Министерство образования и науки РФ

ФГБОУ ВПО «Тамбовский государственный технический университет»

Кафедра «САПР»

Лабораторная работа №4

по дисциплине «Интернет технологии»

«PHP сценарий»

Выполнил: студент гр. qwinmen

Проверил: А. А.

Тамбов,

**Цель работы:** приобретение навыков работы со скриптовым языком PHP, написание простейших функций.

**Задание:**

Разработать серверный php-сценарий, реализующий определенную вычислительную задачу (выбор произвольный, рекомендуются вычислительные задачи из других дисциплин).

Основные условия:

* использование html-форм для инициализации запроса к серверному сценарию и передачи параметров для решения задачи (реализация связи html-страница → php-сценарий);
* работа с массивами;
* работа с файлами.

Решение:

Так как язык php я не изучал, то необходимы основы, чтоб сформировать минимальное представление о синтаксисе, правилах, стандартах. Благо об этом позаботился заранее - <http://www.codecademy.com/> прекрасный ресурс для обучения основам таких языков, как:

* [HTML & CSS](http://www.codecademy.com/tracks/web)
* [Javascript](http://www.codecademy.com/tracks/javascript)
* [jQuery](http://www.codecademy.com/tracks/jquery)
* [Python](http://www.codecademy.com/tracks/python)
* [Ruby](http://www.codecademy.com/tracks/ruby)
* [PHP](http://www.codecademy.com/tracks/php)

Потратив пару дней на прохождение всего php квеста уже имел кое-что для выполнения простых алгоритмов.

Найденный за второй курс в папке вычмата ~~работающий~~ метод Симпсона прекрасно подошел под критерий «реализовать алгоритм». По традиции сначала итоговый код html (Таблица 1) и php (Таблица 2).

Таблица 1. Код HTML

|  |
| --- |
| <div style="background-color: Oldlace">  <hr color='Lightpink'><p align=center>LABAr 4 PHP</p><hr color='Lightpink'>      <table border=1 cellspacing=0 cellpadding=0 style='border-collapse:collapse;border:none;mso-border-alt:solid black .5pt;'>  <tr>  <td width=600 colspan=4> |

Таблица 1. Код HTML продолжение

|  |
| --- |
| <p align="center"><span style="font-weight: bold">Задание:</span> произвести численное интегрирование нескольких функций по правилу Симпсона.</p>  </td>  </tr>    <form name="formPHPin" method=POST>    <tr>  <td><input type=radio name="fun" value="1" checked>1 / (1 + p \* p) </td>  <td><input type=radio name="fun" value="2">2 / (1 + p \* p) </td>  <td><input type=radio name="fun" value="3">3 / (1 + p \* p) </td>  </tr>    <tr>  <td colspan=3>Отрезок интегрирования [a,b] &rarr; (a) =<input type=text name="intA" size=70></td>  </tr>    <tr>  <td colspan=3>Отрезок интегрирования [a,b] &rarr; (b) =<input type=text name="intB" size=70></td>  </tr>    <tr>  <td colspan=3>На сколько частей нужно разделить отрезок? n=<input type=text name="intN" size=58></td>  </tr>    <tr bgcolor='red'><td colspan=4 align=center><input type=submit value="[&Xi;&Xi;&Xi;&Xi;OK&Xi;&Xi;&Xi;&Xi;]" name="btn"></td></tr>    </form>    <tr><td colspan=3><?php include("algoritmic.php"); ?><?php include("tmp.php"); ?>  <?php include("btnCh.php"); ?></td></tr>    </table>        </div> |

Таблица . Код PHP

|  |
| --- |
| <?php    $a = $\_POST['intA'];  $b = $\_POST['intB'];  $n = $\_POST['intN'];  $f = $\_POST['fun'];      function result($a, $b, $n, $f)  {  $s = 0.0; $index = 0.0; |

