Міністерство освіти і науки України

Тернопільський національний економічний університет

Факультет комп’ютерних інформаційних технологій

Кафедра комп’ютерних наук

**Лабораторна робота №3**

З дисципліни:

«Web-програмування»

**Виконав:**

студент групи ІПЗ-32

Фатюк В.І

**Перевірила:**

викл. Дарморост І.А.

Тернопіль 2018

**Варіант №16**

**Тема роботи:** Створення сценаріїв на мові PHP при розв’язанні типових та нестандартних завдань .

**Мета роботи:** отримати навички створення сценаріїв на мові РНР з використанням принципів об’єктно-орієнтованого програмування.

**Виконання роботи:**

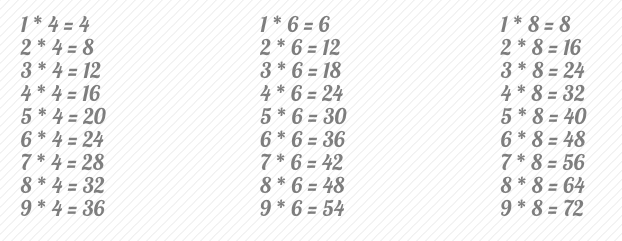
**Завдання 1:**

Створіть клас для виведення таблиці множення для вказаного числа (передавати в конструкторі). Створити окремий метод для обчислення. Далі створити кілька об'єктів даного класу для демонстрації працездатності класу. Висновок оформити у вигляді таблиці.

**<?php  
  
class** MultiplicationTable {  
  
 **public** $value;  
 **function** \_\_construct($value) {  
 $this->value = $value;  
 }  
  
 **public function** getMultiplicationTable() {  
 $result\_array = **array**();  
 **for** ($i=0; $i <= 9; $i++) array\_push($result\_array, $i \* $this->value);  
 **return** $result\_array;  
 }  
  
 **public function** renderResultTable() {  
 **for** ($i=1; $i < count($this->getMultiplicationTable()); $i++) {  
 **echo** $i;  
 **echo " \* "**;  
 **echo** $this->value;  
 **echo " = "**;  
 **echo** $this->getMultiplicationTable()[$i];  
 **echo "<br>"**;  
 }  
 }  
  
}  
**?>**

<**div class="content"**>  
 <**div**>  
 **<?php** $table1 = **new** MultiplicationTable(**"4"**);  
 $table1 -> renderResultTable();  
 **?>** </**div**>  
 <**div**>  
 **<?php** $table1 = **new** MultiplicationTable(**"6"**);  
 $table1 -> renderResultTable();  
 **?>** </**div**>  
 <**div**>  
 **<?php** $table1 = **new** MultiplicationTable(**"8"**);  
 $table1 -> renderResultTable();  
 **?>** </**div**>

**Рисунок 1.1:** Виконання завдання мовою PHP+HTML



**Рисунок 1.2:** Результат роботи програми

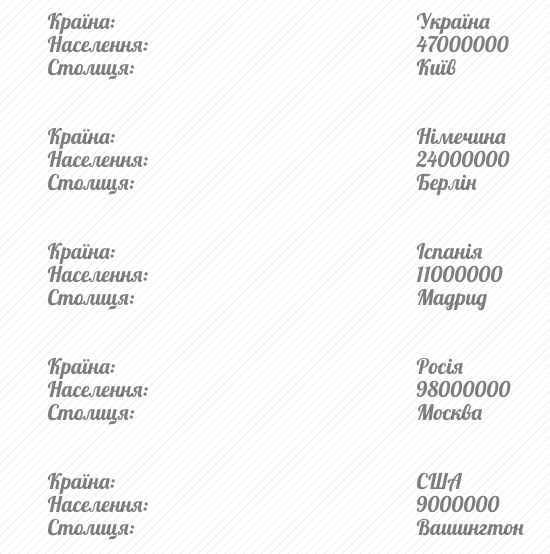
**Завдання 2:**

Створіть клас країни в якому будуть поля: назва країни, населення і назва столиці (англійські назви придумайте самостійно). Створіть масив об'єктів, виведіть кожний з них у таблицю в три рядки по дві комірки в кожному. У лівій комірці ім'я елемента, в правій - його значення.

