



Національний університет  
водного господарства  
та природокористування

Міністерство освіти і науки України  
Національний університет водного господарства  
та природокористування

Кафедра обчислювальної техніки

**04-04-186**

**МЕТОДИЧНІ ВКАЗІВКИ**

до виконання лабораторних робіт  
з дисципліни “Web-технології та web-дизайн”  
студентами напряму підготовки  
6.050102 “Комп’ютерна інженерія”  
денної форми навчання  
Частина 2 РНР

Рекомендовано методичною  
космісією за напрямом  
підготовки “Комп’ютерна  
інженерія”  
Протокол № 3 від 6.02.2015р.

Рівне 2015



Методичні вказівки до виконання лабораторних робіт з дисципліни “Web-технології і Web-дизайн” студентами напряму підготовки 6.050102 “Комп’ютерна інженерія” денної форми навчання / Л.В.Зубик. - Рівне: НУВГП, 2015. - 24 с.

Упорядник: Л.В. Зубик, старший викладач кафедри обчислювальної техніки

Відповідальний за випуск: Б.Б.Круліковський, к.т.н., завідувач кафедри обчислювальної техніки

## ЗМІСТ

1. Передмова.....	3
2. Лабораторна робота № 1.....	4
3. Лабораторна робота № 2.....	6
4. Лабораторна робота № 3.....	9
5. Лабораторна робота № 4.....	11
6. Лабораторна робота № 5.....	13
7. Лабораторна робота № 6.....	14
8. Лабораторна робота № 7.....	16
9. Лабораторна робота № 8.....	19
10. Лабораторна робота № 9.....	21
11. Лабораторна робота № 10.....	22
Література	24

© Зубик Л.В., 2015

© НУВГП, 2015



## Передмова

Мета цих методичних вказівок – сформувати навички написання РНР-скриптів, які надалі можуть бути використані у процесі роботи із вмістом динамічних web-сайтів; навчити застосовувати можливості сучасного програмного забезпечення для створення власних лічильників відвідуваності сторінок сайту, розробки і аналізу роботи систем голосування, реалізації аутентифікації користувачів, ведення адресних книг, організації розсилок тощо.

У методичних вказівках вміщено завдання, реалізуючи які читач зможе опанувати базові прийоми підготовки сценаріїв, що виконуються у серверній частині додатків.

Методичні вказівки можуть бути використані у навчальному процесі для опанування дисципліни “Web-технології та Web-дизайн” студентами напряму підготовки 6.050102 “Комп’ютерна інженерія” денної форми навчання.

Вказівки містять кілька розділів, кожен з яких передбачає наявність: мети роботи; постановки задачі; посилань на джерела інформації, які необхідно опрацювати на етапі підготовки до виконання лабораторної роботи; прикладів розв’язування типових задач; завдань для виконання лабораторної роботи та самостійного опрацювання; переліку контрольних запитань за матеріалами розглянутої теми та отриманими результатами виконання поставлених задач.

**Звіт** до кожної виконаної лабораторної роботи виконується відповідно до вимог, що висуваються до оформлення документації у ВНЗ, і повинен містити:

- титульний аркуш;
- формулювання мети роботи;
- постановку задачі відповідно до варіанту завдання;
- результати роботи;
- висновки.



## Лабораторна робота № 1

### Тема: ОЗНАЙОМЛЕННЯ ІЗ СИНТАКСИСОМ PHP. ВИВЧЕННЯ РЕДАКТОРА PHP EXPERT EDITOR

**Мета:** Ознайомитися з основами синтаксису PHP. Опанувати прийоми роботи в редакторі PHP Expert Editor

#### Завдання:

1) Ознайомитися із середовищем підготовки PHP-скриптів та синтаксисом PHP. Для цього опрацювати джерела інформації:

- [1] стор. 395 - 401; 421 - 441;
- [2] стор. 39 - 54; 56 - 62.

2) Виконати всі пункти лабораторної роботи.

3) Надати звіт про виконану роботу.

