>アルファベットは半角で入力してください</h5>
</font>

```

```
<form method="post" action="Check.php">
<input type="text" name="answer" size="50"><br>
<input type="hidden" name="question" value="<?= $question ?>">
<input type="hidden" name="questionNumber" value="<?= $questionNumber?>">
<input type="submit" value="決定">
</form>
```

```
<form action="AnswerQuestionTop.php">
<input type="submit" value="戻る">
</form>
```

```
<?php
```

```
}
```

```
?>
```

```
</body>
```

```
</html>=====
```

AnswerData.php

?S????\?b?h:3/4

?!u????\?b?h:3/3

?0u????\?b?h:0/1

QuestionData.php

?S????\?b?h:3/6

?!u????\?b?h:0/0

?0u????\?b?h:3/6

CorrectionKindData.php

?S????\?b?h:0/2

?!u????\?b?h:0/0

?0u????\?b?h:0/2

CorrectionData.php

?S????\?b?h:0/3

?!u????\?b?h:0/0

?0u????\?b?h:0/3

StudentData.php

?S????\?b?h:6/6

?!u????\?b?h:6/6

?0u????\?b?h:0/0

DBAccessor.php

?S????\?b?h:6/6

?S?t?@?C??

?S????\?b?h:18/27

?!u????\?b?h:9/9

?0u????\?b?h:3/12

=====

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>中国語研究室 SystemJUN ver.STUDENT</title>
</head>

<body>

<?php
    $answer = $_POST['answer'];
    $question = $_POST['question'];
    $questionNumber = $_POST["questionNumber"];

    print "<h1>".$answer . "</h1><br>";
    print "これでいいですか?";

?>

<form method="post" action="AddNewAnswer.php">
<input type="hidden" name="questionNumber" value="<?= $questionNumber?>">
<input type="hidden" name="answer" value="<?= $answer ?>"><br>
<input type="submit" value="決定">
</form>

<form method="post" action="Answer.php">
<input type="hidden" name="questionNumber" value="<?= $questionNumber ?>">
<input type="submit" value="戻る">
</form>

</body>
</html>=====
```



```

<?php
class CheckedAnswers extends AnsLog{
    var $student;
    var $classname;
    var $mistake;
    function CheckedAnswers() {
        $this->student="default_student";
        $this->problem = "問題のテストデータです";
        $this->answer = "這是樣品。。。";
        $this->mistake="タイプ A の間違いです";
        $this->mistakes = "1,2,3";
        $this->correctans="修理了錯兒：：：";
        $this->anstime = "20061213";
        $this->classname="ClassA";
    }
    function set_student($student) {
        $this->student=$student;
    }
}
?>=====

```

あいうえお,abcde,亜位卵絵尾=====

問題	正解例	誤用例	正解	簡体字	文型	場所（始）	場 所 （ 終 ）
	個別の注釈						
わたしは街を歩いています。	テスト 1		正解です				
よくできました。							
わたしは街を歩いています。	テスト 3		正解です				
よくできました。							
わたしは街を歩いています。	テスト 4		正解です				
よくできました。							
わたしは街を歩いています。	ああ		正解です				
よくできました。							
わたしは街を歩いています。	Object		正解です				
よくできました。							
わたしは街を歩いています。	テスト		正解です				
よくできました。							
わたしは街を歩いています。	テスト\テスト		正解です				
よくできました。							
わたしは街を歩いています。	テスト\テスト		正解です				
よくできました。							
わたしは街を歩いています。	テスト\テスト		正解です				
よくできました。							
わたしは街を歩いています。	テスト\テスト		正解です				
よくできました。							
わたしは街を歩いています。	?不定希望我?着?入恰当的??文想。		正解です				
よくできました。							
わたしは街を歩いています。	ほげほげ 1		正解です				
よくできました。							
わたしは街を歩いています。	ほげほげ 2		正解です				
よくできました。							
わたしは街を歩いています。	ほげほげ 5		正解です				
よくできました。							
わたしは街を歩いています。		テスト 2	簡体字のミスです。	あ			
い	ほげ。						
わたしは街を歩いています。		ほげほげ 3	簡体字のミスです。				

あ い ああ
わたしは街を歩いています。 ほげほげ4
あ い やりなおし。

文型のミスです。

問題	正解例	誤用例	正解=====
----	-----	-----	---------

```

<?php
if(count($_POST) > 0){//$_POST で受け取ったら処理する
    $name = $_POST["name"];
    if(count($_POST["groop1"]) > 0){$groop1 = join("/",$_POST["groop1"]);}//配列を[/]で連結する
    $mes = $_POST["message"];
    $mes = str_replace("\r\n","\n",$mes);
    $mes = str_replace("\r","\n",$mes);
    $mes = str_replace("\n",<br>,$mes);
    $mes = str_replace(' ','',$mes);
    $line = $name." ".$groop1." ".$mes;
}
?>

<html>
<head>
<title>フォームの受け渡し</title>
<meta http-equiv="Content-Type" content="text/html; charset=SHIFT_JIS">
</head>
<body bgcolor="ffffff">
<form method="get" action="form.php">
名前: <input type="text" name="name" ><br>
複数チェックボックス:
<input type="checkbox" name="groop1[]" value="1">
<input type="checkbox" name="groop1[]" value="2">
<input type="checkbox" name="groop1[]" value="3">
<br>
メッセージ: <textarea name="message"></textarea><br>
<input type="submit" name="submit" value="送信">
</form>
<?php print $line; ?>
</body>
</html>
=====

```

```
<?php
/*
 * Created on 2006/11/27
 *
 * To change the template for this generated file go to
 * Window - Preferences - PHPeclipse - PHP - Code Templates
 */
?>
<html lang="ja">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>失敗</title>
</head>
<BODY>
<h1>ログイン名、パスワード共に入力してください。</h1>
</body>
</html>=====
```