**<?php  
  
class** Country {  
  
 **public $name**;  
 **public $populations**;  
 **public $capital**;  
 **function** \_\_construct($name, $populations, $capital) {  
 $this -> **name** = $name;  
 $this -> **populations** = $populations;  
 $this -> **capital** = $capital;  
 }  
  
}  
**?>  
  
<?php**$array\_with\_country = [**'Україна'**, **'Німечина'**, **'Іспанія'**, **'Росія'**, **'США'**];  
$array\_with\_population = [47000000, 24000000, 11000000, 98000000, 9000000];  
$array\_with\_capital = [**'Київ'**, **'Берлін'**, **'Мадрид'**, **'Москва'**, **'Вашингтон'**];  
  
$object\_array = **array**();  
**for** ($i=0; $i < *count*($array\_with\_country); $i++) {  
 *array\_push*($object\_array, **new** Country($array\_with\_country[$i], $array\_with\_population[$i], $array\_with\_capital[$i]));  
}  
  
**?>**

<**div class="result"**>  
 **<?php  
 for** ($i=0; $i < *count*($object\_array); $i++) {  
 **echo "<div>Країна:</div>"**; **echo "<div>"**; **echo** $object\_array[$i]->**name**; **echo "</div>"**;  
 **echo "Населення: "**; **echo "<div>"**; **echo** $object\_array[$i]->**populations**; **echo "</div>"**;  
 **echo "<div>Столиця:</div>"**; **echo "<div>"**; **echo** $object\_array[$i]->**capital**; **echo "</div>"**;  
 **echo "<br> </br>"**;  
 **echo "<div> </div>"**;  
 }  
 **?>**</**div**>

**Рисунок 2.1:** Виконання завдання мовою PHP+HTML



**Рисунок 2.2:** Результат роботи програми

**Завдання 3:**

Створіть клас користувача, з полями: прізвище, ім'я, вік і e-mail.

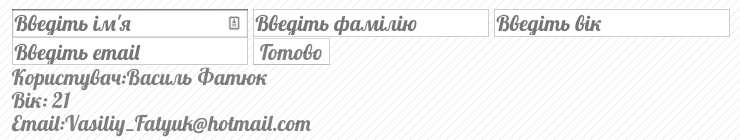
У HTML формі користувач вводить в чотири різні поля: прізвище, ім'я, вік і e-mail. Після натискання клавіші кнопки ГОТОВО створюється об'єкт користувача, з методом, який вносить ці дані в поле об'єкту і далі виводить їх використовуючи другий метод класу користувача.

У формі передбачити перевірку, що всі поля перед відправленням не порожні.

**<?php  
  
class** User {  
  
 **private $name** = **""**;  
 **private $surname** = **""**;  
 **private $age** = **""**;  
 **private $email** = **""**;  
 **function** \_\_construct($name,$surname, $age, $email){  
 $this -> **name** = $name;  
 $this -> **age** = $age;  
 $this -> **email** = $email;  
 $this -> **surname** =$surname;  
 }  
  
 **public function** parametrValidation() {  
 **if**($\_SERVER[**'REQUEST\_METHOD'**] == **"POST"** && **isset**($\_POST[**'name'**])) {  
 $errors = **array**();  
 **if** (*strlen*($this->**name**) == 0) *array\_push*($errors, **"Ім'я не введено"**);  
 **else if** (*strlen*($this->**surname**) == 0) *array\_push*($errors, **"Фамілію не введено"**);  
 **else if** (*strlen*($this->**age**) == 0) *array\_push*($errors, **"Вік не введено"**);  
 **else if** (*strlen*($this->**email**) == 0) *array\_push*($errors, **"Email не введено"**);  
 **return** $errors;  
 }  
 }  
  