Таблица 2. Код PHP продолжение

|  |
| --- |
| $h = ($b - $a)/$n;  $x = $a + $h;  $srt = "";    while ($x < $b)  {  $s = $s + (4.0 \* Y($x,$f));  $str = $str."Площадь $index равна $s";  $x = $x + $h;  $s = $s + (2.0 \* Y($x,$f));  $index = $index + 1.0;  $str = $str."\tПлощадь $index равна $s\r\n";  $x = $x + $h;  }  $s = $h / 3.0 \* ($s + Y($a,$f) - Y($b,$f));  return $str."\t\tИнтеграл = $s\r\n";  }    function Y($p, $f)  {  switch ($f)  {  case 1:  return 1.0 / (1.0 + $p \* $p);  break;  case 2:  return 2.0 / (1.0 + $p \* $p);  break;  case 3:  return 3.0 / (1.0 + $p \* $p);  break;  default:  return 4.0 / (2.0 + $p \* $p);  }    }    if($a=='' || $b=='' || $n=='') print "<textarea cols=72 rows=5>".result(0.4,1.2,8.0, $f)."</textarea>";  else print "<textarea cols=72 rows=5>".result($a, $b, $n, $f)."</textarea>"; |

Принцип такой – на странице html размещен блок, в блоке форма с методом POST(то есть отправка данных скрыта). На форме размещены input компоненты типа text и кнопка submit для отправки данных в глобальный массив \_POST[] на стороне скрипта. Дальше дело техники, ввел числа в поля, нажал на подтверждение и работает скрипт.

Принцип такой – в отдельном файле.php, дабы не засорять html страницу, объявляем переменные a-b-n-f со значениями глобального массива ( в нем ориентируемся на значение поля name тегов input) и результатами равными значениям value ( введенные значения в текстовых полях формы). Далее, ради эксперимента, объявляем функцию result(), принимающую назначенные выше переменные. Опускаю объяснения операции вычитания, сложения и т.д., единственное место с «непоняткой», это Y( ) – простая функция выбора «по какой математике считать», применил ради всё того же эксперимента.

Собственно точкой входа служит запись выбора «направления» прорешивания. Если на форме хоть что-то пусто, то делаем заранее подготовленную операцию с выводом результатов в текстовый блок textarea. Иначе тоже самое, но с опросом введенных данных.

Итог этому стал рисунок 1.

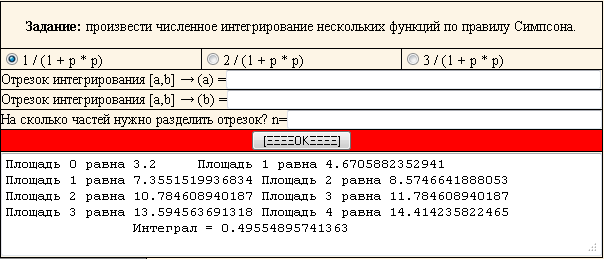


Рисунок . Если на форме что-то пусто, то выполним подготовленный блок

На рисунке 2 отображен результат прорешивания с вводом исходных значений на форме.

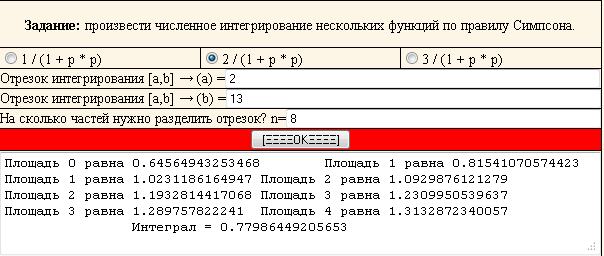


Рисунок . Иначе тот же расчет, но с введенными данными

Реализовав эту всю ~~белиберду~~ часть, стало понятно, что не хватает работы с массивами и файлами. Приспичило написать что-то посерьёзней, сначала надумал игру «морской бой», но хорошо подумав – а как это «рисовать» на форме и столкнувшись с множеством ~~глубинных~~ подводных камней, отказался от этой идеи. Следующим на очереди всплыла игра «пятнашки»! Хм, почему нет, кажется простым – таблицу, для начала 3х3, кнопки в ячейки и считай игра готова… В итоге проблем возникло более чем достаточно (~~еще и в сети ничего такого нет~~).

В общем, сперва код html таблица 3.