★marking.php…添削画面

…画面で入力される情報…

セッションに

*問題

*解答

*問題 ID

を登録している。

■■■■ ♪ NEW ■■■■

…画面で入力される情報…

セッションに

*問題

*解答

*問題 ID

*解答 ID

を登録する。

未添削の解答セットを取得する ()

```
select * from answer,question where answer.question_id=question.question_id and  
answer.is_corrected=0;
```

```
$answer_set->question=question.question;
```

```
$answer_set->answer=answer.answer;
```

```
$answer_set->question_id=question.question_id;
```

```
$answer_set->anserr_id=answer.answer_id;
```

```
return $answer_set;
```

→問題、解答、問題 ID、解答 ID を返す

DB から指定された解答 ID の添削を削除する (解答 ID)//添削のやりなおしのために

*重大なエラーを発見。解答 ID で削除したらダメですよー；

→指定された解答 ID の添削が削除される

■■■■ ♪ NEW ■■■■

DB から指定された解答 ID の添削を削除する (添削 ID、解答 ID)//添削のやりなおしのために

→指定された添削 ID の添削を削除する

→指定された解答 ID に添削があるかどうか調べて、なければ「iscorrected を 0 にする」

★mark_check.php…添削完了画面

…画面で入力される情報…

セッションに

*問題

*解答

*問題 ID

が登録されている。

■■■■ ♪ NEW■■■■

…画面で入力される情報…

セッションに

*問題

*解答

*問題 ID

*解答 ID

を登録する。

添削を追加する（問題 ID、解答、添削の種類、解答に対するコメント、間違いの位置、添削の日にち）

■■■■ ♪ NEW■■■■

添削を追加する（問題 ID、解答、添削の種類 ID、解答に対するコメント、間違いの位置、添削の日にち）

★edit_marking_choose.php…添削編集画面の添削選択画面

…画面で入力される情報…

HIDDEN に

*問題

*解答

*問題 ID

*解答 ID

を登録している。

■■■■ ♪ NEW■■■■

…画面で入力される情報…

HIDDEN に

*問題

*解答

*問題 ID

*解答 ID

*添削 ID

を登録する。

選択された問題から問題 ID を取得する（選択された問題）

→問題 ID を返す

問題 ID から添削された解答と添削をすべて取得する（問題 ID）

→解答

→解答 ID

→添削 ID

→添削の種類

→解答に対するコメント

→間違いの位置

→問題

→問題 ID

■■■■ ♪ NEW ■■■■

問題 ID から添削された解答と添削をすべて取得する（問題 ID）

問題 ID から解答をすべて取得する（問題 ID）

```
select answer.answer_id from answer,question where question_id=$question_id;
```

→解答 ID

```
return answer.answer_id;
```

解答 ID から添削をすべて取得する（解答 ID）

```
select *from ...
```

→解答

→解答 ID

→添削 ID

→添削の種類

→解答に対するコメント

→間違いの位置

→問題

→問題 ID

★edit_marking_choose_question.php…添削編集画面の問題選択画面

削除されていない問題をすべて取得する（）

→問題

=====

```

<?
require_once("Correction_Judge.php");
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>
<body>

<?
/*****
*画面遷移後の処理**
*****/

//日付を取得する
$today = date('Y年m月d日');
//添削と画面の状態を取得する
$correction_view_state=getCorrectionAndViewState();
//添削に必要な情報を取得する
$correction_state=$correction_view_state->correction_state;
$view_state=$correction_view_state->view_state;
$answer_set=getDataForCorrection($correction_state,$view_state);

//添削の種類を追加するボタンが押されていたら
//添削の種類を追加する
if($view_state==FROM_THIS_VIEW){
    if(isset($_POST['correction_kind']) && isset($_POST['correction_kind_comment'])){
        $correction_kind=$_POST['correction_kind'];
        $correction_kind_comment=$_POST['correction_kind_comment'];
        $correction_kind_comment = htmlspecialchars($correction_kind_comment);
    }
}

```

```

        addCorrectionKind($correction_kind, $correction_kind_comment);
    }
}

```

```

$uneditedanswer=$answer_set ->answer;
$uneditedanswerquestion=$answer_set ->question;
$uneditedanswerquestionid=$answer_set ->question_id;

```

```

?>

```

```

<h1 align="center">添削画面</h1><br><br>

```

```

<table align="center">

```

```

<?php

```

```

//添削する問題と回答を保存する

```

```

$_SESSION['PROBLEM'] = $answer_set ->question;

```

```

$_SESSION['PROBLEM_ID'] = $answer_set ->question_id;

```

```

$_SESSION['ANSWER'] = $answer_set ->answer;

```

```

$_SESSION['ANSWER_ID']=$answer_set ->answer_id;

```

```

if($uneditedanswer){

```

```

    print

```

```

    "

```

```

    <h3>質問：$uneditedanswerquestion</h3>

```

```

    <h3>回答：$uneditedanswer</h3>

```

```

    <br>

```

```

    ";

```

```

}else{

```

```

    print"<h3>未処理の回答はありません</h3>

```

```

    <form>

```

```

    <input type="button" onClick="location.href='TeacherTop.php'" value="TOP ページ

```

```

に戻る">

```

```

    </form>

```

```

    ";

```

```

}

```

```

?>

```

```

<h4>添削する</h4>

```

上記の回答が正解の場合、

正解をチェックし、添削完了ボタンを押してください

上記の回答が不正解の場合、

不正解をチェックし、不正解の種類を選択し、

不正解の場所とその回答に対するコメントを入れてください。

そして、添削完了ボタンを押してください。

不正解の種類はこのページの下方で追加することが出来ます。

詳しくは下方の「不正解の種類を追加する」を参照してください。


```
<form method="post" action="Correction_Completed.php" >
```