 **public function** renderInfoAboutUser() {  
 **echo "Користувач:"**;**echo** $this->**name**;**echo " "**;**echo** $this->**surname**; **echo "<div></div>"**;  
 **echo "Вік: "**;**echo** $this->**age**;  
 **echo "<div></div>"**;  
 **echo "Email:"**;**echo** $this->**email**; **echo "</div>"**;  
  
 }  
  
}  
**?>**

**<?php**$result\_valid = **false**;  
$user = **new** User($\_POST[**'name'**],$\_POST[**'surname'**], $\_POST[**'age'**], $\_POST[**'email'**]);  
**if** (*count*($user->parametrValidation()) == 0) {  
 $result\_valid = **true**;  
}  
**?>**<**form action="task\_3\_3.php" method="post"**>  
 <**input type="text" placeholder="Введіть ім'я" name="name" value="<?php echo** $\_POST[**'name'**]; **?>"**>  
 <**input type="text" placeholder="Введіть фамілію" name="surname" value="<?php echo** $\_POST[**'surname'**]; **?>"**>  
 <**input type="text" placeholder="Введіть вік" name="age" value="<?php echo** $\_POST[**'age'**]; **?>"**>  
 <**input type="text" placeholder="Введіть email" name="email" value="<?php echo** $\_POST[**'email'**]; **?>"**>  
 <**button type="submit"**>Готово</**button**>  
</**form**>  
<**div class="result"**>  
  
 **<?php if** ($result\_valid) $user->renderInfoAboutUser(); **?>**</**div**>  
  
<**div class="error"**>  
 **<?php  
 for** ($i=0; $i < *count*($user->parametrValidation()); $i++) {  
 **echo** $user->parametrValidation()[$i];  
 **break**;  
 }  
 **?>**</**div**>

**Рисунок 3.1:** Виконання завдання мовою PHP+HTML



**Рисунок 3.1:** Результат роботи програми

**Завдання 4:**

Створити клас калькулятор (Calc). В клас включити методи, що реалізують всі арифметичні дії (додавання, віднімання, ділення, ділення по модулю, добування кореня, піднесення до степеня). У кожному методі передбачити перевірку допустимості аргументів. Якщо аргумент виходить за рамки допустимих значень - видати відповідне повідомлення.

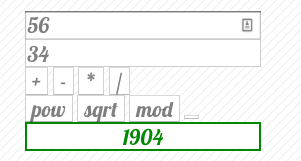
Перевірити працездатність класу створивши об'єкт і, викликавши кожний з його методів з довільними аргументами.

Створити клас для виводу і обробки форми розрахунку (CalcDispatcher). Розхрахунок повинен здійснюватися усередині методу display (). Обробка введення форми повинна відбуватися усередині методу dispatch (). Передбачити перевірку на введення тільки чисел. Зв'яжіть класи Calc і CalcDispatcher ставленням агрегації, таким чином, що б вся логіка калькулятора відбувалася за допомогою об'єкту класу Calc.

