

Міністерство освіти і науки України
Державний заклад «Південноукраїнський національний
педагогічний університет імені К.Д. Ушинського»
Фізико-математичний факультет

Шувалова Ольга Ігорівна

WEB-ПРОГРАМУВАННЯ СТОРОНИ СЕРВЕРУ

методичні рекомендації

Одеса 2019

УДК: 378.013+004.9

Друкується за рішенням Вченої ради Державного закладу
«Південноукраїнський національний педагогічний університет
імені К.Д. Ушинського» (протокол №11 від 27 червня 2019 року).

Шувалова О.І.

Web-програмування сторони серверу. Методичні рекомендації до лабораторних робіт з теми «Web-програмування». - Одеса: Університет Ушинського, 2019. - 95с.

Рецензенти:

Максимов Максим Віталійович, доктор технічних наук, професор, завідувач кафедри Комп'ютерних технологій автоматизації Одеського національного політехнічного університету.

Брескіна Лада Валентинівна, кандидат педагогічних наук, доцент кафедри Прикладної математики та інформатики Південноукраїнського національного педагогічного університету імені К. Д. Ушинського.

Методичні рекомендації «**Web-програмування сторони серверу**» – це навчальне видання з методики засвоєння лабораторно-практичної частини другого змістового модулю «Web-програмування» навчальної дисципліни ВС 3.4 «Комп'ютерні мережі» для студентів 3 курсу спеціальності 014 Середня освіта (Фізика) або 014 Середня освіта (Математика) з другою спеціальністю 014 Середня освіта (Інформатика). Поданий у методичних рекомендаціях матеріал охоплює питання, що формують професійні базові підходи до структурування Web-орієнтованої інформаційної системи на основі мови серверного програмування PHP, шаблонизатору Twig та мови SQL запитів при роботі з СУБД MySQL.

Реалізація навчальних прикладів до лабораторно-практичних робіт відповідає архітектурі MVC (Model-View-Controller), що є основою для розуміння принципів розмежування частини формування дизайну інформаційної системи, проектування бази даних та розроблення програмної частини, частини контролеру. Цей підхід наближає до розуміння принципів побудови і функціонування сучасних Web-орієнтованих інформаційних систем, що розроблено на базі CMS (Content Management System) таких, як CMS Joomla.

Ключові слова: Web-програмування, MVC архітектура, шаблонизатор Twig, мова PHP, методика інформатики, інформаційна система, бази даних.

Зміст

Вступ	4
1. Лабораторна робота №1. Мова HTML. CSS стилі. Мова JavaScript. Повторення.	5
2. Лабораторна робота №2. Мова php. Налаштування інтерфейсу виведення результатів. MVC-архітектура (Model-View-Controller) Web-орієнтованої системи. Розгортання системи “MVC-старт”.	13
3. Лабораторна робота №3. Мова PHP. Побудова лінійних алгоритмів. Передавання даних з частини контролера в частину дизайну.	25
4. Лабораторна робота №4. Алгоритми розгалуження в мові php.	33
5. Лабораторна робота №5. Алгоритми розгалуження, що реалізуються з використанням оператора вибору в мові php. Умовний оператор в шаблонизаторі Twig.	41
6. Лабораторна робота №6. Циклічні алгоритми в мові php.	46
7. Лабораторна робота №7. Масиви в мові php. Спеціальні цикли елементами масиву в мові php та в шаблонизаторі Twig.	52
8. Лабораторна робота №8. Функції. Створення графічного інтерфейсу для введення даних. Методи опрацювання форм в мові PHP. Стани файлів частини контролера.	60
9. Лабораторна робота №9. Об’єктно-орієнтовані основи мови php.	66
10. Лабораторна робота №10. Клас PDO в мові PHP. Налаштування роботи з базою даних. Табличний шаблон сторінок інформаційної системи.	71
11. Лабораторна робота №11. Опрацювання більш складних запитів до бази даних. Повторення мови SQL. Шаблон - блог для сторінок інформаційної системи.	86
Список використаних джерел	95

Вступ

Навчальна дисципліна ВС 3.4 «Комп'ютерні мережі» для студентів 3 курсу спеціальності 014 Середня освіта (Фізика), 014 Середня освіта (Математика) з другою спеціальністю 014 Середня освіта (Інформатика) за навчальним планом підготовки фахівця освітнього рівня «бакалавр» охоплює 5 кредитів загальною кількістю 150 годин. Аудиторні заняття - 80 годин (лекції — 16 годин, лабораторно-практичні роботи — 64 години), 70 годин відведено на самостійну роботу студентів. Зміст навчальної дисципліни розділено за програмою на два змістових модуля, перший з яких охоплює загальну теорію комп'ютерних мереж, а другий присвячено питанням Web-програмування.