添削の種類:

```
<select name = "category">
```

```
<?php
```

```
// (不正解) 種類 DB からデータを読み込む
```

```
$correction_kinds=getCorrectionKinds();
```

```
// (不正解) 種類 DB から種類名を取り出し、配列に格納する
```

```
echo $correction_kinds;
```

```
foreach($correction_kinds as $correction_kind){
```

```
    $id=$correction_kind->correction_kind_id;
```

```
    $kind=$correction_kind->correction_kind;
```

```
    $contant=$correction_kind->correction_kind_comment;
```

```
    echo "<option value=\".$id.\">".$kind.":".$contant."</option><br>";
```

```
}
```

```
?>
```

```
</select>
```

```
<br>
```

間違いの位置:

```
<?php make_checkboxed_answer($answer_set->answer);
```

```
?>
```

```
<BR>
```

コメント:<input type="text" name = "comment" size="50">

```
<br><br>
```

```
<input type="submit" value="添削完了">
```

```
</form>
```

[添削編集画面を表示する](EditCorrectionQuestionList.php)

<form method='POST' action="Correction_Judge.php">

<h4>添削の種類を追加する</h4>

添削の種類の名称を決めてください

**
**

例) 単語、文型、簡体字**
**

種類名:

**

**

この添削をしたときに表示するコメントを決めてください

**
**

例) 単語の間違いです。

**
**

コメント:

**

**

</form>

**

**

<form>

</form>

</body>

</html>=====

```
<?
// セッションを開始。
session_start();
// セッションに変数を登録
$_SESSION['PROBLEM'];
$_SESSION['PROBLEM_ID'];
$_SESSION['ANSWER'];
$_SESSION['ANSWER_ID'];
?>

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>中国語研究室 SystemJUN ver.Shige</title>
</head>
<body>

<?php
/**
 * TODO すごくアドホックに DB に接続(なおす必要あり)
 */
//database に接続
global $db;
$db = mysql_connect("localhost","jun","shige");
//database jun に接続
mysql_select_db("jun",$db);
mysql_query("SET NAMES utf8");

$correction_state = 0;//添削の状態
define("NEW_CORRECTION","1");
define("ADD_CORRECTION","2");
define("OVERWRITE_CORRECTION","3");
```



```

$view_state =0;//画面の状態
define("FROM_MENU_VIEW","1");
define("FROM_DONE_VIEW","2");
define("FROM_EDIT_VIEW","3");

/**
 * 添削の状態を取得する
 */
function getCorrectionAndViewState(){
    $view_state = 0;

    if(isset($_POST['same_answer'])){//添削完了画面でフラグを渡された場合
        $correction_state = ADD_CORRECTION;
        $view_state=FROM_DONE_VIEW;
    }else if(isset($_POST['correctanswer_from_edit_marking'])){//添削編集画面で正解フラ
グを渡された場合
        if($_POST['redo']){//上書きの場合
            $view_state=FROM_EDIT_VIEW;
            $correction_state = OVERWRITE_CORRECTION;
        }else{//追加の場合
            $view_state=FROM_EDIT_VIEW;
            $correction_state = ADD_CORRECTION;
        }
    }else if(isset($_POST['incorrectanswer_from_edit_marking'])){//添削編集画面で不正解
フラグを渡された場合
        if($_POST['redo']){//上書きの場合
            $view_state=FROM_EDIT_VIEW;
            $correction_state = OVERWRITE_CORRECTION;
        }else{//追加の場合
            $view_state=FROM_EDIT_VIEW;
            $correction_state = ADD_CORRECTION;
        }
    }else{//まったくフラグがない場合
        $correction_state = NEW_CORRECTION;
    }
}

```

```

        $correction_view_state->correction_state=$correction_state;
        $correction_view_state->view_state=$view_state;

        return $correction_view_state;
    }

    /**
     * 添削に必要な情報を取得する
     */
    function getDataForCorrection($correction_state,$view_state){
        //添削の状態を調べて
        if($correction_state == NEW_CORRECTION){//添削がされていない場合は
            //DB からデータを読み込む
            $answer_set=getAnswerSetFromDB();
        }else if($correction_state == OVERWRITE_CORRECTION){//添削をやり直す場合は
            //画面からデータを読み込む
            $answer_set=getAnswerSetFromView();
            $answer_id=$answer_set->answer_id;
            //DB から添削データを消去する
            mysql_query("DELETE FROM correctanswer WHERE answer_id = $answer_id");
        }else if($correction_state == ADD_CORRECTION){//添削を追加する場合は
            //画面からデータを読み込む
            $answer_set=getAnswerSetFromView();
        }
        return $answer_set;
    }

    /**
     * 添削用データを DB から取得する
     */
    function getAnswerSetFromDB(){
        global $db;
        //問題 DB からデータを読み込む
        $data_problem = mysql_query("SELECT * FROM problem", $db);
        $size_problem = mysql_result(mysql_query("select count(*) from
        problem",$db),0,"count(*)");
    }