**<?php  
class** Calc {  
 **public $value1**;  
 **public $value2**;  
 **public $errors** = **array**();  
  
 **function** \_\_construct(){  
 $arguments = *func\_get\_args*();  
 $arguments\_amount = *func\_num\_args*();  
 **if** ($arguments\_amount == 1) {  
 $this -> **value1** = $arguments[0];  
 }  
 **else if** ($arguments\_amount == 2){  
 $this -> **value1** = $arguments[0];  
 $this -> **value2** = $arguments[1];  
 }  
 **else** {  
 *array\_push*($this->**errors**, **"Ви не ввели жодного значення"**);  
 }  
 }  
 **public function** plus () {  
 **return** $this->**value1** + $this->**value2**;  
 }  
 **public function** minus () {  
 **return** $this->**value1** - $this->**value2**;  
 }  
 **public function** division () {  
 **return** $this->**value1** / $this->**value2**;  
 }  
 **public function** product () {  
 **return** $this->**value1** \* $this->**value2**;  
 }  
 **public function** pow () {  
 **return** *pow*($this->**value1**, $this->**value2**);  
 }  
 **public function** sqrt () {  
 **if** ($this->**value1** >= 2) **return** *sqrt*($this->**value1**);  
 **else** *array\_push*($this->**errors**, **"Значення повинне бути не менше 2"**);  
 }  
 **public function** mod () {  
 **return** $this->**value1** % $this->**value2**;  
 }  
}  
**?>  
<?php  
class** CalcDispatcher {  
 **private $value1**;  
 **private $value2**;  
 **private $action**;  
 **public $errors** = **array**();  
  
 **public function** dispatch(){  
 $this -> **value1** = $\_GET[**'value1'**];  
 **if** (!$this->validOnlyNumbers($this->**value1**)) *array\_push*($this->**errors**, **"Значення 1 не коректне"**);  
 **if** (*strlen*($this->**value1**) == 0) *array\_push*($this->**errors**, **"Значення 1 не введене"**);  
 $this -> **action** = $\_GET[**'action'**];  
 **if** ($this->**action** != **'sqrt'**) {  
 $this->**value2** = $\_GET[**'value2'**];  
 **if** (!$this->validOnlyNumbers($this->**value2**)) *array\_push*($this->**errors**, **"Значення 2 не коректне"**);  
 **if** (*strlen*($this->**value2**) == 0) *array\_push*($this->**errors**, **"Значення 2 не введене"**);  
 }  
 }  
  
 **public function** display() {  
 $result = **null**;  
 **if** ($this->**action** == **"sqrt"**) {  
 $calc = **new** Calc($this->**value1**);  
 **if** ($this->**action** == **"sqrt"**) $result = $calc->sqrt();  
 $this->**errors** = $this->array\_concat($this->**errors**, $calc->**errors**);  
 **if** (*count*($this->**errors**) == 0) {  
 **return** $result;  
 } **else return** $this->**errors**;  
 } **else** {  
 $calc = **new** Calc($this->**value1**, $this->**value2**);  
 **if** ($this->**action** == **"+"**) $result = $calc->plus();  
 **if** ($this->**action** == **"-"**) $result = $calc->minus();  
 **if** ($this->**action** == **"\*"**) $result = $calc->product();  
 **if** ($this->**action** == **"/"**) $result = $calc->division();  
 **if** ($this->**action** == **"/"**) $result = $calc->division();  
 **if** ($this->**action** == **"pow"**) $result = $calc->pow();  
 **if** ($this->**action** == **"mod"**) $result = $calc->mod();  
  
 $this->**errors** = $this->array\_concat($this->**errors**, $calc->**errors**);  
 **if** (*count*($this->**errors**) == 0) {  
 **return** $result;  
 } **else return** $this->**errors**;  
 }  
  
 }  
  
 **private function** validOnlyNumbers($value) {  
 $dot\_amount = 0;  
 **for** ($i=0; $i < *strlen*($value); $i++) {  
 **if** (!*is\_numeric*($value[$i])) {  
 **if** ($value[$i] == **"."**) {  
 **if** ($dot\_amount != 0) **return false**;  
 **else** $dot\_amount++;  
 }  
 **else return false**;  
 }  
 }  
 **return true**;  
 }  
  
 **function** array\_concat ($arr1, $arr2) {  
 **for** ($i=0; $i < *count*($arr2); $i++) *array\_push*($arr1, $arr2[$i]);  
 **return** $arr1;  
 }  
}  
**?>  
  
<?php  
if**($\_SERVER[**'REQUEST\_METHOD'**] == **"GET"**) {  
 $is\_error = **false**;  
 $result = **null**;  
  