Таблица . Код HTML

|  |
| --- |
| <div>  <table border=1 bgcolor="green">  <form name=BtnCh method=POST><!--Отдельно form дабы нерушить созданный на фоне php класс-->  <tr>  <td><button style="width:40px;height:40px" name="btn1" id="1" value=1 ><?php if($work->btn1 == "" && $DSbtn1 == "") print $gloB1;else print $work->btn1."".$DSbtn1; ?></button></td>  <td><button style="width:40px;height:40px" name="btn2" id="2" value=2 ><?php if($work->btn2 == "" && $DSbtn2 == "") print $gloB2;else print $work->btn2."".$DSbtn2; ?></button></td>  <td><button style="width:40px;height:40px" name="btn3" id="3" value=3 ><?php if($work->btn3 == "" && $DSbtn3 == "") print $gloB3;else print $work->btn3."".$DSbtn3; ?></button></td>    </tr>  <tr>  <td><button style="width:40px;height:40px" name="btn4" id="4" value=4 ><?php if($work->btn4 == "" && $DSbtn4 == "") print $gloB4;else print $work->btn4."".$DSbtn4; ?></button></td>  <td><button style="width:40px;height:40px" name="btn5" id="5" value=5 ><?php if($work->btn5 == "" && $DSbtn5 == "") print $gloB5;else print $work->btn5."".$DSbtn5; ?></button></td>  <td><button style="width:40px;height:40px" name="btn6" id="6" value=6 ><?php if($work->btn6 == "" && $DSbtn6 == "") print $gloB6;else print $work->btn6."".$DSbtn6; ?></button></td>  </tr>  <tr>  <td><button style="width:40px;height:40px" name="btn7" id="7" value=7 ><?php if($work->btn7 == "" && $DSbtn7 == "") print $gloB7;else print $work->btn7."".$DSbtn7; ?></button></td>  <td><button style="width:40px;height:40px" name="btn8" id="8" value=8 ><?php if($work->btn8 == "" && $DSbtn8 == "") print $gloB8;else print $work->btn8."".$DSbtn8; ?></button></td>  <td><button style="width:40px;height:40px" name="btn-" id="9" value=9><?php if($gloB9==0) print "";else print $gloB9; ?></button></td>  </tr>  <tr bgcolor='red'>  <td colspan=4 align=center style='background:red;'> |

Таблица 3. Код HTML продолжение

|  |
| --- |
| <button name="newGame" id="123" value=123 >[&Xi;&Xi;&Xi;&Xi;New game&Xi;&Xi;&Xi;&Xi;]</button><br>    <button name="btnSAVE" id="200" value=200 >[&Xi;&Xi;&Xi;&Xi;Save state&Xi;&Xi;&Xi;&Xi;]</button><br>    <button name="btnLOAD" id="300" value=300>[&Xi;&Xi;&Xi;&Xi;Load state&Xi;&Xi;&Xi;&Xi;]</button><br>    </td>  </tr>  </tr>  </form>  <!--нажатие на кнопку "Новая игра" посылает в php, там рандомом назначим числа кнопкам,  на html при помощи js опросим php назначеные кнопки  </table>  </div> |

Тут главное неспешить, так как эти дебри писались не за одну ночь

В начале был блок. В нем - форма с методом POST, в форме – таблица, в таблице – кнопки, в кнопках – переменные с файлов php. Назначены всем имена, значения – ведь по ним будем определять компонент.

Теперь начинается самое главное – модули php, каждый логический блок вынесен в отдельный файл и подключается командой includ. Файлов аж три, вот на меньшее меня нехватило. Коротко о каждом:

* algoritmic.php – некая начальная точка в игре, делаем расчет рандомных чисел от 1 до 8, сохраняем в массив( ). Содержимое этого массива мы видим в самом начале игры, нажав кнопку «Новая игра». Тут же создается temp.txt файл, выполняющий роль буфера для общения между разными классами\файлами.php, этот файл содержит рассчитанный массив( ) с рандомными числами.
* tmp.php – вторая по очередности выполнения точка в игре, делаем сохранение текущей игры в отдельный файл save.txt, и тут же решаем проблему сброса чисел на кнопках (класс из первого файла.php уже отработал, все состояния равны нулю\пустоте), тут выручает файл-буфер temp.txt с последними активными значениями формы\класса. На создание опций «Загрузить сохранку» меня нехватило и так наворотил.
* btnCh.php – последняя и самая активная точка в игре. Содержит «движок» игры, логику перестановки местами двух чисел (ноля и любого другого числа кроме самого нуля☺ ). Также идет активная работа с файл-буфером temp.txt

Код первого файла в таблице 4.