```

```

//未分類 DB からデータを読み込む
$data_unclassifiedanswer = mysql_query("SELECT * FROM unclassifiedanswer",$db);
$size_unclassifiedanswer = mysql_result(mysql_query("select count(*) from
unclassifiedanswer",$db),0,"count(*)");
//正解 DB からデータを読み込む
$data_correctanswer = mysql_query("SELECT * FROM correctanswer",$db);
$size_correctanswer = mysql_result(mysql_query("select count(*) from
correctanswer",$db),0,"count(*)");
//不正解 DB からデータを読み込む
$data_incorrectanswer = mysql_query("SELECT * FROM incorrectanswer",$db);
$size_incorrectanswer = mysql_result(mysql_query("select count(*) from
incorrectanswer",$db),0,"count(*)");
//問題 DB から問題要素を取り出し、配列に格納する
for($i = 0; $i < $size_problem; $i++){
    $questions[$i] = mysql_result($data_problem,$i,"question");
}
//未分類 DB から回答要素を取り出し、配列に格納する
for($i = 0; $i < $size_unclassifiedanswer; $i++){
    $unclassifiedanswers[$i] =
mysql_result($data_unclassifiedanswer,$i,"answer");
}
//未分類 DB から問題 ID 要素を取り出し、配列に格納する
for($i = 0; $i < $size_unclassifiedanswer; $i++){
    $unclassifiedanswerquestionid[$i] =
mysql_result($data_unclassifiedanswer,$i,"problem_id");
}
//正解 DB から回答要素を取り出し、配列に格納する
for($i = 0; $i < $size_correctanswer; $i++){
    $correctanswers[$i] = mysql_result($data_correctanswer,$i,"answer");
}
//不正解 DB から回答要素を取り出し、配列に格納する
for($i = 0; $i < $size_incorrectanswer; $i++){
    $incorrectanswers[$i] = mysql_result($data_incorrectanswer,$i,"answer");
}
//未分類 DB にあり、正解 DB、不正解 DB にないものを取り出す。
$is_new_answer = true;

```

```

//正解 DB,不正解 DB にまだ登録されていない場合
if($size_correctanswer == 0 && $size_incorrectanswer == 0){
    $is_new_answer = true;
    $row = 0;
}
//
for($i = 0; $i < $size_unclassifiedanswer; $i++){
    $is_new_answer = true;
    for($j = 0; $j <= $size_correctanswer; $j++){
        for($k = 0; $k <= $size_incorrectanswer; $k++){

            if(($unclassifiedanswers[$i]!=$correctanswers[$j])&&($unclassifiedanswers[$i]!=$incorrectanswers[$k])){

                $row = $i;
            } else {
                //正解、不正解 DB に存在したら、未処理 DB から削除する
                $is_new_answer = false;
                mysql_query("delete from unclassifiedanswer
where answer = '$unclassifiedanswers[$i]'", $db);
            }
        }
    }
    //もし新しい回答だったら
    if($is_new_answer){
        break;
    }
}
//未処理のデータを取得する
if($is_new_answer && $size_unclassifiedanswer!=0){
    $uneditedanswer = $unclassifiedanswers[$row]; //未添削回答
    $uneditedanswerquestionid = $unclassifiedanswerquestionid[$row]; //未添削回答の問題の ID
    $data_uneditedanswerquestion = mysql_query("SELECT * FROM problem where
problem_id = $uneditedanswerquestionid", $db);
    $uneditedanswerquestion = mysql_result($data_uneditedanswerquestion, 0,

```

```
"question");//未添削回答の問題
```

```
    }

    //      $answer_set ->answer=$uneditedanswer;
    //      $answer_set ->question=$uneditedanswerquestion;
    //      $answer_set ->question_id=$uneditedanswerquestionid;
    $answer_set ->answer="中国版こんにちは";
    $answer_set ->question="こんにちは";
    $answer_set ->question_id="2";
    return $answer_set;
}

/**
 * 添削用データを画面から取得する
 */
function getAnswerSetFromView($view_state){
    if($view_state==FROM_EDIT_VIEW){//編集画面から遷移した場合
        $answer_from_edit_marking = $_POST['answer_from_edit_marking'];
        $answer_id_from_edit_marking = $_POST['answer_id_from_edit_marking'];
        $question_from_edit_marking = $_POST['question_from_edit_marking'];
        $question_id_from_edit_marking = $_POST['question_id_from_edit_marking'];

        $answer_set ->question=$question_from_edit_marking;
        $answer_set ->question_id=$question_id_from_edit_marking;

        if($correctanswer_from_edit_marking){//正解の解答の場合
            $answer_set ->answer=$correctanswer_from_edit_marking;
            $answer_set ->answer_id=$correctanswer_id_from_edit_marking;
        }else if($incorrectanswer_from_edit_marking){//不正解の解答の場合
            $answer_set ->answer=$incorrectanswer_from_edit_marking;
            $answer_set ->answer_id=$incorrectanswer_id_from_edit_marking;
        }

        return $answer_set;

    }else if($view_state==FROM_DONE_VIEW){//完了画面から遷移した場合
```

```

        $uneditedanswerquestion = $_SESSION['PROBLEM'];
        $uneditedanswerquestionid = $_SESSION['PROBLEM_ID'];
        $uneditedanswer = $_SESSION['ANSWER'];

        $answer_set >answer=$uneditedanswer;
        $answer_set >question=$uneditedanswerquestion;
        $answer_set >question_id=$uneditedanswerquestionid;
        return $answer_set;
    }
}

```

```

/*****
*画面遷移後の処理**
*****/

```

```

//日付を取得する

```

```

$today = date('Y年m月d日');

```

```

//database に接続

```

```

$db = mysql_connect("localhost","jun","shige");

```

```

//database jun に接続

```

```

mysql_select_db("jun",$db);

```

```

mysql_query("SET NAMES utf8");

```

```

//添削と画面の状態を取得する

```

```

$correction_view_state=getCorrectionAndViewState();

```

```

//添削に必要な情報を取得する

```

```

$correction_state=$correction_view_state >correction_state;

```

```

$view_state $correction_view_state >view_state;