 $dispatch = **new** CalcDispatcher();  
 $dispatch->dispatch();  
 **if** (*gettype*($dispatch->display()) == **'array'**) { $is\_error = **true**; }  
 **else** {  
 $result = $dispatch->display();  
 }  
}  
  
**?>**

<**form action="task\_3\_4.php" method="get"**>  
 <**input type="text" placeholder="Введіть вичло 1" name="value1" value="<?php echo** $\_GET[**'value1'**]; **?>"**>  
 <**div**> </**div**>  
 <**input type="text" placeholder="Введіть вичло 2" name="value2" value="<?php echo** $\_GET[**'value2'**]; **?>" <?php if** ($\_GET[**'action'**] == **'sqrt'**) **echo "disabled"**; **?>**>  
 <**div class="calc-action-box"**>  
 <**button type="submit" name="action" value="+"**>+</**button**>  
 <**button type="submit" name="action" value="-"**>-</**button**>  
 <**button type="submit" name="action" value="\*"**>\*</**button**>  
 <**button type="submit" name="action" value="/"**>/</**button**>  
 </**div**>  
 <**div class="calc-action-box"**>  
 <**button type="submit" name="action" value="pow"**>pow</**button**>  
 <**button type="submit" name="action" value="sqrt"**>sqrt</**button**>  
 <**button type="submit" name="action" value="mod"**>mod</**button**>  
 <**button disabled**></**button**>  
 </**div**>  
  
 **<?php  
 if** ($is\_error) {  
 **echo "<input type='text' class='error-input' value='"**;  
 **echo** $dispatch->**errors**[0];  
 **echo "'>"**;  
 }  
 **?>  
 <?php  
 if** ($result) {  
 **echo "<input type='text' class='sucsses-input' value='"**;  
 **echo** $result;  
 **echo "'>"**;  
 }  
 **?>**

**Рисунок 4.1:** Виконання завдання мовою PHP+HTML



**Рисунок 4.2:** Результат роботи програми

**Завдання 5:**

Розробити гру «Хрестики нулики».

**<?php  
class** TicTacGame {  
 **private $fieldWidth** = 4;  
 **private $fieldHeight** = 4;  
  
 **private $countToWin** = 4;  
  
  
 **private $field** = **array**();  
  
  
 **private $winnerCells** = **array**();  
  
 **private $currentPlayer** = 1; *// 1 або 2, а після закінчення гри - null.* **private $winner** = **null**; *// після закінчення гри буде містити 1 або 2.* **public function** makeMove($x, $y) {  
*// Враховуємо хід, якщо виконуються всі умови:  
// 1) гра ще йде,  
//    2) клітина знаходиться в межах ігрового поля.  
//       3) в поле на зазначеному місці ще порожньо* **if**(  
 $this->**currentPlayer** &&  
 $x >= 0 && $x < $this->**fieldWidth** &&  
 $y >= 0 && $y < $this->**fieldHeight** &&  
 **empty**($this->**field**[$x][$y]))  
 {  
 $current = $this->**currentPlayer**;  
  
 $this->**field**[$x][$y] = $current;  
 $this->**currentPlayer** = ($current == 1) ? 2 : 1;  
  
 $this->checkWinner();  
 }  
 }  
  
 */\*\*  
 \* Поїск  
 \*/* **private function** checkWinner() {  
 **for**($y = 0; $y < $this->**fieldHeight**; $y++) {  
 **for**($x = 0; $x < $this->**fieldWidth**; $x++) {  
 $this->checkLine($x, $y, 1, 0);  
 $this->checkLine($x, $y, 1, 1);  
 $this->checkLine($x, $y, 0, 1);  
 $this->checkLine($x, $y, -1, 1);  
 }  
 }  
 **if**($this->**winner**) {  
 $this->**currentPlayer** = **null**;  
 }  
 }  
  