Таблица . Код algoritmic.php

|  |
| --- |
| /\*--------------------------------------------------------------------------------  //+----+----+----+  //| 1 | 2 | 3 |  //+--------------+  //| 4 | 5 | 6 |  //+--------------+  //| 7 | 8 | |  //+----+----+----+  \*/    $btn1 = ""; $btn5 = "";  $btn2 = ""; $btn6 = "";  $btn3 = ""; $btn7 = "";  $btn4 = ""; $btn8 = "";  $btn0 = "";      $globalArr = "";    class DataAlg  {  public $arr = array();  public static $flag;    public function \_\_construct()  {  $this->globalArr = $this->newGame();  }      //# Функция вернет массив рандомных чисел  public function newGame()  {//Заполнить массив рандомом без совпадений:  $tmp = 0;  for($i=1; $i < 8; $i++)  {  if($i==1) {$arr[0] = rand(1,8); $tmp = $arr[0];}    //пройти от начала до конца  for($j=0; $j<count($arr);$j++)  {  //если tmp уже есть  while($tmp==$arr[$j])  {  //то опять рандомить  $tmp = rand(1,8);  $j=0;  }  }  //в итоге получим уникальное число  $arr[$i] = $tmp;  }    //sort($arr);  //foreach($arr as $key)  //echo $key;  //echo count($arr);  $this->btn1 = $arr[0]; $this->btn5 = $arr[4];  $this->btn2 = $arr[1]; $this->btn6 = $arr[5];  $this->btn3 = $arr[2]; $this->btn7 = $arr[6];  $this->btn4 = $arr[3]; $this->btn8 = $arr[7];  $this->btn0 = "-";  //  array\_push($arr, "0");//добавить в конец массива    return $arr;      }      }//end class      //получаем нажатие кнопки New Game  $startNewGame = $\_POST['newGame'];    //если нажали New Game то инициализируем класс  if($startNewGame==123) {$work = new DataAlg();}        /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Формируем промежуточный файл\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  touch( "temp.txt" );  $fop = fopen( "temp.txt", "r+" ) or die ( "Не удалось открыть файл" );  //на начало  fseek($fop,0);  if(count($work->globalArr)>3)  //перезаписать  foreach($work->globalArr as $key)  fputs( $fop, $key );//пишем в одну строку  fclose( $fop );  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Формируем промежуточный файл\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/      ?> |

Код второго файла в таблице 5.

Таблица . Код tmp.php

|  |
| --- |
| <?php    class DataSave  {  public $svArr = array();    public function \_\_construct()  {  $svArr[0] = ""; unset($svArr[0]);//удалить масив\элемент  //выдернем текущее поле из файла, хз как обработать незатирающиеся массивы  $efp = fopen( "temp.txt", "r+" ) or die ( "Не удалось открыть файл" );  while (!feof ( $efp ) )  {  $char = ( fgetc( $efp ) );  if($char < 9 && $char > -1)  array\_push($svArr, $char);//добавить в конец массива  }  fclose( $efp );    $this->save(true, $svArr);  //$btn1 = 34;  //print $this->btn1;    }        //сохранить содержимое $arr в файл  public function save($stateJS, $svArr)  {  global $DSbtn1, $DSbtn2, $DSbtn3, $DSbtn4,  $DSbtn5, $DSbtn6, $DSbtn7, $DSbtn8;    if($stateJS)  {  //Если значение a меньше b, то возвращается a, иначе возвращается b  $DSbtn1 = ( $arrToForm[1] == 0 ? "" : $arrToForm[1]);$DSbtn2 = ( $arrToForm[2] == 0 ? "" : $arrToForm[2]);  $DSbtn3 = ( $arrToForm[3] == 0 ? "" : $arrToForm[3]);$DSbtn4 = ( $arrToForm[4] == 0 ? "" : $arrToForm[4]);  $DSbtn5 = ( $arrToForm[5] == 0 ? "" : $arrToForm[5]);$DSbtn6 = ( $arrToForm[6] == 0 ? "" : $arrToForm[6]);  $DSbtn7 = ( $arrToForm[7] == 0 ? "" : $arrToForm[7]);$DSbtn8 = ( $arrToForm[8] == 0 ? "" : $arrToForm[8]);    touch( "save.txt" );  //создать файл-контейнер чтение и запись. Указатель файла устанавливается на его начало  $fp = fopen( "save.txt", "r+" ) or die ( "Не удалось открыть файл" );  //на начало  fseek($fp,0);  //перезаписать  foreach($svArr as $key)  fputs( $fp, $key );  //echo "->".$svArr[2];    //закрыть файл-контейнер  fclose( $fp );  //return "SAVEDDDD.....";  }//else return "NOT";  }      }    //получаем нажатие кнопки Save game  $saveGame = $\_POST['btnSAVE'];    //если нажали Save Game то инициализируем класс  if($saveGame==200) {$sBtn = new DataSave();}      ?> |