#### Хід роботи

1. Створіть окрему сторінку, на якій будуть розміщуватися посилання на PHP-скрипти.
2. Реалізуйте наведений нижче приклад і з'ясуйте принцип його роботи:

```
<?PHP
```

```
$rows=5;
```

```
$columns=3;
```

```
echo '<html><body>';
```

```
echo '<table border="2">';
```

```
for ($i=1;$i<=$rows;$i++)
```

```
{ echo '<tr>';
```

```
  for ($j=1;$j<=$columns;$j++)
```

```
    { if (((($i+$j)%2)==0){ $c="green"; } else { $c="red"; }
```

```
    echo '<td bgcolor=".'$c.'><font color="yellow">'. $i, $j. '</font></td>';
```

```
    } /*end of for $j*/
```

```
  echo '</tr>';
```

```
} /*end of for $i*/
```



```
echo '</table>';  
echo '</body>  
</html>'; ?>
```

3. Отримайте інформацію про налаштування PHP за допомогою команди `PHPinfo ()`;
4. Підготуйте PHP-скрипт, який друкує сторінку "Виконуємо завдання лабораторної роботи № 1 !!!";
5. Створіть PHP-скрипт, який завантажує у web-сторінку файл з інформацією про розробника.
6. Реалізуйте PHP-скрипт, який виводить сторінку з таблицею кольорів HTML.
7. Розробіть PHP-скрипт, який формує web-сторінку з таблицею множення.

**Оформлення лабораторної роботи** надається у паперовому вигляді з описом порядку лабораторної роботи та наданням PHP-скриптів, створених під час лабораторної роботи.

### Контрольні запитання

1. Яке програмне забезпечення можна використовувати для роботи з PHP-скриптами?
2. Яким чином можна запустити Web-сервер?
3. Як Web-сервер виконує PHP-скрипт?
4. Яке призначення команди `PHPinfo ()`?
5. Які вимоги до імен змінних у PHP?
6. Назвіть відомі Вам PHP-константи.
7. Перелічіть типи даних PHP.
8. Поясніть прийоми перетворення типів даних у PHP.
9. Опишіть оператори виведення даних у PHP.
10. Яким чином можна організувати введення даних у PHP-скриптах?



## Лабораторна робота № 2

### Тема: УПРАВЛЯЮЧІ ОПЕРАТОРИ PHP, РОБОТА З ФОРМАМИ

**Мета:** Ознайомитися з синтаксисом операторів управління PHP, навчитися аналізувати дані з форм при розв'язуванні задач з підключенням PHP-скриптів

#### Завдання:

- 1) Ознайомитися з синтаксисом PHP-операторів if, case, for, while. Для цього опрацювати джерела інформації:
  - [1] стор. 510 - 524;
  - [2] стор. 62 - 68.
- 2) Виконати всі пункти лабораторної роботи.
- 3) Надати звіт про виконану роботу.

#### Хід роботи

1. У html-документі створіть форму з двома текстовими полями, в які користувач зможе вводити числа, та кнопкою **"Вивести дії"**. Після клацання на кнопці повинен виконатися PHP-скрипт, який буде демонструвати п'ять різних варіантів виконання арифметичної дії над вказаними раніше числами, наприклад так:  $5 + 2 = 7$ ;  $5 - 2 = 3$ ;  $5 * 2 = 10$ ;  $5 / 2 = 2.5$ ;  $5 \% 2 = 1$ .
2. У html-документі створіть форму з двома текстовими полями, в які користувач вводить два числа, та кнопкою **"Вивести більше число"**. Після клацання на кнопці повинен викликатися PHP-скрипт, який виведе більше з двох чисел або повідомлення, що два числа рівні.
3. У html-документі створіть форму з текстовим полем, в яке користувач вводить свій логін, та кнопкою **"Перевіримо..."**. Після клацання на кнопці повинен викликатися PHP-скрипт який буде перевіряти, чи цей користувач зареєстрований. При



цьому користувачів з різними логінами повинно бути чотири. Якщо введений один з існуючих логінів, то повинно виводитися привітання для відповідного користувача. Наприклад, при введенні логіну **“log1”**, повинно виводитися привітання **“Доброго дня, Рамський Петре Івановичу”**. У випадку введення невідомого логіну, передбачити виведення повідомлення: **“Вибачте, Ви у нас не зареєстровані!”**

4. Удосконалити програму п.3 таким чином: крім логіна користувач повинен вводити також пароль. Для зареєстрованого користувача пароль повинен перевірятися на коректність та відповідність вказаному користувачеві. Реалізувати завдання включенням блоку if в блоки case.
5. У html-документі створіть форму з текстовим полем, в яке користувач вводитиме число, та кнопкою **“Перевіримо число на парність:”**. Після клацання на кнопці повинен викликатися PHP-скрипт, який буде перевіряти число: парне воно чи ні, і виводити результат.
6. Завдання аналогічне наведеному у п. 5, але потрібно перевіряти, чи користувач ввів ціле число.
7. На сторінці з формою користувачеві пропонується відгадати число (ввести в текстове поле). Після натискання на кнопку **“Перевірка!”** повинен запускатися PHP-скрипт, який перевірятиме, чи відгадав користувач число, і якщо ні, то друкуватиме текст: **“Число N - не правильно”**. Далі буде надруковано текст **“Спробуйте ще раз!”**. Якщо користувач вгадає число, то великими червоними літерами повинно бути надруковано – **“ПРАВИЛЬНО!”**.
8. В процесі відгадування числа у п.7 передбачити виведення підказок типу – **“Загадане число більше”** або **“Загадане число менше”**.
9. Виведіть таблицю множення на введене користувачем число у вигляді, наведеному на рис. 1 (у прикладі користувач ввів число 3). Кожен рядок повинен бути іншого кольору.