```

```

$answer_set=getDataForCorrection($correction_state,$view_state);

```

```

$uneditedanswer=$answer_set >answer;

```

```

$uneditedanswerquestion=$answer_set >question;

```

```

$uneditedanswerquestionid=$answer_set >question_id;

```

```

/**セッションから必要なデータをとっているらしい*/
////メニュー画面以外から遷移してきた場合
////添削確認画面で、同じ問題を添削するを選んだ場合
//$_same_answer = $_POST['same_answer'];
////添削編集画面から遷移してきた場合
//$_correctanswer_from_edit_marking = $_POST['correctanswer_from_edit_marking'];
//$_correctanswer_id_from_edit_marking = $_POST['correctanswer_id_from_edit_marking'];
//$_incorrectanswer_from_edit_marking = $_POST['incorrectanswer_from_edit_marking'];
//$_incorrectanswer_id_from_edit_marking = $_POST['incorrectanswer_id_from_edit_marking'];
//$_question_from_edit_marking = $_POST['question_from_edit_marking'];
//$_question_id_from_edit_marking = $_POST['question_id_from_edit_marking'];
////添削のしなおしかしなおしてないか
//$_is_redo = $_POST['redo'];
//
//
////同じ問題を解くと添削確認画面で選んだ場合
//if($_same_answer){
//    $_uneditedanswerquestion = $_SESSION['PROBLEM'];
//    $_uneditedanswerquestionid = $_SESSION['PROBLEM_ID'];
//    $_uneditedanswer = $_SESSION['ANSWER'];
//}else
////問題編集からきた場合
//if($_correctanswer_from_edit_marking){
//    $_uneditedanswerquestion = $_question_from_edit_marking;
//    $_uneditedanswerquestionid = $_question_id_from_edit_marking;
//    $_uneditedanswer = $_correctanswer_from_edit_marking;
//    if($_is_redo){
//        mysql_query("DELETE FROM correctanswer WHERE answer_id =
$_correctanswer_id_from_edit_marking");
//    }
//}else if($_incorrectanswer_from_edit_marking){
//    $_uneditedanswerquestion = $_question_from_edit_marking;
//    $_uneditedanswerquestionid = $_question_id_from_edit_marking;
//    $_uneditedanswer = $_incorrectanswer_from_edit_marking;
//    if($_is_redo){

```

```

//          mysql_query("DELETE FROM incorrectanswer WHERE answer_id =
$incorrectanswer_id_from_edit_marking");
//      }
//}
//else
////メニュー画面から来た場合
//{
//
//    //問題 DB からデータを読み込む
//    $data_problem = mysql_query("SELECT * FROM problem", $db);
//    $size_problem = mysql_result(mysql_query("select count(*) from
problem",$db),0,"count(*)");
//    //未分類 DB からデータを読み込む
//    $data_unclassifiedanswer = mysql_query("SELECT * FROM unclassifiedanswer",$db);
//    $size_unclassifiedanswer = mysql_result(mysql_query("select count(*) from
unclassifiedanswer",$db),0,"count(*)");
//    //正解 DB からデータを読み込む
//    $data_correctanswer = mysql_query("SELECT * FROM correctanswer",$db);
//    $size_correctanswer = mysql_result(mysql_query("select count(*) from
correctanswer",$db),0,"count(*)");
//    //不正解 DB からデータを読み込む
//    $data_incorrectanswer = mysql_query("SELECT * FROM incorrectanswer",$db);
//    $size_incorrectanswer = mysql_result(mysql_query("select count(*) from
incorrectanswer",$db),0,"count(*)");
//    //問題 DB から問題要素を取り出し、配列に格納する
//    for($i = 0; $i < $size_problem; $i++){
//        $questions[$i] = mysql_result($data_problem,$i,"question");
//    }
//    //未分類 DB から回答要素を取り出し、配列に格納する
//    for($i = 0; $i < $size_unclassifiedanswer; $i++){
//        $unclassifiedanswers[$i] =
mysql_result($data_unclassifiedanswer,$i,"answer");
//    }
//    //未分類 DB から問題 ID 要素を取り出し、配列に格納する
//    for($i = 0; $i < $size_unclassifiedanswer; $i++){
//        $unclassifiedanswerquestionid[$i] =

```



```

mysql_result($data_unclassifiedanswer,$i,"problem_id");
//      }
//      //正解 DB から回答要素を取り出し、配列に格納する
//      for($i = 0; $i < $size_correctanswer; $i++){
//          $correctanswers[$i] = mysql_result($data_correctanswer,$i,"answer");
//      }
//      //不正解 DB から回答要素を取り出し、配列に格納する
//      for($i = 0; $i < $size_incorrectanswer; $i++){
//          $incorrectanswers[$i] = mysql_result($data_incorrectanswer,$i,"answer");
//      }
//      //未分類 DB にあり、正解 DB、不正解 DB にないものを取り出す。
//      $is_new_answer = true;
//      //正解 DB,不正解 DB にまだ登録されていない場合
//      if($size_correctanswer == 0 && $size_incorrectanswer == 0){
//          $is_new_answer = true;
//          $row = 0;
//      }
//      //
//      for($i = 0; $i < $size_unclassifiedanswer; $i++){
//          $is_new_answer = true;
//          for($j = 0; $j <= $size_correctanswer; $j++){
//              for($k = 0; $k <= $size_incorrectanswer; $k++){
//
//                  if(($unclassifiedanswers[$i]!=$correctanswers[$j])&&($unclassifiedanswers[$i]!=$inco
rrectanswers[$k])){
//
//                      $row = $i;
//                      } else {
//                          //正解、不正解 DB に存在したら、未処理 DB から削
除する
//
//                          $is_new_answer = false;
//                          mysql_query("delete from unclassifiedanswer
where answer = '$unclassifiedanswers[$i]'", $db);
//
//                      }
//                  }
//              }
//          }
//      }
//      //もし新しい回答だったら