Методичні рекомендації «Web-програмування сторони серверу» розроблено відповідно до програми лабораторно-практичних робіт другого змістового модуля навчальної дисципліни ВС 3.4 «Комп'ютерні мережі». В методичних рекомендаціях представлено 11 лабораторно-практичних робіт, що за програмою навчальної дисципліни розраховано на 36 годин аудиторної роботи, одне підсумкове заняття (2 години) по звітуванню результатів проектної роботи та 40 годин самостійної роботи студентів.

Зміст лабораторно-практичних робіт спирається на базові знання студентів, що отримано в межах навчальної дисципліни «Інформатика» на першому та другому курсах навчання. Перша лабораторна робота — це повторення мови HTML та CSS, що було опановано за програмою навчальної дисципліни «Інформатика». У лабораторних роботах 3, 4, 5, 6, 8 проводиться паралель з навчальною дисципліною «Програмування», що вивчається студентами на другому курсі. У лабораторних роботах 9, 10, 11 виконується робота з базами даних, що оглядово вивчаються за програмою навчальної дисципліни «Інформатика» та використовуються на лабораторних заняттях при вивченні теми «CMS Joomla».

Розроблений цикл лабораторно-практичних робіт є основою для подальшого вивчення тематичного розділу «Проектування та програмна реалізації Web-орієнтованої інформаційної системи», що планується в програмі навчальної дисципліни «Бази даних» на четвертому курсі навчання студентів спеціальності 014 Середня освіта (Фізика), 014 Середня освіта (Математика) з другою спеціальністю 014 Середня освіта (Інформатика) в відповідності до навчального плану.

Лабораторна робота №1.

Мова HTML. CSS стилі. Мова JavaScript. Повторення.

Ціль: Повторення мови HTML та CSS. Визначення сутності об'єктно-орієнтованого підходу до вивчення гіпертекстового документу.

Зміст роботи

1. Термінологія об'єктно-орієнтованого підходу до вивчення мови HTML
2. Ієрархія об'єктів браузера та правила однозначної ідентифікації об'єкта document і вкладених об'єктів гіпертекстового документу у мовах HTML, CSS, JavaScript
3. Базова структура гіпертекстового документа.
4. Таблиця базових об'єктів мови HTML, що відображаються в робочому вікні браузера.
5. Групи CSS стилів.
6. Приклад виконання завдання.
7. Індивідуальні завдання.

1. Термінологія об'єктно-орієнтованого підходу до вивчення мови HTML

В основі об'єктно-орієнтованої мови програмування лежать два основних поняття: **об'єкт** (клас) та **екземпляр об'єкта** (об'єкт). Екземпляр об'єкта – це базове поняття в ООП, це конкретна реалізація.

Основні властивості екземпляр об'єктаів:

Інкапсуляція це здатність екземпляр об'єктаів скривати деякі способи обробки даних (методи) та самі дані від навколишнього цифрового середовища.

Поліморфізм - це здатність екземпляр об'єкта використовувати методи батьківського класу. Слово «поліморфізм» можна перекласти як «багато форм». Тобто це можливість використання одного і того ж імені методу до екземплярів об'єктів різних об'єктів (класів).

Спадкування - стосується здатності мови дозволяти будувати нові об'єкти (класи) на основі значень існуючих об'єктів (класів). По суті, спадкування дозволяє розширювати поведінку базового об'єкту (класу), наслідуючи основну функціональність батьківського об'єкта (класа) у похідному об'єкті (класі).

2. Ієрархія об'єктів браузера та правила однозначної ідентифікації об'єкта document і вкладених об'єктів гіпертекстового документу у мовах HTML, CSS, JavaScript (Рис.1)

window

- frames
- history
- location
- navigator

- **document**

- links
- images
- ...
- forms
- elements

→ **HTML** <p id="id_identified1">

→ **JavaScript**
document.getElementById('id_identified1')

→ **CSS** #id_identified1{...}

Рис. 1. Ієрархія об'єктів браузера та однозначна їх ідентифікація в мовах HTML, CSS, JavaScript

3. Базова структура гіпертекстового документа (Рис. 2)

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
<title>  
Гіпертекстовий документ  
</title>
```