$3 * 1 = 3$
$3 * 2 = 6$
...
$3 * 10 = 30$

Рис. 1



10. Користувач визначає довільний діапазон (наприклад, від 5 до 6). Необхідно вивести таблицю відношень числа 10 до всіх цілих чисел вказаного діапазону. Приклад показано на рис. 2.

$10 / 5 = 2$
$10 / 4 = 2.5$ т.д.

Рис. 2

11. У програмі з п. 10 після введення чисел перевіряти, яке число з двох введених є більшим і друк виконувати від меншого числа до більшого незалежно від порядку їх введення у формі. Зупинити цикл, якщо відбувається ділення на 0.
12. Скласти програму **“Вгадай число”**. Передбачити циклічний повтор процедури відгадування. Нехай користувач вводить довільне число на сторінці з формою і натискає кнопку **“Відгадати”**. При цьому повинен викликатися РНР-скрипт до тих пір, поки користувач не відгадає число. РНР-скрипт має передавати управління функції JavaScript, яка спочатку виводитиме **alert** з повідомленням **“Не відгадали”**, а потім – передаватиме управління формі. Цикл переривається, коли користувач вже відгадав число. У процесі вгадування передбачити виведення підказок: **“Загадане число більше”** або **“Загадане число менше”** аналогічно п. 8.
13. Використовуючи цикл з післяумовою, розв’яжіть наступну задачу. Відомо, що кількість бактерій подвоюється щогодини. Нехай користувач вводить такі дані: початкове число бактерій (наприклад, 10) і кінцеве (наприклад, 10 000). Потрібно у циклі виводити в різних рядках наступне:
1. минула 1 год. - маємо 20 бактерій
  2. минуло 2 год. - маємо 40 бактерій тощо.

При введенні перевіряти, чи початкове число менше кінцевого.

14. Створіть функцію, яка як аргумент отримує довільне число і потім виводить таблицю множення на це число у вигляді:  $3 * 1=3$ ,  $3 * 2 = 6$  і т.д. до  $3 * 10 = 30$ . Далі, викликаючи цю функцію в циклі, потрібно отримати таблицю множення для чисел від 2 до 9 (приклад на рис. 3).



$2 * 1 = 2$	$3 * 1 = 3$	$4 * 1 = 4$	$5 * 1 = 5$	$6 * 1 = 6$	$7 * 1 = 7$	$8 * 1 = 8$	$9 * 1 = 9$
$2 * 2 = 4$	$3 * 2 = 6$	$4 * 2 = 8$	$5 * 2 = 10$	$6 * 2 = 12$	$7 * 2 = 14$	$8 * 2 = 16$	$9 * 2 = 18$
$2 * 3 = 6$	$3 * 3 = 9$	$4 * 3 = 12$	$5 * 3 = 15$	$6 * 3 = 18$	$7 * 3 = 21$	$8 * 3 = 24$	$9 * 3 = 27$
$2 * 4 = 8$	$3 * 4 = 12$	$4 * 4 = 16$	$5 * 4 = 20$	$6 * 4 = 24$	$7 * 4 = 28$	$8 * 4 = 32$	$9 * 4 = 36$
$2 * 5 = 10$	$3 * 5 = 15$	$4 * 5 = 20$	$5 * 5 = 25$	$6 * 5 = 30$	$7 * 5 = 35$	$8 * 5 = 40$	$9 * 5 = 45$
$2 * 6 = 12$	$3 * 6 = 18$	$4 * 6 = 24$	$5 * 6 = 30$	$6 * 6 = 36$	$7 * 6 = 42$	$8 * 6 = 48$	$9 * 6 = 54$
$2 * 7 = 14$	$3 * 7 = 21$	$4 * 7 = 28$	$5 * 7 = 35$	$6 * 7 = 42$	$7 * 7 = 49$	$8 * 7 = 56$	$9 * 7 = 63$
$2 * 8 = 16$	$3 * 8 = 24$	$4 * 8 = 32$	$5 * 8 = 40$	$6 * 8 = 48$	$7 * 8 = 56$	$8 * 8 = 64$	$9 * 8 = 72$
$2 * 9 = 18$	$3 * 9 = 27$	$4 * 9 = 36$	$5 * 9 = 45$	$6 * 9 = 54$	$7 * 9 = 63$	$8 * 9 = 72$	$9 * 9 = 81$
$2 * 10 = 20$	$3 * 10 = 30$	$4 * 10 = 40$	$5 * 10 = 50$	$6 * 10 = 60$	$7 * 10 = 70$	$8 * 10 = 80$	$9 * 10 = 90$