 */\*\*  
 \* Провірка чи знаходиться тут хтось  
 \*/* **private function** checkLine($startX, $startY, $dx, $dy) {  
 $x = $startX;  
 $y = $startY;  
 $field = $this->**field**;  
 $value = **isset**($field[$x][$y])? $field[$x][$y]: **null**;  
 $cells = **array**();  
 $foundWinner = **false**;  
 **if**($value) {  
 $cells[] = **array**($x, $y);  
 $foundWinner = **true**;  
 **for**($i=1; $i < $this->**countToWin**; $i++) {  
 $x += $dx;  
 $y += $dy;  
 $value2 = **isset**($field[$x][$y])? $field[$x][$y]: **null**;  
 **if**($value2 == $value) {  
 $cells[] = **array**($x, $y);  
 } **else** {  
 $foundWinner = **false**;  
 **break**;  
 }  
 }  
 }  
 **if**($foundWinner) {  
 **foreach**($cells **as** $cell) {  
 $this->**winnerCells**[$cell[0]][$cell[1]] = $value;  
 }  
 $this->**winner** = $value;  
 }  
 }  
  
 */\*  
 \* Функції нижче дозволяють отримати поточний стан гри і ігрового поля.)  
 \*/* **public function** getCurrentPlayer() { **return** $this->**currentPlayer**; }  
 **public function** getWinner() { **return** $this->**winner**; }  
 **public function** getField() { **return** $this->**field**; }  
 **public function** getWinnerCells() { **return** $this->**winnerCells**; }  
 **public function** getFieldWidth() { **return** $this->**fieldWidth**; }  
 **public function** getFieldHeight() { **return** $this->**fieldHeight**; }  
}  
**?>**

**<?php***session\_start*();  
  
*//Сохранення сесії (кукі)  
//Якщо ні то начати нову гру*$game = **isset**($\_SESSION[**'game'**])? $\_SESSION[**'game'**]: **null**;  
**if**(!$game || !*is\_object*($game)) {  
 $game = **new** TicTacGame();  
}  
  
*// Опрацювання запроса користувача*$params = $\_GET + $\_POST;  
**if**(**isset**($params[**'action'**])) {  
 $action = $params[**'action'**];  
  
 **if**($action == **'move'**) {  
 *// Опрацювання* $game->makeMove((int)$params[**'x'**], (int)$params[**'y'**]);  
  
 } **else if**($action == **'newGame'**) {  
 *// Користувач вирішив грати зпочатку* $game = **new** TicTacGame();  
 }  
}  
  
$\_SESSION[**'game'**] = $game;  
  
  
*// Вітображаєм сторінку HTML*$width = $game->getFieldWidth();  
$height = $game->getFieldHeight();  
$field = $game->getField();  
$winnerCells = $game->getWinnerCells();  
  
**?>**

**<?php if**($game->getCurrentPlayer()) { **?>** *<!-- Виводить зробити хід -->* Хід робить гравець  
 <**div class="icon player<?php echo** $game->getCurrentPlayer() **?>"**></**div**>...  
 **<?php** } **?>  
  
 <?php if**($game->getWinner()) { **?>** *<!-- Виводить хто виграв -->* Виграв гравець  
 <**div class="icon player<?php echo** $game->getWinner() **?>"**></**div**>!  
 **<?php** } **?>** *<!-- Малюємо ігарове поле, відображаючи зроблені ходи  
    і підсвічуючи перемогла комбінацію. -->* <**div class="ticTacField"**>  
 **<?php for**($y=0; $y < $height; $y++) { **?>** <**div class="ticTacRow"**>  
 **<?php for**($x=0; $x < $width; $x++) {  
 *// $player - игрок, сходивший в эту клетку :), или null, если клетка свободна.  
 // $winner - флаг, означающий, что эта клетка должна быть подсвечена при победе.* $player = **isset**($field[$x][$y])? $field[$x][$y]: **null**;  
 $winner = **isset**($winnerCells[$x][$y]);  
 $class = ($player? **' player'** . $player: **''**) . ($winner? **' winner'**: **''**);  
 **?>** <**div class="ticTacCell<?php echo** $class **?>"**>  
 **<?php if**(!$player) { **?>** *<!-- Клітка вільна. Відображаємо тут посилання,  
                            на яку потрібно клікнути для здійснення ходу. -->* <**a href="?action=move&amp;x=<?php echo** $x **?>&amp;y=<?php echo** $y **?>"**></**a**>  
 **<?php** } **?>** </**div**>  
 **<?php** } **?>** </**div**>  
 **<?php** } **?>** </**div**>  
  
