**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"**

Лабораторная работа № 4

Тема: «Создание PHP-скрипта»

Проверил: Выполнил:

асс. каф. ПИ ст. гр. ПИ-18б

Артёменко О.Г. Моргунов А.Г.

\_\_\_\_.\_\_\_\_.2021г. \_\_\_\_.\_\_\_\_.2021г.

асс. каф. ПИ

Филипишин Д.А.

\_\_\_\_.\_\_\_\_.2021г.

Донецк – 2021

**Задание к лабораторной работе:**

Во всех вариантах заданий необходимо разработать PHP-скрипт, реализующий некоторый тест и счетчик выполнения теста.

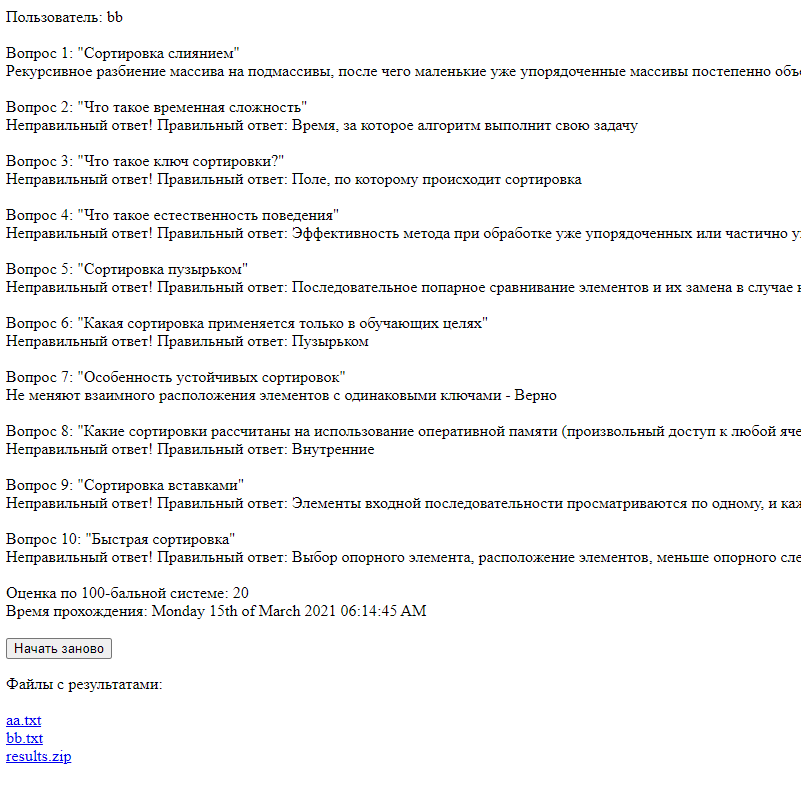
Тест должен содержать не менее десяти вопросов с не менее чем тремя вариантами ответа на каждый вопрос. На некоторые вопросы может предлагаться несколько правильных вариантов ответов. Чередование вопросов должно быть в случайном порядке. Вопросы должны быть разделены на три и более тем.

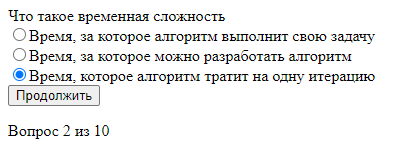
Результаты теста должны отображаться в браузере и сохраняться   
в данных **текущей сессии браузера** пока пользователь полностью не пройдет тест. В случае прерывания выполнения теста накопленные данные не хранить. По мере завершения теста, сохранить итог в файл (файл упаковать   
в zip архив) доступный по ссылке на странице с результатами теста (при скачивании распаковать и отдать пользователю только txt файл). Кроме результатов на странице и в файле должны быть указаны дата и время прохождения теста.

11. Проверка знаний по методам сортировки таблиц.