Рис. 3

### Контрольні запитання

1. Запишіть умовний оператор if мови PHP.
2. Поясніть принцип роботи оператора switch мови PHP.
3. Наведіть приклади операторів циклу у PHP.
4. Поясніть принцип роботи оператора for мови PHP.
5. Розкажіть схему роботи оператора foreach у PHP.
6. Запишіть оператор while для PHP.
7. Розгляньте на прикладі послідовність виконання оператора do – while.
8. Як записати оператор циклу з передумовою у PHP?
9. Як записати оператор циклу з післяумовою у PHP?
10. Як можна перервати виконання циклу у PHP?
11. Поясніть механізм обміну даними між формою і PHP-скриптом.
12. Поясніть механізм обміну даними між формою і JavaScript-функцією.

### Лабораторна робота № 3

#### Тема: МАСИВИ У PHP

**Мета:** Ознайомитися з описом масивів у PHP та прийомами їх застосування при розв'язуванні задач

#### Завдання:

- 1) Ознайомитися з описом масивів та функціями опрацювання масивів у PHP. Для цього опрацювати джерела інформації:



- [1] стор. 444 - 459;
  - [2] стор. 71 -74.
- 2) Виконати всі пункти лабораторної роботи.
  - 3) Надати звіт про виконану роботу.

### Хід роботи

1. У масив внесіть 5 довільних чисел. Використовуючи конструкцію **foreach**, виведіть їх і їх квадрати у вигляді:  
 $4^2=16$   
 $2^2=4$   
 $5^2=25$  і т.д.
2. Створіть і заповніть асоційований масив, в якому іменами елементів будуть назви країн, а значеннями – кількість населення і назва столиці (дані придумайте самостійно). Виведіть з масиву таблицю 3 x 2 (три рядки по дві комірки в кожному). У лівій комірці - ім'я елемента, в правій - його значення. Змініть таблицю виведення на 2x3 (два рядки по три комірки в кожному). Виведіть в першому рядку - імена елементів, у другому – їх значення.
3. У HTML формі користувач вводить у чотири різні поля: прізвище, ім'я, вік і e-mail. Після натискання кнопки “**Готово**” запускається PHP-скрипт, який вносить ці дані в асоційований масив і далі виводить їх в таблицю, використовуючи конструкцію **foreach**. У формі передбачити перевірку, щоб кожне із полів перед відправленням не було порожнім.
4. Заповніть в циклах перший масив квадратами чисел від 10 до 20, а другий - кубами чисел від 1 до 10. Далі об'єднайте ці масиви і виведіть об'єднаний масив.



## Контрольні запитання

1. Наведіть приклад одного з можливих варіантів заповнення масиву значеннями у PHP.
2. Наведіть приклад одного з можливих варіантів заповнення асоційованого масиву значеннями у PHP.
3. Поясніть схему роботи оператора `foreach` у PHP.
4. Поясніть механізм передачі значень за посиланнями.
5. Наведіть приклади функцій, які можуть бути використані для опрацювання масивів у PHP.

## Лабораторна робота № 4

### Тема: ФАЙЛИ І КАТАЛОГИ

**Мета:** Ознайомитися з методами опрацювання даних файлів і каталогів у PHP-скриптах

#### Завдання:

- 1) Опрацювати інформацію, яка стосується роботи з файлами і каталогами у PHP-скриптах. Для цього опрацювати джерела інформації:
  - [1] стор. 549 - 572.
- 2) Виконати всі пункти лабораторної роботи.
- 3) Надати звіт про виконану роботу.

#### Хід роботи

1. Використовуючи PHP-скрипт і форму в рамках одного документа створіть сценарій, в якому користувач буде вводити у текстове поле ім'я файла і після натискання кнопки **“Готово”** виконається перевірка, чи існує такий файл у поточному каталозі. Якщо він не існує, виведеться повідомлення: **“Файл з іменем *Zrazok.txt* у поточному каталозі не існує”**. Якщо ж файл існує, виведеться інше повідомлення, крім того дані про розмір, час створення, час останньої модифікації і вміст файла.