```
<meta charset="UTF-8">
```

```
<link rel="stylesheet"  
href="style1/style.css">
```

```
<script src="script/script.js">  
</script>
```

```
</head>
```

```
<body>
```

```
...
```

```
</body>
```

```
</html>
```

ГОЛОВА

(Керуючі
конструкції
HTML
документа)

ТІЛО

(Конструкції,
що
відображають
ся в робочому
вікні
браузера)

1. Визначення заголовку HTML документа
2. Визначення кодування HTML документа
3. Підключення файлу з CSS стилями
4. Підключення файлу з JavaScript програмою

Рис. 2. Базова структура гіпертекстового документа

4. Таблиця базових об'єктів мови HTML, що відображаються в робочому вікні браузера (Таблиця1)

Таблиця1. Об'єкти HTML документу

Назва об'єкта	Об'єкт	Екземпляр об'єкта
документ	<body></body>	<body id="bdd1">...</body>
блок		<div id="block1">...</div>
заголовок	 ... 	<h1 id="z1">...</h1> <h2 id="z2">...</h2> ... <h6 id="z6">...</h6>
абзац		<p id="pp1">...</p>
гіперпосилання		...
рисунок		
список		<ol id="sp1">... <ul id="sp2">...
таблиця	<table><tr><td>...</td></tr>...</table>	<table id="tb1"><tr id="tr1"><td id="td1">...</td></tr>...</table>
форма	<form method="get/post" action="URL">...</form>	<form method="get/post" action="URL" id="fm1">...</form>
елементи форми	1.<input type="" name="" ...> 2.<select name="">...</select> 3.<textarea name="">...</textarea>	1.<input type="" name="" id="el1"> 2.<select name="" id="el2">...</select> 3.<textarea name="" id="el3">...</textarea>

5. Групи CSS стилів (Таблиця 2)

Таблиця 2. Групи стилів CSS

Стилі блоку			Одиниці вимірювання
width	ширина блоку	#b11{width:200px;}	px, %
height	висота блоку	#b11{height:200px;}	px, %
margin	зовнішній відступ блоку	#b11{margin:2px;}	px, %
padding	внутрішній відступ блоку	#b11{padding:0px;}	px, %
border	бордюр по периметру блоку	#b11{border:2px solid blue;}	2px-товщина в пікселях solid - стиль (суцільний) blue-колір - #00f - rgb(0,0,255))
float	обтікання блоку	#b11{float:left;}	left - притиснути блок до лівого краю right - до правого
clear	відміна обтікання	#b11{clear:both;}	both-всіх left-лівого right - правого
Стилі кольору і фону			
color	колір тексту	#b11{color:red;}	
background	налаштування фону	#b11{background: url(images/hand.png) repeat-y #fc0;}	url(images/hand.png) шлях до фонового зображення repeat-y вертикальне повторення фонового зображення #fc0 колір фону
Стилі тексту			

font	налаштування шрифтів	#b1{font: 12pt/10pt sans-serif;}	12pt/10pt - розмір шрифта/міжрядковий інтервал sans-serif - сімейство шрифтів
text-align	вирівнювання тексту	#b1{text-align: center;}	center - вирівнювання тексту по центру

6. Приклад виконання завдання (Рис. 3).

The diagram illustrates the structure of an HTML design based on blocks. It consists of three main parts:

- Visual Representation:** A simplified website layout with a header area labeled "Назва сайту", a navigation menu with "Домашня сторінка" and "Контент сайту", and a footer with "© Одеса 2018".
- HTML Code:** The code defines the structure using `<div>` elements with IDs `b1` through `b5`.


```

<!doctype html>
<html>
<head>
<title>Гіпертекстовий документ</title>
<meta charset="UTF-8">
<link rel="stylesheet" href="style/style.css">
</head>
<body>
<div id="b1">
<div id="b2">

<h1>Назва сайту</h1>
</div>
<div id="b3">
<ul>
<li><a href="/index.html">Домашня сторінка</a></li>
</ul>
</div>
<div id="b4">
<p>Контент сайту</p>
</div>
<div id="b5">
&copy; Одеса 2018
</div>
</body>
</html>

```
- CSS Styles:** The `style.css` file defines the styles for the blocks.


