**Отчёт 5 (Яковлев Д.М.)**

**Код**

**Chapter5.php**

<!doctype html>

<html lang=ru>

<head>

    <title>Глава 5: Улучшенная обработка массивов и строк</title>

    <link rel=stylesheet href=Chapter5.css>

</head>

<body>

    <p>Создатель кроссвордов</p>

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

        <p>Название кроссворда</p>

        <input type=text name='name'>

        <br>

        Высота: <input type=text name=height value=10 size=6>

        <br>

        Ширина: <input type=text name=width value=10 size=6>

        <hr class=short>

        <hr class=wide>

        <hr class=short>

        <p>Список слов</p>

        <textarea rows=10 cols=60 name=wordList></textarea>

        <br><br>

        Пожалуйста, вводите только одно слово в каждой строке, без пробелов

        <br>

        <input type=submit value="Создать кроссворд" />

    </form>

</body>

</html>

**Chapter5.css**

body{

    border-style:groove;

    border-radius:30px 25px 20px 35px;

    border-width:10px 30px 10px 30px;

    border-color:rgb(153,153,204);

    background-color:rgb(43,43,43);

    color:rgb(204,204,204);

    letter-spacing:1px;

    word-spacing:4px;

    text-align:center;

    font-family:Cochin;

    font-weight:200;

    font-size:medium;

    font-stretch:expanded;

    font-variant-numeric:oldstyle-nums;

    padding:5px;

}

pre{

    color: inherit;

    background-color: inherit;

    letter-spacing:inherit;

    word-spacing:inherit;

    text-align:inherit;

    font-size:inherit;

    font-weight:inherit;

    font-family:inherit;

    font-stretch:inherit;

    padding:inherit;

    color:inherit;

}

hr.short{

    border-style:ridge;

    border-color:rgb(153,153,204);

    border-radius:2px;

    width:10%;

}

hr.wide{

    border-style:ridge;

    border-color:rgb(153,153,204);

    border-radius:2px;

}

div{

    text-align:justify;

    font-size:inherit;

    font-stretch:inherit;

    font-variant-numeric:inherit;

    font-weight:inherit;

    font-family:inherit;

    word-spacing:inherit;

    letter-spacing:inherit;

}

p{

    font-size:medium;

    font-style:italic;

    font-weight:400;

    letter-spacing:2px;

    font-family:sans-serif;

    font-variant:small-caps;

    text-align:inherit;

    font-variant-numeric:lining-nums;

    color:inherit;

}

**Functions.php**

<?php

function parseList()

{

    global $word, $wordList, $boardData;

    $itWorked = TRUE;

    /\* var\_dump($wordList); \*/

    $wordList = mb\_strtoupper($wordList, "utf-8");

    /\* var\_dump($wordList); \*/

    /\* print $wordList; \*/

    $word = preg\_split("/\n/u", $wordList, -1, PREG\_SPLIT\_NO\_EMPTY);

    /\* var\_dump($word);

    foreach ($word as $currentWord) {

        print "<table border=0 align=center>";

        print <<<TEXT

        <tr>

        <td>$currentWord</tr>

        </tr>

        TEXT;

        print '</table>';

    } \*/

    foreach ($word as $currentWord) {

        $currentWord = rtrim($currentWord);

        var\_dump($currentWord);

        if ((strlen($currentWord)/2 > $boardData["width"]) && (strlen($currentWord)/2 > $boardData["height"])) {

            print "$currentWord - слишком длинное слово для кроссворда<br>";

            $itWorked = false;

        }

    }

    /\* print "<table border=0 align=center>";

    foreach($boardData as $index => $value){

        print <<<HERE

        <tr><td>$index</td>

        <td>$value</td></tr>

        HERE;

    }

    print "</table>"; \*/

    /\* foreach($word as $theWord)

    {

        $theWord = rtrim($theWord);

        print "$theWord: ".strlen($theWord)/2 ."<br>";

    } \*/

     return $itWorked;

}

function clearBoard()

{

    global $board, $boardData;

    for ($row = 0; $row < $boardData["height"]; $row++) {

        for ($col = 0; $col < $boardData["width"]; $col++) {

            $board[$row][$col] = ".";

        }

    }

}

function fillBoard()

{

    global $word;

    $direction = array("E", "W", "S", "N");

    $itWorked = true;

    $counter = 0;

    $keepGoing = true;

    while ($keepGoing) {

        $dir = rand(0, 3);

        $result = addWord($word[$counter], $direction[$dir]);

        if ($result == false) {

            $keepGoing = false;

            $itWorked = false;

        }

        $counter++;

        if ($counter >= count($word)) {

            $keepGoing = false;

        }

    }

    return $itWorked;

}

function addWord($theWord, $dir)

{

    global $board, $boardData;

    $theWord = rtrim($theWord);

/\*     var\_dump($theWord); \*/

    $wordLength = strlen($theWord)/2;/\*

    print "wordLength: $wordLength"; \*/

    $theWord = preg\_split("//u", $theWord, -1);

    /\* var\_dump($theWord); \*/

    $itWorked = true;

    switch ($dir) {

        case "E":

            $newCol = rand(0, $boardData["width"] - 1 - $wordLength);

            $newRow = rand(0, $boardData["height"] - 1);

            for ($i = 1; $i < $wordLength+1; $i++) {

                $boardLetter = $board[$newRow][$newCol + $i - 1];

                $wordLetter = $theWord[$i];

                /\* var\_dump($wordLetter); \*/

                if (($boardLetter == $wordLetter) || ($boardLetter == ".")) {

                    $board[$newRow][$newCol + $i - 1] = $wordLetter;