Код третьего файла в таблице 6.

Таблица . Код btnCh.php

|  |
| --- |
| <?php  // print "интернет технологии среда 5 число<br>";  $btn11 = $\_POST['btn1']; $btn55 = $\_POST['btn5'];  $btn22 = $\_POST['btn2']; $btn66 = $\_POST['btn6'];  $btn33 = $\_POST['btn3']; $btn77 = $\_POST['btn7'];  $btn44 = $\_POST['btn4']; $btn88 = $\_POST['btn8'];  $btn00 = $\_POST['btn-'];  $btnArr = array($btn11, $btn22, $btn33, $btn44, $btn55, $btn66, $btn77, $btn88, $btn00);  foreach($btnArr as $key)  { if($key) $tempVal = $key; }    //если нажали btn1 то инициализируем класс  $cBtn = new ChBtn($tempVal);    class ChBtn  {    public function \_\_construct($tmpVal)  {  //touch( "temp.txt" );  //создать файл-контейнер чтение и запись. Указатель файла устанавливается на его начало  $fp = fopen( "temp.txt", "r+" ) or die ( "Не удалось открыть файл" );  $string = "";  while (!feof ( $fp ) )  $string = ( fgets( $fp, 1024 ) );  //закрыть файл-контейнер  fclose( $fp );  //выдернуть\вырезать из СТРОКИ цифру на позиции $tmpVal  $tempChar = array();  $tempChar = str\_split($string);//строку перегнать в массив  //($tmpVal-1)=1\_кнопка и цифра в файле=$tempChar[]    //нажатая кнопка с цыфрой:  $digitBtn = $tempChar[$tmpVal-1];  $ziroPoz = "";  //положение ноля в файле:  $ziroPoz = strrpos($string, "0");  if ($ziroPoz === false) { print "<p>Положение 0 не найдено</p>"; }  else  switch ($ziroPoz)  {  case 0:  {  //if($tmpVal!=1){//типо не нажали на сам ноль  //если расстояние от нажатой кнопки до ноля равно 1+  if($tempChar[($tmpVal-1)-1] == 0 && ( (($tmpVal-1)-1) >=0 && (($tmpVal-1)-1) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 3+  if($tempChar[($tmpVal-1)-3] == 0 && ( (($tmpVal-1)-3) >=0 && (($tmpVal-1)-3) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если $stateP = true то поменять местами 0 И нажатая\_кнопа  //-1: в файле //-2: на форме  if($stateP) $this->toForm($this->replaceDigit($digitBtn));  //}  }  break;  case 1:  {  //if($tmpVal!=2){//типо не нажали на сам ноль  //если расстояние от нажатой кнопки до ноля равно 1+  if($tempChar[($tmpVal-1)-1] == 0 && ( (($tmpVal-1)-1) >=0 && (($tmpVal-1)-1) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 1- //интервал [0..8]  if($tempChar[($tmpVal-1)+1] == 0 && ( (($tmpVal-1)+1) >=0 && (($tmpVal-1)+1) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 3+  if($tempChar[($tmpVal-1)-3] == 0 && ( (($tmpVal-1)-3) >=0 && (($tmpVal-1)-3) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если $stateP = true то поменять местами 0 И нажатая\_кнопа  //-1: в файле //-2: на форме  if($stateP) $this->toForm($this->replaceDigit($digitBtn));  //}  }  break;  case 2:  {  //if($tmpVal!=3){//типо не нажали на сам ноль  //если расстояние от нажатой кнопки до ноля равно 1-  if($tempChar[($tmpVal-1)+1] == 0 && ( (($tmpVal-1)+1) >=0 && (($tmpVal-1)+1) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 3+  if($tempChar[($tmpVal-1)-3] == 0 && ( (($tmpVal-1)-3) >=0 && (($tmpVal-1)-3) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если $stateP = true то поменять местами 0 И нажатая\_кнопа  //-1: в файле //-2: на форме  if($stateP) $this->toForm($this->replaceDigit($digitBtn));  //}  }  break;  case 3:  {  //if($tmpVal!