 <**br**/><**a href="?action=newGame"**>Почати наву гру</**a**>

**Рисунок 5.1:** Виконання завдання мовою PHP+HTML



**Рисунок 5.2:** Результат роботи програми

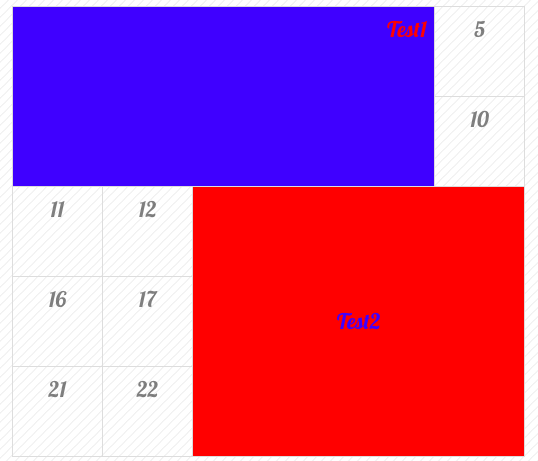
**Завдання 6:**

Необхідно реалізувати метод (функцію), яка згенерує HTML сторінку по заданих параметрах. На вхід подається двовимірний масив з параметрами.

**<?php**$arr = **array**(  
 **array**(  
 **'text'** => **"Test1"**,  
 **'cells'** => **'1,2,3,4,6,7,8,9'**,  
 **'align'** => **'right'**,  
 **'valign'** => **'top'**,  
 **'color'** => **'red'**,  
 **'bgcolor'** => **'blue'** ),  
 **array**(  
 **'text'** => **"Test2"**,  
 **'cells'** => **'13,14,15,18,19,20,23,24,25'**,  
 **'align'** => **'center'**,  
 **'valign'** => **'middle'**,  
 **'color'** => **'blue'**,  
 **'bgcolor'** => **'red'** ),  
);  
**function** getTable(**array** $arr){  
*// Створюю масив з номерів осередків які ввів користувач в массівв $ arr під  
// ключем cells і тут же його сортую в порядку зростання***for**($i = 0;$i < *count*($arr);$i++)  
{  
 $delimiter =**','**;  
 $arr\_cells[] = *explode*($delimiter, $arr[$i][**'cells'**]);  
 *sort*($arr\_cells[$i]);  
}  
*// КІНЕЦЬ Створення масиву з номерів, які ввів користувач*$size =5; *// Размер Таблицы = $size\*$size.  
// МОЖНА вибирати будь-який розмір матриці.***static** $k=1;  
*// Під кожен елемент створюю окремий масив*$colspan = **array**();  
$rowspan = **array**();  
$width = **array**();  
$height = **array**();  
$class = **array**();  
$color = **array**();  
$bgcolor = **array**();  
$text = **array**();  
*// Заповнюю тадліцу даними за замовчуванням***for**($i=1; $i <= $size\*$size; $i++)  
{  
 $width[] = **'100'**;  
 $height[] = **'100'**;  
 $colspan[] = **'1'**;  
 $rowspan[] = **'1'**;  
 $text[] = $i;  
}  
*// Основний код програми***for**($i=0;$i<*count*($arr\_cells); $i++)  
{  
 *// запам'ятовуючи для кожної групи осередків потрібні дані* $color[$arr\_cells[$i][0]-1] = $arr[$i][**'color'**];  
 $bgcolor[$arr\_cells[$i][0]-1] = $arr[$i][**'bgcolor'**];  
 $text[$arr\_cells[$i][0]-1] = $arr[$i][**'text'**];  
 $align[$arr\_cells[$i][0]-1] = $arr[$i][**'align'**];  
 $valign[$arr\_cells[$i][0]-1] = $arr[$i][**'valign'**];  
  