                } else {

                    $itWorked = false;

                }

            }

            break;

        case "W":

            $newCol = rand($wordLength+1, $boardData["width"] - 1);

            $newRow = rand(0, $boardData["height"] - 1);

            for ($i = 1; $i < $wordLength+1; $i++) {

                $boardLetter = $board[$newRow][$newCol - $i];

                $wordLetter = $theWord[$i];

                if (($boardLetter == $wordLetter) || ($boardLetter == ".")) {

                    $board[$newRow][$newCol - $i] = $wordLetter;

                } else {

                    $itWorked = false;

                }

            }

            break;

        case "S":

            $newCol = rand(0, $boardData["width"] - 1);

            $newRow = rand(0, $boardData["height"] - 1 - $wordLength+1);

            for ($i = 1; $i < $wordLength+1; $i++) {

                $boardLetter = $board[$newRow + $i - 1][$newCol];

                $wordLetter = $theWord[$i];

                if (($boardLetter == $wordLetter) || ($boardLetter == ".")) {

                    $board[$newRow + $i - 1][$newCol] = $wordLetter;

                } else {

                    $itWorked = false;

                }

            }

            break;

        case "N":

            $newCol = rand(0, $boardData["width"] - 1);

            $newRow = rand($wordLength+1, $boardData["height"] - 1);

            for ($i = 1; $i < $wordLength+1; $i++) {

                $boardLetter = $board[$newRow - $i][$newCol];

                $wordLetter = $theWord[$i];

                if (($boardLetter == $wordLetter) || ($boardLetter == ".")) {

                    $board[$newRow - $i][$newCol] = $wordLetter;

                } else {

                    $itWorked = false;

                }

            }

            break;

    }

    return $itWorked;

}

function makeBoard($theBoard)

{

    global $boardData;

    $puzzle = "<table border=0 align=center>";

    for ($row = 0; $row < $boardData["height"]; $row++) {

        $puzzle .= "<tr>";

        for ($col = 0; $col < $boardData["width"]; $col++) {

            $puzzle .= "<td width=15>" . $theBoard[$row][$col] . "</td>";

        }

        $puzzle .= "</tr>\n";

    }

    $puzzle .= "</table>";

    return $puzzle;

}

function addFoils()

{ // Заполняет поле случайными буквами

    global $board, $boardData;

    for ($row = 0; $row < $boardData["height"]; $row++) {

        for ($col = 0; $col < $boardData["width"]; $col++) {

            if ($board[$row][$col] == ".") {

                $newLetter = rand(192, 223);

                $cyrillic = iconv("windows-1251//IGNORE", "UTF-8//IGNORE", chr($newLetter));

                $board[$row][$col] = $cyrillic;

            }

        }

    }

}

function printPuzzle()

{ // Печатает кроссворд

    global $puzzle, $word, $keyPuzzle, $boardData;

    print "<h1>" . $boardData["name"] . "</h1>";

    print <<<TEXT

    <center>

    $puzzle

    <h3>Список слов</h3>

    <table border=0>

    TEXT;

    foreach ($word as $theWord) {

        print "<tr><td>$theWord</td></tr>\n";

    }

    print "</table></center>\n";

    $puzzleName = $boardData["name"];

    print <<<HERE

    <br><br><br><br><br><br><br><br>

    <form action="wordHuntKey.php"

    method="post">

    <input type=hidden name=key

    value=$keyPuzzle>

    <input type=hidden name=puzzleName

    value=$puzzleName>

    <input type=submit value="Показать ключ ответов">

    </form>

    HERE;

}

?>

**wordHunt.php**

<!doctype html>

<html lang=ru>

<head>

    <title>Кроссворд</title>

    <link rel=stylesheet href=Chapter5.css>

</head>

<body>

    <?php

    include("functions.php");

    extract($\_REQUEST);

    extract($\_POST);

    if ($wordList == NULL) {

        $word = array(

            "ДОМ",

            "ШИМПАНЗЕ",

            "ЯНТАРЬ",

            "СОЛОНИНА",

            "ГИРЛЯНДА"

        );

        $boardData = array(

            "width" => 10,

            "height" => 10,

            "name" => "Шаблонный кроссворд",

        );

    } else {

        $boardData = array(

            "width" => $width,

            "height" => $height,

            "name" => $name,

        );

        if (parseList()) {

            $legalBoard = false;

            while (!$legalBoard) {

                clearBoard();

                $legalBoard = fillBoard();

            }

            $key = $board;

            // var\_dump($key);

            $keyPuzzle = makeBoard($key);

            addFoils();

            $puzzle = makeBoard($board);

            printPuzzle();

        }

    }

    ?>

</body>

</html>

**wordHuntKey.php**

<!doctype html>

<html>

<head>

    <title>Ключи к кроссворду</title>

    <link rel=stylesheet href=Chapter5.css>

</head>

<body>

    <?php

    extract($\_REQUEST);

    extract($\_POST);

    print <<<HERE

    <h1>$puzzleName. Ключи к кроссворду</h1>

    $key

    HERE;

    ?>

    <?php

    print <<<TEXT

    <table border=1 cellpadding=3 cellspacing=4 align=center>

    <tr>

    <th>Индекс</th>

    <th>Значение</th>

    </tr>

    TEXT;

    foreach ($\_REQUEST as $index => $value) {

        print <<<TEXT

    <tr>

    <td>$index</td>

    <td>$value</td>

    </tr>

    TEXT;

    }

    print '</table>';

    ?>

</body>

</html>

**Скриншоты**

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, компьютер, программное обеспечение

Автоматически созданное описание