

Лабораторная работа №3

Дисциплина: «Технологии организации сетевых приложений»

Тема: «Обработка HTML-запросов и генерация страниц»

Выполнил	ст. Рясков А.С.
Группа	ПрИн-6
Проверил	к.т.н., доц. Сычёв О.А.
Дата	25.12.13

Задача

Общее задание

Привести систему оценивания модуля Moodle Poasassignment к формату Advanced Grading Method. А также привести метод оценивания на основе взвешенной суммы, применённый в Poasassignment к виду AGM.

Задание на лабораторную работу

1) Создать зарисовки экранных форм будущего модуля. Реализовать пользовательский интерфейс модуля в Moodle.

Выполнение

Define weighted total

Name*	<input type="text" value="WeightedTotal 1"/>
Description	<input type="text" value="WeightedTotal 1 Description"/>

Criterion name	<input type="text" value="WT-Crit-1"/>
Criterion description	<input type="text" value="WT-Crit-1-Desc"/>
Criterion weight	<input type="text" value="0.4"/>
Criterion source	<input type="text" value="manually"/> ▼
<input type="checkbox"/> Delete criterion	

Add criterion

Save

Save as draft

Cancel

Зарисовка пользовательского интерфейса будущего модуля.
Добавление определения (definition) метода оценивания.

Name* Weighted Total 1

Description Weighted Total 1 Description

Moodle auto-format ▾

Criterion name ? WT-Crit-1

Criterion description ? WT-Crit-1 Description

Criterion weight ? 0.5

Criterion source ? manually ▾

Delete criterion ☐

Criterion name ? WT-Crit-2

Criterion description ? WT-Crit-2 Description

Criterion weight ? 0.5

Criterion source ? manually ▾

Delete criterion ☐

Add 1 field(s) to form

Save Save as draft Cancel

There are required fields in this form marked *.

Реализованный пользовательский интерфейс модуля.
Добавление определения (definition) метода оценивания.

Исходный код формы

```
public function definition() {
    global $DB;
    $form = $this->_form;

    $form->addElement('hidden', 'areaid');
    $form->setType('areaid', PARAM_INT);

    $form->addElement('hidden', 'returnurl');

    $form->addElement('text', 'name', get_string('name',
'gradingform_weightedtotal'), array('size'=>52));
    $form->addRule('name', get_string('required'), 'required');
```

```

$form->setType('name', PARAM_TEXT);

$options =
gradingform_weightedtotal_controller::description_form_field_options($this->_c
ustomdata['context']);
    $form->addElement('editor', 'description_editor',
get_string('description', 'gradingform_weightedtotal'), null, $options);
    $form->setType('description_editor', PARAM_RAW);

    $repeatarray = array();
    $repeatarray[] = $form->createElement('header', 'criterionheader');
    $repeatarray[] = $form->createElement('text', 'name',
get_string('criterionname', 'gradingform_weightedtotal'), array('size'=>45));
    $repeatarray[] = $form->createElement('textarea', 'description',
get_string('criteriondescription', 'gradingform_weightedtotal'));
    $repeatarray[] = $form->createElement('text', 'weight',
get_string('criterionweight', 'gradingform_weightedtotal'));
    $sources[0] = 'manually';

    $usedgraders = $DB->get_records('gradingform_wt_crits');
    foreach($usedgraders as $usedgraderrecord) {
        $grader = $DB->get_record('gradingform_wt_graders', array('id' =>
$usedgraderrecord->graderid));
        $gradername = $grader->name;
        require_once($grader->path);
        $sources[$usedgraderrecord->graderid] = $gradername::name();

        $form->addElement('hidden', 'grader' . (count($sources) - 1),
$usedgraderrecord->graderid);
        $form->setType('grader' . (count($sources) - 1), PARAM_INT);
    }
    $repeatarray[] = $form->createElement('select', 'source',
get_string('criterionsource', 'gradingform_weightedtotal'), $sources);
    $repeatarray[] = $form->createElement('checkbox', 'delete',
get_string('deletecriterion', 'gradingform_weightedtotal'));
    $repeatarray[] = $form->createElement('hidden', 'criterionid', -1);

    $repeatno = 1;

    $repeateloptions = array();

    $repeateloptions['name']['helpbutton'] = array('criterionname',
'gradingform_weightedtotal');
    $repeateloptions['description']['helpbutton'] =
array('criteriondescription', 'gradingform_weightedtotal');
    $repeateloptions['weight']['helpbutton'] = array('criterionweight',
'gradingform_weightedtotal');
    $repeateloptions['source']['helpbutton'] = array('criterionsource',

```

```

'gradingform_weightedtotal');

    $repeateoptions['delete']['default'] = 0;
    $repeateoptions['delete']['disabledif'] = array('criterionid', 'eq',
-1);

    $form->setType('criterionid', PARAM_INT);

    $this->repeat_elements($repeataarray, $repeatno, $repeateoptions,
'option_repeats', 'option_add_fields', 1);

    //$mform->addElement('hidden', 'id', $instance['id']);
    //$mform->setType('id', PARAM_INT);

    $form->addElement('hidden', 'page', 'criteriaions');
    $form->setType('page', PARAM_TEXT);

    $buttonarray = array();
    $buttonarray[] = &$form->createElement('submit', 'saveweightedttotal',
get_string('saveweightedttotal', 'gradingform_weightedttotal'));
    if ($this->_customdata['allowdraft']) {
        $buttonarray[] = &$form->createElement('submit',
'saveweightedttotaldraft', get_string('saveweightedttotaldraft',
'gradingform_weightedttotal'));
    }
    $editbutton = &$form->createElement('submit', 'editweightedttotal', '
');
    $editbutton->freeze();
    $buttonarray[] = &$editbutton;
    $buttonarray[] = &$form->createElement('cancel');
    $form->addGroup($buttonarray, 'buttonar', '', array(' '), false);
    $form->closeHeaderBefore('buttonar');
}

```