 $count =*count*($arr\_cells[$i]);  
 $row=1;  
 **for**($s= $count-2 ,$j=1;$j<$count,$s >= 0; $j++, $s--)  
 {  
 $class[$arr\_cells[$i][$j]-1] = **'hidden'**;*// для непотрібних мені осередків привласнюю клас  
 // Установлюю число осередків, які повинні бути об'єднані по вертикалі.* **if**(($arr\_cells[$i][$count -$j] - $arr\_cells[$i][$s]) != 1 )  
 {  
 $row++;  
 }  
 }  
 $col =$count/$row;*// Установлюю число осередків, які повинні бути об'єднані по горизонталі* $colspan[$arr\_cells[$i][0]-1] = $col;  
 $rowspan[$arr\_cells[$i][0]-1] = $row;  
 *// Перевірка того якщо номери осередків йдуть підряд* **for**($l=1;$l<$count;$l++)  
 {  
 **if**((*max*($arr\_cells[$i])-*min*($arr\_cells[$i])) == $l\*$size-1)  
 {  
 $rowspan[$arr\_cells[$i][0]-1] = $count/$size;  
 $colspan[$arr\_cells[$i][0]-1] = $size;  
 }  
 }  
 *// КІНЕЦЬ Перевірки того якщо номери осередків йдуть підряд*}  
*// КІНЕЦЬ Основного коду програми***?>**

<**div class=" container-fluid"**>  
  
 <**div class=" col-lg-<?php echo**$size**?>"**>  
 <**table class="table table-bordered"**>  
 **<?php for**($i=0;$i<$size;$i++)  
 {**?>** <**tr**>  
 **<?php for**($j=0;$j<$size;$j++,$k++){ **?>** <**td colspan="<?php echo** $colspan[$k-1]**?>"  
 rowspan="<?php echo** $rowspan[$k-1]**?>"  
 class="<?php echo** $class[$k-1]**?>"  
 style="  
 width**: **<?php echo** $colspan[$k-1]\*$width[$k-1].**'px'**;**?>**;  
 **height**: **<?php echo** $rowspan[$k-1]\*$height[$k-1].**'px'**;**?>**;  
 **background**: **<?php echo** $bgcolor[$k-1]**?>**;  
 **color**: **<?php echo** $color[$k-1]**?>**;  
 **text-align**: **<?php echo** $align[$k-1]**?>**;  
 **vertical-align**: **<?php echo** $valign[$k-1]**?>**;  
 **"**>  
 **<?php echo** $text[$k-1];**?>**</**td**>  
 **<?php** }**?>** </**tr**>  
 **<?php** }**?>** </**table**>  
 </**div**>  
  
  
 **<?php** }**?>** <! - КІНЕЦЬ тіла функції getTable ->  
  
 **<?php** getTable($arr); **?>** <! - Виклик функції getTable ($ arr) ->  
  
 *<!-- </div>-->*</**div**>

**Рисунок 6.1:** Виконання завдання мовою РHP+HTML



**Рисунок 6.1:** Результат роботи програми

**Висновок:** виконуючи данну лабораторну роботу я навчився створювати сценарії мовою програмування PHP з використанням ООП, засвоїв такі базові поняття як класи, екземпляри, поліморфізм, інкапсуляція, тощо...