- У текстовому файлі в першому рядку вписати тег (без дужок <>) у другому - його опис, в третьому - інший тег, у четвертому – його опис і так далі, усього 5 - 6 тегів. Далі в PHP-скрипті організувати зчитування даних файла по одному рядку і виведення його вмісту у вигляді, поданому на рис. 4:

 	розриває рядок
<td>	відкриває комірку в таблиці

Рис. 4. Таблиця з результатом роботи скрипта до п. 2

- Удосконалити задачу з п. 2 таким чином: скрипт повинен порахувати, скільки всього тегів описано у файлі. Вивести відповідь нижче таблиці у вигляді наступного тексту: **“Всього у файлі описано тегів: 5”**.
- Реалізуйте наступний сценарій: Користувач заходить на сторінку, у двох окремих текстових полях вводить свої логін і пароль, із списку вибирає один з режимів: **“Рєєстрація”** або **“Вхід”**. У разі вибору **“Вхід”** перевіряється, чи існує файл з іменем, що співпадає з логіном, і, якщо так, перевіряється пароль, чи відповідає він вмісту файла. Далі виводиться одне з повідомлень: **“Доброго дня, <логін>!”** або **“Пароль не правильний”**. Якщо файлу з іменем користувача, що входить як зареєстрований, не виявлено - вивести повідомлення: **“Такий користувач не зареєстрований!”**. Якщо користувач входить в режимі **“Рєєстрація”**, перевірити, чи не зайнятий логін і, якщо він вільний - створити файл з іменем логін, зберегти пароль в цьому файлі. Якщо ж логін зайнятий, - запропонувати змінити його.

### Контрольні запитання

- Перелічіть функції, які використовують у PHP-скриптах для роботи з файлами.
- Поясніть призначення `file_exists()`.
- Наведіть приклади варіантів використання `fopen()`, `fclose()`.
- Яке призначення функцій `fread()`, `fwrite()`?
- Наведіть приклад використання `feof()`.



6. Що отримаємо в результаті застосування `filesize()`?
7. Яка функція дозволяє вивести властивості файлу?
8. Як дізнатися дату останньої модифікації файла?

## Лабораторна робота № 5

### Тема: РЕГУЛЯРНІ ВИРАЗИ

**Мета:** Навчитися будувати і застосовувати регулярні вирази

#### Завдання:

- 1) Вивчити правила формування регулярних виразів. Для цього опрацювати джерела інформації:
  - [1] стор. 470-487.
- 2) Виконати всі пункти лабораторної роботи.
- 3) Надати звіт про виконану роботу.

#### Хід роботи

1. Перевірте довільний рядок, чи він є правильною e-mail адресою. Використайте функцію `preg_match()`.
2. Напишіть функцію, яка із введеного тексту буде виводити тільки назви тегів (назвою тегу є текст, що знаходиться всередині `<>`, наприклад: `<div>`).
3. Перевірте телефонні номери на відповідність шаблону `+38 (097) 65-94-888`.
4. Є масив дат у різних форматах (`rrrr-мм-дд`, `дд-мм-rrrr`). Приведіть їх до формату `мм.дд.rrrr` за допомогою функції `preg_replace()`.

#### Контрольні запитання

1. Назвіть відомі Вам формати регулярних виразів.
2. Поясніть використання функцій `preg_grep()`, `preg_match()`, `preg_replace()`.
3. Який пошук здійснюватиме вираз:
  - `“/^\w{6,9}\d {3}/s”`;



- “/\d{3}-\d{2}-\d{2}/m”?

4. Що вказують за допомогою квантифікаторів?
5. Що вказують в параметрі модифікатор?

### Лабораторна робота № 6

## Тема: РОБОТА З XML-ДОКУМЕНТАМИ. РОЗШИРЕННЯ DOM і SimpleXML

**Мета:** Здобути навички опрацювання XML-документів

### Завдання:

- 1) Навчитися опрацьовувати XML-дані засобами PHP. Для цього опрацювати джерела інформації:

- [7]

- 2) Виконати всі пункти лабораторної роботи.
- 3) Надати звіт про виконану роботу.