```

```

//          if($is_new_answer){
//          break;
//          }
//      }
//      //未処理のデータを取得する
//      if($is_new_answer && $size_unclassifiedanswer!=0){
//          $uneditedanswer = $unclassifiedanswers[$row]; //未添削回答
//          $uneditedanswerquestionid = $unclassifiedanswerquestionid[$row]; //未添削回答の問題の ID
//          $data_uneditedanswerquestion = mysql_query("SELECT * FROM problem where problem_id = $uneditedanswerquestionid",$db);
//          $uneditedanswerquestion = mysql_result($data_uneditedanswerquestion, 0, "question");//未添削回答の問題
//      }
//}
////不正解種類 DB に追加するボタンから遷移した場合
////不正解種類 DB に追加の処理

//
//include ("junregulator.php");
//*****/
////print $category_category."<br>";
//$category_category=normalize($category_category);
////print $category_category."<br>";
//
////print $category_context."<br>";
//$category_context=normalize($category_context);
////print $category_context."<br>";
//*****/
////$category_category = $_POST['category'];
//$category_context = $_POST['context'];
//
////HTML タグを無効にする
//$category_context = htmlspecialchars($category_context);
//
//if($category_category && $category_context){

```

```
//      // (不正解) 種類 DB からデータを読み込む
//      $size_category      =      mysql_result(mysql_query("SELECT      count(*)      FROM
category",$db),0,"count(*)");
//
//      mysql_query("INSERT      INTO      category      (category,content,date)
VALUES('$category_category','$category_context','$today')");
//}
```

?>

<h1 align="center">添削画面</h1>

<table align="center">

<?php

// 添削する問題と回答を保存する

\$_SESSION['PROBLEM'] = \$uneditedanswerquestion;

\$_SESSION['PROBLEM_ID'] = \$uneditedanswerquestionid;

\$_SESSION['ANSWER'] = \$uneditedanswer;

if(\$uneditedanswer){

print

"

<h3>質問: \$uneditedanswerquestion</h3>

<h3>回答: \$uneditedanswer</h3>

";

}else{

print"<h3>未処理の回答はありません</h3>

<form>

<input type="button" onClick="location.href='TeacherTop.php'" value="TOP ページ

に戻る">

</form>

";

}

?>

<h4>添削する</h4>

上記の回答が正解の場合、

正解をチェックし、添削完了ボタンを押してください

上記の回答が不正解の場合、

不正解をチェックし、不正解の種類を選択し、

不正解の場所とその回答に対するコメントを入れてください。

そして、添削完了ボタンを押してください。

不正解の種類はこのページの下方で追加することが出来ます。

詳しくは下方の「不正解の種類を追加する」を参照してください。


```
<form method="POST" action="Correction_Completed.php" >
```

```
<input type="radio" checked name="radios" value = "correct"> 正 解    <input type="radio"
name="radios" value = "incorrect">不正解<br><br>
```

不正解の種類:

```
<select name = "category">
```

```
<?php
```

```
// (不正解) 種類 DB からデータを読み込む
```

```
$data_category = mysql_query("SELECT * FROM category", $db);
```

```
$size_category = mysql_result(mysql_query("SELECT count(*) FROM category",$db),0,"count(*)");
```

```
// (不正解) 種類 DB から種類名を取り出し、配列に格納する
```

```
for($i = 0; $i < $size_category; $i++){
```

```
    $categorys[$i] = mysql_result($data_category,$i,"category");
```

```
    $category_id[$i] = mysql_result($data_category,$i,"category_id");
```

```
    $category_content[$i] = mysql_result($data_category,$i,"content");
```

```
    $count = $category_id[$i] + 1;
```

```
    print
```

```
    "
```

```
    <option value = $category_id[$i]>$categorys[$i] : $category_content[$i]</option><br>
```

```
    ";
```

```
    }
```

```
mysql_close($db);
```

```
?>
```

```
</select>
```

```
<br>
```

不正解の場所:

```
<input type="text" name = "range_start" size="4">
```

と

<input type="text" name = "range_end" size="4">

の間

コメント : <input type="text" name = "comment" size="50">

<input type="submit" value="添削完了">

</form>

添削編集画面を表示する

<form method='POST' action="Correction_Judge.php">

<h4>不正解の種類を追加する</h4>

不正解の種類の名称を決めてください

例) 単語、文型、簡体字

種類名 : <input type="text" name ="category" size="10">

その不正解をしたときに表示するコメントを決めてください

例) 単語の間違いです。

コメント : <input type="text" name ="context" size="30">

<input type="submit" value="追加する">

</form>

<form>

<input type="button" onClick="location.href='TeacherTop.php'" value="TOP ページに戻る">