```

div
{border:solid 1px blue;
margin:2px;}
#b1
{width:100%;}
#b2
{background-color:grey;}
#b3
{background-color: grey;
width:25%;
float:left;}
#b4
{background-color:white;
float:left;
border:solid 0px blue;
margin:2px;}
#b5
{background-color: grey;
clear:both;}

```

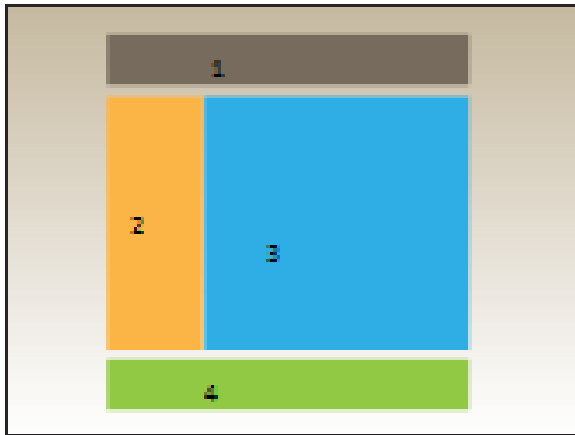
Arrows indicate the mapping from the visual layout to the HTML code and then to the CSS styles.

Рис.3. Структурування HTML дизайну на основі блоків

7. Індивідуальні завдання

Побудувати базовий шаблон сайту, що відповідає структурі, що наведена на рисунку.

Варіант 1



Блок 1:

Ширина = 1200px

Висота = 190px

Внутрішній відступ елемента = 10px

Нижній зовнішній відступ = 20px

Колір фону = brown

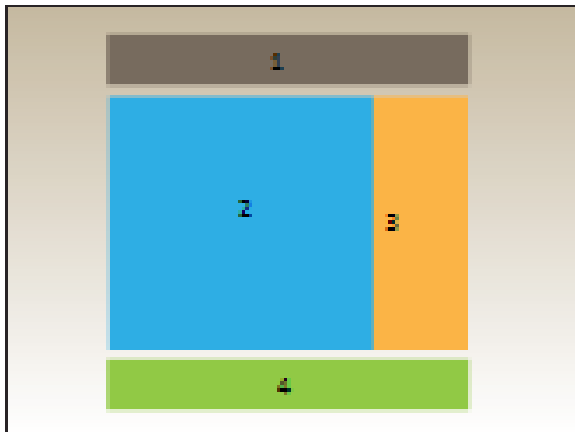
Самостійно визначити параметри блоків 2, 3 и 4 згідно зображення.

Для блока 3 визначити фоновий рисунок pic1.png, що розташований на відстані 10px від верхньої та лівої межі блока.

Визначити шрифт документа «Times New Roman» 14.

Текст в документі вирівняти по центру.

Варіант 2



Блок 1:

Ширина = 1200px

Висота = 190px

Внутрішній відступ елемента = 10px

Нижній зовнішній відступ = 20px

Колір фону = brown

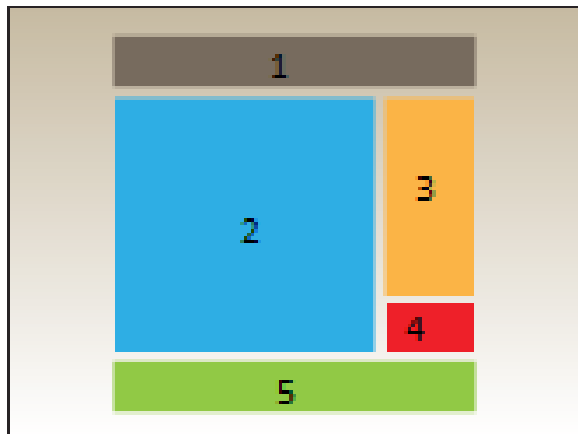
Самостійно визначити параметри блоків 2, 3 и 4 згідно зображення.

Для блока 3 визначити фоновий рисунок pic1.png, що розташований на відстані 10px від верхньої та лівої межі блока.

Визначити шрифт документа «Times New Roman» 14.

Текст в документі вирівняти по центру.

Варіант 3



Блок 1:

Ширина = 1200px

Висота = 190px

Внутрішній відступ елемента = 10px

Нижній зовнішній відступ = 20px

Колір фону = brown

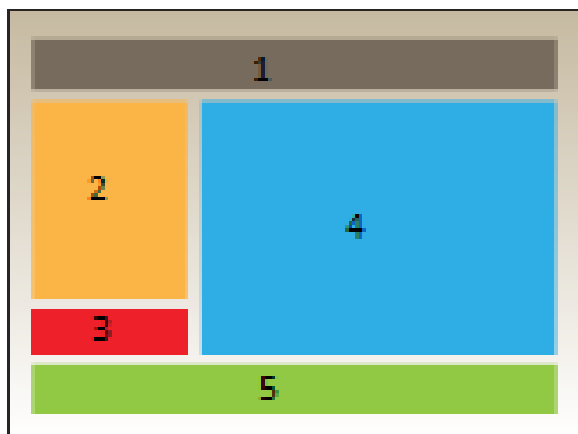
Самостійно визначити параметри блоків 2, 3 и 4 згідно зображення.

Для блока 3 визначити фоновий рисунок ріс1.png, що розташований на відстані 10px від верхньої та лівої межі блока.

Визначити шрифт документа «Times New Roman» 14.

Текст в документі вирівняти по центру.

Варіант 4



Блок 1:

Ширина = 1200px

Висота = 190px

Внутрішній відступ елемента = 10px

Нижній зовнішній відступ = 20px

Колір фону = brown

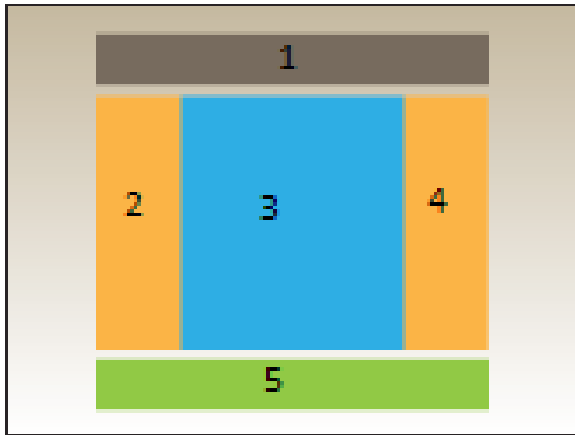
Самостійно визначити параметри блоків 2, 3 и 4 згідно зображення.

Для блока 3 визначити фоновий рисунок ріс1.png, що розташований на відстані 10px від верхньої та лівої межі блока.