=4){//типо не нажали на сам ноль  //если расстояние от нажатой кнопки до ноля равно 1+  if($tempChar[($tmpVal-1)-1] == 0 && ( (($tmpVal-1)-1) >=0 && (($tmpVal-1)-1) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 3+  if($tempChar[($tmpVal-1)-3] == 0 && ( (($tmpVal-1)-3) >=0 && (($tmpVal-1)-3) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 3-  if($tempChar[($tmpVal-1)+3] == 0 && ( (($tmpVal-1)+3) >=0 && (($tmpVal-1)+3) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если $stateP = true то поменять местами 0 И нажатая\_кнопа  //-1: в файле //-2: на форме  if($stateP) $this->toForm($this->replaceDigit($digitBtn));  //}  }  break;  case 4:  {  //if($tmpVal!=5){//типо не нажали на сам ноль  //если расстояние от нажатой кнопки до ноля равно 1+  if($tempChar[($tmpVal-1)-1] == 0 && ( (($tmpVal-1)-1) >=0 && (($tmpVal-1)-1) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 1-  if($tempChar[($tmpVal-1)+1] == 0 && ( (($tmpVal-1)+1) >=0 && (($tmpVal-1)+1) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 3+  if($tempChar[($tmpVal-1)-3] == 0 && ( (($tmpVal-1)-3) >=0 && (($tmpVal-1)-3) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 3-  if($tempChar[($tmpVal-1)+3] == 0 && ( (($tmpVal-1)+3) >=0 && (($tmpVal-1)+3) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если $stateP = true то поменять местами 0 И нажатая\_кнопа  //-1: в файле //-2: на форме  if($stateP) $this->toForm($this->replaceDigit($digitBtn));  //}  }  break;  case 5:  {  //if($tmpVal!=6){//типо не нажали на сам ноль  //если расстояние от нажатой кнопки до ноля равно 1-  if($tempChar[($tmpVal-1)+1] == 0 && ( (($tmpVal-1)+1) >=0 && (($tmpVal-1)+1) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 3+  if($tempChar[($tmpVal-1)-3] == 0 && ( (($tmpVal-1)-3) >=0 && (($tmpVal-1)-3) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 3-  if($tempChar[($tmpVal-1)+3] == 0 && ( (($tmpVal-1)+3) >=0 && (($tmpVal-1)+3) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если $stateP = true то поменять местами 0 И нажатая\_кнопа  //-1: в файле //-2: на форме  if($stateP) $this->toForm($this->replaceDigit($digitBtn));  //}  }  break;  case 6:  {  //if($tmpVal!=7){//типо не нажали на сам ноль  //если расстояние от нажатой кнопки до ноля равно 1+  if($tempChar[($tmpVal-1)-1] == 0 && ( (($tmpVal-1)-1) >=0 && (($tmpVal-1)-1) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 3-  if($tempChar[($tmpVal-1)+3] == 0 && ( (($tmpVal-1)+3) >=0 && (($tmpVal-1)+3) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если $stateP = true то поменять местами 0 И нажатая\_кнопа  //-1: в файле //-2: на форме  if($stateP) $this->toForm($this->replaceDigit($digitBtn));  //}  }  break;  case 7:  {  //if($tmpVal!=8){//типо не нажали на сам ноль  //если расстояние от нажатой кнопки до ноля равно 1+  if($tempChar[($tmpVal-1)-1] == 0 && ( (($tmpVal-1)-1) >=0 && (($tmpVal-1)-1) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 1-  if($tempChar[($tmpVal-1)+1] == 0 && ( (($tmpVal-1)+1) >=0 && (($tmpVal-1)+1) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 3-  if($tempChar[($tmpVal-1)+3] == 0 && ( (($tmpVal-1)+3) >=0 && (($tmpVal-1)+3) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если $stateP = true то поменять местами 0 И нажатая\_кнопа  //-1: в файле //-2: на форме  if($stateP) $this->toForm($this->replaceDigit($digitBtn));  //}  }  break;  case 8:  {  //if($tmpVal!