</form>

</body>

</html>=====

```
<?
```

```
// セッションを開始。
```

```
session_start();
```

```
// セッションに変数を登録
```

```
$_SESSION['PROBLEM'];
```

```
$_SESSION['PROBLEM_ID'];
```

```
$_SESSION['ANSWER'];
```

```
?>
```

```
<?php
```

```
require_once("../DBAccessors/MarkingData.php");
```

```
require_once("../Common/HTMLNormalizer.php");
```

```
require_once("../DBAccessors/CorrectionKindData.php");
```

```
$correction_state = 0;//添削の状態
```

```
define("NEW_CORRECTION","1");
```

```
define("ADD_CORRECTION","2");
```

```
define("OVERWRITE_CORRECTION","3");
```

```
$view_state =0;//画面の状態
```

```
define("FROM_MENU_VIEW","1");
```

```
define("FROM_DONE_VIEW","2");
```

```
define("FROM_EDIT_VIEW","3");
```

```
define("FROM_THIS_VIEW","4");
```

```
/**
```

```
 * 添削の状態を取得する
```

```
*/
```

```
function getCorrectionAndViewState(){
```

```
    $view_state = FROM_MENU_VIEW;
```

```
    if(isset($_POST['same_answer'])){//添削完了画面でフラグを渡された場合
```

```

        $correction_state = ADD_CORRECTION;
        $view_state=FROM_DONE_VIEW;
    }else if(isset($_POST['answer_from_edit_marking'])){//添削編集画面でフラグを渡された
場合
        if(isset($_POST['redo'])){//上書きの場合
            $view_state=FROM_EDIT_VIEW;
            $correction_state = OVERWRITE_CORRECTION;
        }else{//追加の場合
            $view_state=FROM_EDIT_VIEW;
            $correction_state = ADD_CORRECTION;
        }
    }else if (isset($_POST['add_category'])){
        $view_state=FROM_THIS_VIEW;
        $correction_state = NEW_CORRECTION;
    }else{//まったくフラグがない場合
        $correction_state = NEW_CORRECTION;
    }

    $correction_view_state->correction_state=$correction_state;
    $correction_view_state->view_state=$view_state;

    return $correction_view_state;
}

$correction_view_state=getCorrectionAndViewState();
$view_state=$correction_view_state->view_state;
$correction_state=$correction_view_state->correction_state;
echo $view_state;
echo $correction_state;

/**
 * 添削に必要な情報を取得する
 */
function getDataForCorrection($correction_state,$view_state){
    //添削の状態を調べて
    if($correction_state == NEW_CORRECTION){

```

```

        //添削がされていない場合は
            //DB からデータを読み込む
            $answer_set=getAnswerSetFromDB();
        }else if($correction_state == OVERWRITE_CORRECTION){
        //添削をやり直す場合は
            //画面からデータを読み込む
            $answer_set=getAnswerSetFromView($view_state);
            $answer_id=$answer_set->answer_id;
            $correction_id=$answer_set->correction_id;
            //DB から添削データを消去する
            deleteCorrection($correction_id,$answer_id);
        }else if($correction_state == ADD_CORRECTION){
        //添削を追加する場合は
            //画面からデータを読み込む
            $answer_set=getAnswerSetFromView($view_state);
        }

        return $answer_set;
    }

    /**
     * 添削用データを DB から取得する
     */
    function getAnswerSetFromDB(){
        $answer_set=getUnCorrectedAnswerSet();
        return $answer_set;
    }

    /**
     * 添削用データを画面から取得する
     */
    echo $view_state;
    echo $view_state;
    function getAnswerSetFromView($view_state){
        if($view_state==FROM_EDIT_VIEW){//編集画面から遷移した場合
            $answer_from_edit_marking = $_POST['answer_from_edit_marking'];

```



```

        $answer_id_from_edit_marking = $_POST['answer_id_from_edit_marking'];
        $question_from_edit_marking = $_POST['question_from_edit_marking'];
        $question_id_from_edit_marking = $_POST['question_id_from_edit_marking'];
        $correction_id_from_edit_marking =
$_POST['correction_id_from_edit_marking'];

        $answer_set >question=$question_from_edit_marking;
        $answer_set >question_id=$question_id_from_edit_marking;
        $answer_set >answer=$answer_from_edit_marking;
        $answer_set >answer_id=$answer_id_from_edit_marking;
        $answer_set >correction_id=$correction_id_from_edit_marking;

    }else if($view_state==FROM_DONE_VIEW){//完了画面から遷移した場合
        $answer_set >answer=$_SESSION['ANSWER'];
        $answer_set >answer_id=$_SESSION['ANSWER_ID'];
        $answer_set >question=$_SESSION['PROBLEM'];
        $answer_set >question_id=$_SESSION['PROBLEM_ID'];
    }
    return $answer_set;
}

/*****
* 不正解の位置を入力するチェックボックスを表示する
*/
function make_checkboxed_answer($answer){
    $size_of_answer=mb_strlen($answer);//問題文の文字数
    $checkbox_added_answer="";
    $letter_pointer=0;
    while($letter_pointer<$size_of_answer){
        $letter = mb_substr($answer,$letter_pointer, 1);
        $letter=$letter."<input          type=\"checkbox\"          name=\"letters[]\"
value=\"\"$letter_pointer\">\n";
        $checkbox_added_answer=$checkbox_added_answer.$letter;
        $letter_pointer++;
    }
    print $checkbox_added_answer;

```

}

=====

```
<?php
```

```
class MistakeLog {
```

```
    var $location;
```

```
    var $type;
```

```
    var $correction;
```

```
    function MistakeLog() {
```

```
        $this->location = "1";
```

```
        $this->type = "タイプ1の間違いです";
```

```
        $this->correction = "訂正例です";
```

```
    }
```

```
    function set_MistakeLogData($location,$type,$correction){
```

```
        $this->location=$location;
```

```
        $this->type=$type;
```

```
        $this->correction=$correction;
```

```
    }
```

```
}
```

```
?>=====
```

=====

```

<?
/* セッションを開始。セッション処理は HTTP ヘッダの送信前に行わないと
 * いけません。 */
session_start();
/* セッションに変数を登録 */
session_register("student_id");
?>

<? header("Content -type: text/html;charset=EUC_JP"); ?>
<html><head><title>ログイン</title></head>
<body bgcolor="white">
<?
if(isset($HTTP_POST_VARS["student_id"])){
    /* パスワードのチェックはしていません( _;) */
    $student_id = trim($HTTP_POST_VARS["student_id"]);
    $password   = trim($HTTP_POST_VARS["password"]);

    if(empty($student_id) || empty($password)){
?>
        <h1>おおっと</h1>
        入力が不完全です。受講生 ID とパスワードを入力して下さい。

<?
    }
    else{
?>
        <h1>ログインしました。 </h1>
        ようこそ、<?= $student_id ?>さん。 <br>
        <a href="session -main.php">メインページへ</a><br>
        <a href="session -login.php">戻る</a>

<?
    }
}
else{
?>

```

<h1>ログインフォーム</h1>

受講生 ID とパスワードを入力して下さい。

<form action="session_login.php" method="post">

受講生 ID :

<input type="text" name="student_id">

パスワード :

<input type="password" name="password">

<input type="submit" value="ログイン">

<input type="reset" value="クリア">

</form>

<?