**Результат выполнения программы**







**Листинг програмы**

<?php

session\_start();

//session\_destroy();

if(isset($\_GET['file'])) {

download($\_GET['file']);

return;

}

#системные переменные

$count = 10;

$questions = array(

array('question'=>'Что такое ключ сортировки?', 'request'=>'Поле, по которому происходит сортировка/Число, которое запускает сортировку/Пароль, без которого сортировка не запустится', 'answer'=>'Поле, по которому происходит сортировка'),

array('question'=>'Особенность устойчивых сортировок', 'request'=>'Всегда делают одинаковое количество шагов/Не меняют взаимного расположения элементов с одинаковыми ключами/Хорошо работают на частично отсортированных данных', 'answer'=>'Не меняют взаимного расположения элементов с одинаковыми ключами'),

array('question'=>'Что такое естественность поведения', 'request'=>'Эффективность метода при обработке уже упорядоченных или частично упорядоченных данных/Непредсказуемый порядок расположения элементов с одинаковыми ключами/Использование ИИ для улучшения алгоритма', 'answer'=>'Эффективность метода при обработке уже упорядоченных или частично упорядоченных данных'),

array('question'=>'Сортировка пузырьком', 'request'=>'Последовательное попарное сравнивание элементов и их замена в случае неправильного порядка/Элементы входной последовательности просматриваются по одному, и каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов/Рекурсивное разбиение массива на подмассивы, после чего маленькие уже упорядоченные массивы постепенно объединяются с проверкой порядка элементов/Выбор опорного элемента, расположение элементов, меньше опорного слева, а больше него справа. Рекурсивное применение первых двух шагов к подмассивам слева и права от опорного элемента', 'answer'=>'Последовательное попарное сравнивание элементов и их замена в случае неправильного порядка'),

array('question'=>'Сортировка вставками', 'request'=>'Последовательное попарное сравнивание элементов и их замена в случае неправильного порядка/Элементы входной последовательности просматриваются по одному, и каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов/Рекурсивное разбиение массива на подмассивы, после чего маленькие уже упорядоченные массивы постепенно объединяются с проверкой порядка элементов/Выбор опорного элемента, расположение элементов, меньше опорного слева, а больше него справа. Рекурсивное применение первых двух шагов к подмассивам слева и права от опорного элемента', 'answer'=>'Элементы входной последовательности просматриваются по одному, и каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов'),

array('question'=>'Сортировка слиянием', 'request'=>'Последовательное попарное сравнивание элементов и их замена в случае неправильного порядка/Элементы входной последовательности просматриваются по одному, и каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов/Рекурсивное разбиение массива на подмассивы, после чего маленькие уже упорядоченные массивы постепенно объединяются с проверкой порядка элементов/Выбор опорного элемента, расположение элементов, меньше опорного слева, а больше него справа. Рекурсивное применение первых двух шагов к подмассивам слева и права от опорного элемента', 'answer'=>'Рекурсивное разбиение массива на подмассивы, после чего маленькие уже упорядоченные массивы постепенно объединяются с проверкой порядка элементов'),

array('question'=>'Быстрая сортировка', 'request'=>'Последовательное попарное сравнивание элементов и их замена в случае неправильного порядка/Элементы входной последовательности просматриваются по одному, и каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов/Рекурсивное разбиение массива на подмассивы, после чего маленькие уже упорядоченные массивы постепенно объединяются с проверкой порядка элементов/Выбор опорного элемента, расположение элементов, меньше опорного слева, а больше него справа. Рекурсивное применение первых двух шагов к подмассивам слева и права от опорного элемента', 'answer'=>'Выбор опорного элемента, расположение элементов, меньше опорного слева, а больше него справа. Рекурсивное применение первых двух шагов к подмассивам слева и права от опорного элемента'),

array('question'=>'Какая сортировка применяется только в обучающих целях', 'request'=>'Пузырьком/Быстрая/Вставкми/Выбором/Расческой', 'answer'=>'Пузырьком'),

array('question'=>'Какие сортировки рассчитаны на использование оперативной памяти (произвольный доступ к любой ячейке)', 'request'=>'Внешние/Внутренние/Все/Ни одна', 'answer'=>'Внутренние'),

array('question'=>'Что такое временная сложность', 'request'=>'Время, за которое алгоритм выполнит свою задачу/Время, за которое можно разработать алгоритм/Время, которое алгоритм тратит на одну итерацию', 'answer'=>'Время, за которое алгоритм выполнит свою задачу'),

);

$maxcount = count($questions);

#обработка начала сессии

if (!isset($\_SESSION['start'])) {

$\_SESSION['start'] = 1;

$\_SESSION['step'] = -1;

$\_SESSION['answers'] = array();

$\_SESSION['login'] = '';

}

#обработка кнопок

if(isset($\_POST['next'])) {

if (!empty($\_POST['login'])){

$\_SESSION['login'] = $\_POST['login'];

}

if (empty($\_SESSION['login'])) {

$\_SESSION['step']--;

}

if(isset($\_POST['q'])){

if (strcmp($\_POST['q'], 'null')==0){

$\_SESSION['step']--;