### Хід роботи

Нехай дано файл *example.PHP* з кодом:

```
<?PHP
$xmlstr = <<<XML
<?xml version='1.0' standalone='yes'?>
<movies>
<movie>
<title> PHP: Behind the Parser </title>
<characters>
<character>
<name> Ms. Coder </name>
<actor> Olivia Actora </actor>
</character>
<character>
<name> Mr. Coder </name>
<actor> El Act&#211;r </actor>
```



```
</character>
```

```
</characters>
```

```
<plot>
```

So, this language. It's like, a programming language. Or is it a scripting language? All is revealed in this thrilling horror spoof of a documentary.

```
</plot>
```

```
<rating type="thumbs"> 7 </rating>
```

```
<rating type="stars"> 5 </rating>
```

```
</movie>
```

```
</movies>
```

```
XML;
```

```
?>
```

Для наведеного прикладу потрібно:

1. Вивести значення яке зберігається у вузлі `<plot>`.
2. Змінити значення вузла `<name>` з “Ms. Coder” на “Missis Coder” і вивести на екран.
3. Вивести всі атрибути xml-документа.
4. Додати новий атрибут до вузла, значення якого дорівнює “5”.

### Контрольні запитання

1. Поясніть призначення xml.
2. Яке програмне забезпечення використовують для роботи з xml-документами?
3. Як вказати кодування документа і версію мови xml?
4. Які кодування забезпечують відображення кирилиці?
5. Скільки кореневих елементів повинно бути в xml- документі?
6. Чи може початковий тег починатися з цифри?
7. З якого символу може починатися кінцевий тег?
8. У чому відмінність початкового і кінцевого тегів елемента?



## Лабораторна робота № 7

### Тема: ЗНАЙОМСТВО З PHPMyAdmin, РОБОТА З СЕРВЕРОМ БАЗИ ДАНИХ (БД) MySQL. SQL-ЗАПИТИ

**Мета:** Набути навички роботи з PHPMyAdmin, навчитися виконувати адаптацію реєстраційної форми для роботи з БД.  
Сформулювати навички роботи із запитами

#### Завдання:

- 1) Ознайомитися з роботою в PHPMyAdmin.
- 2) Опанувати прийоми роботи з БД MySQL.
- 3) Навчитися реалізувати SQL-запити.

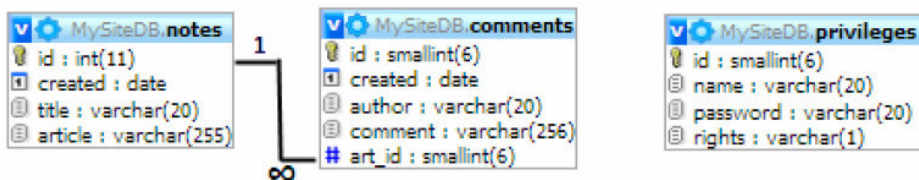
Для цього опрацювати джерела інформації:

- [1] стор. 379 – 395; 661 - 774;
- [4] стор. 286 – 313; 410 - 412.

- 4) Виконати всі пункти лабораторної роботи.
- 5) Надати звіт про виконану роботу.

#### Хід роботи

1. Створіть БД зі своїм ім'ям.
2. Реалізуйте таблицю користувачів **Kor** з наступними полями: логін, пароль, ім'я, дата народження, стать, країна, електронна пошта, примітка.
3. Підготуйте PHP-сценарій обробки форми реєстрації і запис даних про користувача в таблицю користувачів.
4. Розробіть скрипт, який виведе список всіх користувачів, які є в базі.
5. Створіть базу даних **MySiteDB** згідно наступного завдання:



Структура таблиці БД *Notes* наступна:

Ім'я поля	Тип даних	Опис поля	Примітка
Id	Int(11)	Ідентифікатор запису	Ключ
Created	Date	Дата створення	
Title	VarChar(50)	Заголовок	
Article	VarChar(255)	Вміст	

Структура таблиці БД *Comments* наступна:

Ім'я поля	Тип даних	Опис поля	Примітка
Id	Int(11)	Ідентифікатор запису	Ключ
Created	Date	Дата публікації замітки	
Author	VarChar(50)	Автор коментаря	
Comment	VarChar(255)	Вміст коментаря	
Art_Id	Int(11)	Ідентифікатор	Зовнішній ключ для з'єднання коментарів із замітками



Структура таблиці БД *Privileges* наступна:

Ім'я поля	Тип даних	Опис поля	Примітка
Id	Int(11)	Ідентифікатор запису	Ключ
Name	VarChar(20)	Ім'я користувача	
Password	VarChar(20)	Пароль користувача	
Rights	VarChar(1)	Права доступу користувача до сторінок сайту	

6. Встановіть зв'язок між таблицями *notes* і *comments* , використайте для цього поля *id* і *art\_id*.
7. Підготуйте кілька довільних SQL-запитів.