}

?>

</body></html>

=====

```

<?php
/*
 * Created on 2006/12/07
 *
 * To change the template for this generated file go to
 * Window - Preferences - PHPeclipse - PHP - Code Templates
 */
/*****
 * 添削結果データオブジェクトのリストを作成する
 */
require_once("AnsLog.class.php");//添削結果データのクラスを読み込む
require_once("MistakeLog.class.php");//間違い結果データのクラスを読み込む
require_once("CheckedAnswers.class.php");//全添削結果データのクラスを読み込む

function make_anslog_list(){
    // -注釈 -
    //php ではグローバル変数は関数内で宣言しなければならないため、ここで宣言
    //http://php.s3.to/man/language/variables.scope.html
    global $anslog_list;
    global $miss_list;

    $anslog_list[0]=new AnsLog();
    $miss_list[0]=new MistakeLog();
    $miss_list[0] ->set_MistakeLogData("2,3,4","1 -1 漢字の間違いです","是好的天氣。");
    $miss_list[1]=new MistakeLog();
    $miss_list[1] ->set_MistakeLogData("1","1 -2 動詞の間違いです","是好★的天氣。");
    $miss_list[2]=new MistakeLog();
    $miss_list[2] ->set_MistakeLogData("5","1 -3 ! を忘れています","是好的天氣!");
    $anslog_list[0] ->set_AnsLogData("1 いい天氣ですね。","是好的天氣。",$miss_list,"是好的天氣。","200611102345");
    $miss_list=NULL;

    $anslog_list[1]=new AnsLog();

```

```
$anslog_list[1] >set_AnsLogData("2 おはようございます","早上好","", "早上好", "200611132147");
```

```
$anslog_list[2]=new AnsLog();  
$anslog_list[2] >set_AnsLogData("3 あなたの家はどこですか?","?的家在?裡?","", "?的家在?裡?", "200611172045");
```

```
$anslog_list[3]=new AnsLog();  
$anslog_list[3] >set_AnsLogData("4 私は昨日、図書館に行っていました。","我昨天，去著圖書館。","", "我昨天，去著圖書館。", "200611202356");
```

```
$anslog_list[4]=new AnsLog();  
$anslog_list[4] >set_AnsLogData("5 私は日本人です。","我是日本人。","", "我是日本人。", "200611252345");
```

```
$anslog_list[5]=new AnsLog();  
$anslog_list[5] >set_AnsLogData("6 あなたはアメリカに行ったことがありますか?","?去過美國?","", "?去過美國?","", "200611212345");
```

```
$anslog_list[6]=new AnsLog();  
$miss_list[0]=new MistakeLog();  
$miss_list[0] >set_MistakeLogData("3,4,5,6","7 -1 漢字の間違いです","我喜歡★巧★克力。");
```

```
$miss_list[1]=new MistakeLog();  
$miss_list[1] >set_MistakeLogData("4","7 -2 動詞の間違いです","我喜歡★巧克力!");  
$anslog_list[6] >set_AnsLogData("7 私はチョコレートが好きです。","我喜歡巧克力。", $miss_list, "我喜歡巧克力。", "200611212350");  
$miss_list=NULL;
```

```
$anslog_list[7]=new AnsLog();  
$miss_list[0]=new MistakeLog();  
$miss_list[0] >set_MistakeLogData("3,4,5,6","8 -1 漢字の間違いです","我喜歡★巧★克力。");
```

```
$anslog_list[7] >set_AnsLogData("8 あなたは学校が好きですか?","?喜歡學校??", $miss_list, "?喜歡學校??", "200611242333");  
$miss_list=NULL;
```



```

        $anslog_list[8]=new AnsLog();
        $anslog_list[8] >set_AnsLogData("9 私の兄は大学生です。","我的哥哥是大学生。","", "我的哥哥是大学生。","200611252345");
    }

```

```

make_anslog_list();

```

```

/*****

```

```

 * 全添削結果データを作成する

```

```

 */

```

```

function make_checkedanswers_list(){
    global $checkedanswers_list;
    $checkedanswers_list[0]=new CheckedAnswers();
    $checkedanswers_list[0] >set_student("s03037ma");
    $checkedanswers_list[1]=new CheckedAnswers();
    $checkedanswers_list[1] >set_student("s03037ma");
    $checkedanswers_list[2]=new CheckedAnswers();
    $checkedanswers_list[2] >set_student("s03037ma");
    $checkedanswers_list[3]=new CheckedAnswers();
    $checkedanswers_list[3] >set_student("s03037ma");
    $checkedanswers_list[4]=new CheckedAnswers();
    $checkedanswers_list[5]=new CheckedAnswers();
    $checkedanswers_list[6]=new CheckedAnswers();
    $checkedanswers_list[7]=new CheckedAnswers();
}

```

```

make_checkedanswers_list();

```

```

?>

```

```

=====

```