}else{

$\_SESSION['answers'][$\_SESSION['random'][$\_SESSION['step']]]=$\_POST['q'];

}

}

$\_SESSION['step']++;

header("Location: " . $\_SERVER['REQUEST\_URI']);

//return;

}

if ($\_SESSION['step']==-1 || $\_SESSION['step'] > $count ) {

$\_SESSION['random'] = range(0,$maxcount - 1);

shuffle($\_SESSION['random']);

$login = $\_SESSION['login'];

echo 'Введите логин:';

echo "<form method='POST'>

<input type=hidden name=login value=null>

<input type=text name=login placeholder=$login>

<input type=submit name='next' value='Продолжить'>

</form>";

$\_SESSION['step']=-1;

$\_SESSION['answers'] = array();

} else if ($\_SESSION['step']!=$count){

#вопросы с ответами

$step = $\_SESSION['step']+1;

$questionnum = $\_SESSION['random'][$\_SESSION['step']];

echo $questions[$questionnum]['question'];

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

echo "<input type=hidden name=q value='null'>";

$q = explode('/', $questions[$questionnum]['request']);

for ($i = 0;$i < count($q); ++$i){

echo "<input type=radio name=q value='{$q[$i]}'>{$q[$i]}<br>";

}

echo "<input type=submit name='next' value='Продолжить'>

</form>

Вопрос $step из $count<br>";

}else{

$file = fopen($\_SESSION['login'] . '.txt', 'w');

$login = $\_SESSION['login'];

echo "Пользователь: $login<br><br>";

fwrite($file, "Пользователь: $login\n\n");

$\_SESSION['rightAnswer'] = 0;

for ($i = 0; $i < $count; ++$i) {

$index = $\_SESSION['random'][$i];

$answer = $questions[$index]['answer'];

$question = $questions[$index]['question'];

$ii=$i+1;

echo "Вопрос $ii: \"$question\"<br>";

fwrite($file, "Вопрос $ii: \"$question\"\n");

if (strcmp($answer, $\_SESSION['answers'][$index]) == 0) {

echo $answer . " - Верно";

fwrite($file, $answer . " - Верно");

$\_SESSION['rightAnswer']++;

} else {

echo "Неправильный ответ! Правильный ответ: $answer";

fwrite($file, $answer . " - Неверно");

}

echo '<br><br>';

fwrite($file, "\n\n");

}

echo "Оценка по 100-бальной системе: " . ($\_SESSION['rightAnswer'] \* 1.0 / $count) \* 100 . '<br>';

setlocale(LC\_ALL, 'ru\_RU', 'ru\_RU.UTF-8', 'ru', 'russian');

echo "Время прохождения: " . date('l jS \of F Y h:i:s A');

fwrite($file, "Оценка: " . ($\_SESSION['rightAnswer'] \* 1.0 / $count) \* 100);

fwrite($file, "\nВремя прохождения: " . date('l jS \of F Y h:i:s A'));

echo '<br><br>';

fwrite($file, "\n\n");

echo "<form method='POST'> <input type=submit name='next' value='Начать заново'> </form>";

fclose($file);

$zip = new ZipArchive();

$filename = "results.zip";

if ($zip->open($filename, ZipArchive::CREATE)!==TRUE) {

exit("Невозможно открыть <$filename>\n");

}

$zip->addFile($login . '.txt');

$i = 0;

$list = array();

while($name = $zip->getNameIndex($i)) {

$list[$i] = $name;

$i++;

}

echo "Файлы с результатами:<br><br>";

foreach($list as $file) {

echo "<a href='?file=$file'> $file </a><br>";

}

$zip->close();

echo "<a href='?file=results.zip'> results.zip </a><br>";

}

function download($filename) {

$filename = realpath($filename);

$file\_extension = strtolower(substr(strrchr($filename,"."),1));

switch ($file\_extension) {

case "zip": $ctype="application/zip"; break;

default: $ctype="application/force-download";

}

if (!file\_exists($filename)) {

die("NO FILE HERE");

}

header("Pragma: public");

header("Expires: 0");

header("Cache-Control: must-revalidate, post-check=0, pre-check=0");

header("Cache-Control: private",false);

header("Content-Type: $ctype");

header("Content-Disposition: attachment; filename=\"".basename($filename)."\";");

header("Content-Transfer-Encoding: binary");

header("Content-Length: ". filesize($filename));

set\_time\_limit(0);

readfile("$filename");

}

?>