### Контрольні запитання

1. Що таке реляційна БД?
2. Які дії можна виконувати з БД?
3. Який синтаксис має запит SELECT?
4. Яка PHP-функція використовується для підключення до сервера БД?
5. Які етапи роботи PHP-скрипта необхідні для роботи з БД?
6. Який результат роботи поверне функція `mysql_query ()`?



## Лабораторна робота № 8

### Тема: ОСНОВИ ОБ'ЄКТНО ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ (ООП)

**Мета:** Ознайомитися з основами ООП в РНР

#### **Завдання:**

- 1) Ознайомитися з класами, методами, полями. Для цього опрацювати джерела інформації:
  - [1] стор. 627-639.
- 2) Розібратися з парадигмою ООП.
- 2) Виконати всі пункти лабораторної роботи.
- 3) Надати звіт про виконану роботу.

#### **Хід роботи**

1. Створіть клас **“Таблиця множення для вказаного числа”** (передавати в конструкторі). Реалізуйте окремий метод для обчислень. Далі створіть кілька об'єктів даного класу для демонстрації працездатності класу. Висновок можна оформити у вигляді таблиці.
2. Створіть клас **“Країни”**, в якому будуть поля: назва країни, столиця і кількість населення у столиці (дані підберіть довільні). Створіть масив об'єктів, виведіть кожен з них у таблицю в три рядки, по дві комірки в кожному. У лівій комірці - ім'я елемента, у правій - його значення.
3. Створіть клас **“Користувач”** з полями: прізвище, ім'я, вік і e-mail.
4. Нехай користувач вводить у HTML-формі дані в такі поля: прізвище, ім'я, вік і e-mail. Після натискання клавіші **“Готово”** повинен утворитися об'єкт користувача з методом,



який буде заносити ці дані в поле об'єкта і далі виводити їх, використовуючи інший метод класу користувача. У формі передбачити перевірку, що всі поля перед відправленням не порожні.

5. Створіть клас **“Калькулятор”**. В клас включіть методи, що реалізують всі арифметичні дії (додавання, віднімання, ділення, ділення по модулю, добування кореня, піднесення до степеня). У кожному методі передбачте перевірку допустимості аргументів. Якщо аргумент виходитиме за рамки допустимих значень – повинно бути виведене відповідне повідомлення.
6. Перевірте працездатність класу, створивши об'єкт і, викликавши кожен з його методів з довільними аргументами.
7. Створіть клас **“Диспетчер”** для виводу і обробки форми розрахунку. Висновок повинен здійснюватися усередині методу **display()**. Обробка введення форми повинна відбуватися усередині методу **dispatch()**. Передбачте перевірку на введення тільки чисел. Зв'яжіть класи **“Калькулятор”** і **“Диспетчер”** встановленням агрегації таким чином, що б вся логіка калькулятора відбувалася за допомогою об'єкта класу **“Калькулятор”**.
8. Намалюйте діаграму класів Вашої програми.

### Контрольні запитання

1. Дайте визначення класу.
2. Розкажіть про методи.
3. Поясніть призначення поля.
4. опишіть парадигму об'єктно-орієнтованого програмування.



## Лабораторна робота № 9

### Тема: КОМПЛЕКСНЕ ЗАСТОСУВАННЯ ООП

**Мета:** Навчитися застосовувати ООП під час розробки великих за обсягом додатків

#### Завдання:

- 1) Реалізувати гру “Хрестики - нулики”.
- 2) Надати звіт про виконану роботу.

#### Хід роботи

1. Створити клас для виводу на екран поля гри (таблиця 3x3 з кнопками. На кожній кнопці - відповідний символ).
2. Створити клас, що обробляє логіку гри (зберігання інформації про стан у сесії, хід гравця, рішення про результат партії).
3. Створити клас, який забезпечує вхід гравця (форма, аутентифікація користувача).
4. Гра відбувається між двома людьми.

#### Сценарій гри:

##### Основний (вдалий) сценарій.

- 1.1. Гравець запускає гру.
- 1.2. На екран виводиться поле з порожніми клітинами.
- 1.3. Програма пропонує натиснути на клітинку.
- 1.4. Гравець натискає на бажану йому клітинку.
- 1.5. Програма виводить поле з символами: хрестиками та нулями.
- 1.6. Програма оцінює розташування символів на виграш, програш, нічию. Якщо відбулася одна з цих подій, то видається відповідне повідомлення.
  - 1.6.1. Гравцеві пропонується почати гру заново.
  - 1.6.2. Якщо гравець погоджується, то переходимо до пункту 1.2.



1.6.3. Інакше програма завершується.