=9){//типо не нажали на сам ноль  //если расстояние от нажатой кнопки до ноля равно 1-  if($tempChar[($tmpVal-1)+1] == 0 && ( (($tmpVal-1)+1) >=0 && (($tmpVal-1)+1) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если расстояние от нажатой кнопки до ноля равно 3-  if($tempChar[($tmpVal-1)+3] == 0 && ( (($tmpVal-1)+3) >=0 && (($tmpVal-1)+3) <=8) )//-->0?  $stateP = true;//типо можно двигать  //если $stateP = true то поменять местами 0 И нажатая\_кнопа  //-1: в файле //-2: на форме  if($stateP) $this->toForm($this->replaceDigit($digitBtn));    //}  }  break;  default: echo 'The value is NOT';  }    //если флаг==false то нажали НОЛЬ или ЛЕВУЮ\_ЦИФРУ  if(!$stateP) $this->toForm($tempChar);//выводим старый массив без перестановок    }    //# поменять местами ноль и цифру в файле и на форме  public function replaceDigit($digit)  {  $fop = fopen( "temp.txt", "r+" ) or die ( "Не удалось открыть файл" );  $string = "";  while (!feof ( $fop ) )  $string = ( fgets( $fop, 1024 ) );  $newArr = array();  $newArr = str\_split($string);//строку перегнать в массив  //найти ноль, найти цифру и поменять  for($i=0; $i < count($newArr); $i++)  {  $tmp = $newArr[$i];  if($tmp == $digit)  {  $newArr[$i] = 0;  //если 8 правее чем 0:  if(strrpos($string, $digit) > strrpos($string, "0")) $i=0;else $i=$i+1;  for(; $i < count($newArr); $i++)  {  $tmp = $newArr[$i];  if($tmp == 0)  {  $newArr[$i] = $digit;  break;  }  }  }  }    //print\_r( $newArr);  //на начало  fseek($fop,0);  //перезаписать  foreach($newArr as $key)  {fputs( $fop, $key );}    //закрыть файл-контейнер  fclose( $fop );  return $newArr;  }    public function toForm($arrToForm)  {  global $gloB1;global $gloB2;global $gloB3;  global $gloB4;global $gloB5;global $gloB6;  global $gloB7;global $gloB8;global $gloB9;    //Если значение a меньше b, то возвращается a, иначе возвращается b  $gloB1 = ( $arrToForm[0] == 0 ? "" : $arrToForm[0]);$gloB2 = ( $arrToForm[1] == 0 ? "" : $arrToForm[1]);  $gloB3 = ( $arrToForm[2] == 0 ? "" : $arrToForm[2]);$gloB4 = ( $arrToForm[3] == 0 ? "" : $arrToForm[3]);  $gloB5 = ( $arrToForm[4] == 0 ? "" : $arrToForm[4]);$gloB6 = ( $arrToForm[5] == 0 ? "" : $arrToForm[5]);  $gloB7 = ( $arrToForm[6] == 0 ? "" : $arrToForm[6]);$gloB8 = ( $arrToForm[7] == 0 ? "" : $arrToForm[7]);  $gloB9 = ( $arrToForm[8] == 0 ? "" : $arrToForm[8]);    $etalon = array(1, 2, 3, 4, 5, 6, 7, 8, 0); $index = 0;  //если в массиве все цифры по порядку от 1 до 9 ТО ПОБЕДА  for($i=0;$i<count($arrToForm); $i++)  {  if($arrToForm[$i] == $etalon[$i]) $index++;//сколько совпадений  }  if($index == count($etalon))  print "<h2 style=\"border-style:dashed;width:140px;\">&nbsp;&nbsp;YOU WIN!</h2>";    }    }  ?> |

Итогом этого брейншторма стали рисунки 3, 4.

Рисунок 3. Процесс игры Рисунок 4. Конец игры

Как вариант развития юзабилити игры, можно прицепить картинки к кнопкам, делать подсчет числа перезаписи файла temp.txt для ведения статистики сделанных ходов.