### **Гілки невдалих сценаріїв:**

- 2.1. (Варіант 1.4) Гравець натискає на клітинку, яка вже зайнята.
- 2.2. Програма видає повідомлення, що клітина вже зайнята.
- 2.3. Перехід до пункту 1.3 основного сценарію.

## **Лабораторна робота № 10**

### **Тема: ПАТТЕРНИ ПРОЕКТУВАННЯ**

**Мета:** Сформувані уявлення про сферу застосування паттернів проектування

#### **Завдання:**

- 1) Ознайомитися з класифікацією паттернів та рекомендаціями щодо їх застосування. Для цього опрацювати джерела інформації: [6], стор. 93 – 102; 130 - 139; 141 – 152; 173 – 183.
- 2) Виконати усі пункти лабораторної роботи.
- 3) Надати звіт про виконану роботу.

#### **Хід роботи**

1. Опишіть клас, який реалізує паттерн *Singleton*.

**Singleton (Сінглтон).** Паттерн є “хрестоматійним” і одним з найлегших. Він відноситься до так званих *породжуючих* паттернів, які дозволяють абстрагувати процес ініціювання. Він дозволяє створити деякий абстрактний інтерфейс доступу до примірників деяких об'єктів. Призначення паттерна *Сінглтон* - забезпечити наявність у класу тільки одного примірника-об'єкта на всю систему і надати до нього глобальний доступ.

2. Опишіть класи, які реалізують паттерн *Decorator*.



**Decorator (Декоратор).** Паттерн відноситься до класу *структурних*. Він використовується для динамічного розширення функціональності об'єкта. Є гнучкою альтернативою спадкоємництва. Сутність роботи паттерну **Декоратор** полягає в “огортанні” готового об'єкта новим функціоналом, при цьому весь оригінальний інтерфейс об'єкта залишається доступним. **Декоратор** переадресовує всі запити об'єкту. Сенс у тому, щоб можна було безболісно комбінувати різні **Декоратори** у довільному порядку, призначаючи їх різним об'єктам. У деякому сенсі, це схоже на технологію *traits*, за винятком того, що **Декоратори** динамічно призначаються об'єктам, а *traits* статично класам.

3. Опишіть класи, які реалізує паттерн **Adapter**.

**Adapter (Адаптер).** Паттерн відноситься до класу *структурних*. Він використовується для перетворення одного інтерфейсу в інший, необхідний клієнту. **Адаптер** забезпечує сумісність несумісних інтерфейсів, реалізуючи прошарок. **Адаптер** успадковує відкритим способом цільовий інтерфейс **Target**, і закритим способом - інтерфейс, що адаптується **Adaptee**. У реалізації методів цільового інтерфейсу відбувається перенаправлення (делегування) запитів класу з адаптованим інтерфейсом.

4. Опишіть клас який, реалізує паттерн **Abstract Factory**.

**Abstract Factory.** Паттерн має відношення до класу *породжуючих*. Його основне призначення - надати інтерфейс для створення сімейства взаємопов'язаних об'єктів, не специфікуючи їх класи.

### Контрольні запитання

1. Поясніть принцип паттернового проектування.
2. Подайте класифікацію паттернів.



3. Розкажіть про підходи до вибору паттернів для розв'язування конкретних задач.

## ЛІТЕРАТУРА

1. Прохоренок Н.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. 3-е изд. – СПб.: БХВ-Петербург, 2010 – 912 с. ISBN 978-5-9775-0540-6
2. Кухарчик А. PHP: обучение на примерах. – Мн.: Новое знание, 2004. – 237 с. ISBN 985-475-050-7
3. Клименко Р.А. Веб-мастегинг на 100%. – СПб.: Питер, 2013. – 512 с.: ил. ISBN 978-5-496-00079-6
4. Ташков П. Веб-мастеринг на 100%: HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка. – СПб: Питер, 2010. – 512 с.: ил. ISBN 978-5-49807-829-7
5. Росс В.С. Создание сайтов: HTML, CSS, PHP, MySQL. Учебное пособие, ч.1 – МГДД(Ю)Т, М.: 2010. – 107 с.
6. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб: Питер, 2001. – 368 с. : ил. ISBN 5-272-00355-1
7. <http://PHP.net/manual/ru/index.PHP>
8. <http://www.on-line-teaching.com/PHP/>
9. <http://PHPclub.ru>
10. <http://www.PHP5.ru/>
11. [http://myblaze.ru/category/PHP\\_lessons/](http://myblaze.ru/category/PHP_lessons/)
12. <http://www.puzzleweb.ru/PHP/>