Визначити шрифт документа «Times New Roman» 14.

Текст в документі вирівняти по центру.

Варіант 5



Блок 1:

Ширина = 1200px

Висота = 190px

Внутрішній відступ елемента = 10px

Нижній зовнішній відступ = 20px

Колір фона = brown

Самостійно визначити параметри блоків 2, 3 и 4 згідно зображення.

Для блока 3 визначити фоновий рисунок ріс1.png, що розташований на відстані 10px від верхньої та лівої межі блока.

Визначити шрифт документа «Times New Roman» 14.

Текст в документі вирівняти по центру.

Лабораторна робота №2

Мова php. Налаштування інтерфейсу виведення результатів. MVC-архітектура (Model-View-Controller) Web-орієнтованої системи.

Розгортання системи “MVC-старт”

Ціль роботи: Налаштування інтерфейсу виведення даних в Web-орієнтованій системі. Базова структура файлів частини контролера (Controller) та частини представлення (View). Огляд задіяних технологій.

Зміст.

1. Огляд технологій, що вивчатимуться.
 - a. Мова PHP.
 - b. Інструкція щодо користування хостингом
 - c. Базові підходи до виведення результатів роботи php програми. Приклад.
 - d. Модель MVC та шаблонізатор Twig.
2. Розгортаємо базові файли та папки системи “MVC-старт”.
3. Вивчаємо правила підключення нового шаблону до системи і налаштування варіативного блоку “content”.
4. Індивідуальні завдання.

Хід виконання лабораторної роботи.

1. Огляд технологій, що вивчатимуться.

1.а. Мова php - це мова серверного програмування, що означає:

- 1) Запуск програми здійснюється під керуванням Web-серверу
- 2) Результати виконання програми виводяться в гіпертекстовий інтерфейс
- 3) Для запуску програми php файли необхідно розмістити на хостингу

1.б. Інструкція щодо користування хостингом

- 1) Найкраще працювати з реальним хостингом. Безкоштовний хостинг на даний момент працює на [zzz.com.ua](https://www.zzz.com.ua) Інструкція щодо правил роботи. (<https://www.zzz.com.ua/ru/pomoshch>).
- 2) Альтернативою стає робота на віртуальному Web сервері Denwer (<http://www.denwer.ru/>). Інструкція щодо правил роботи (<http://forum.dklab.ru/denwer/>).

1.в. Базові підходи до виведення результатів роботи php програми:

- 1) Пряме виведення (через використання функції echo в php файлі)
- 2) Виведення в шаблон сторінки. В цьому випадку ми формуємо в php файлі 1-масив змінних, що будемо виводити. Формуємо 2-файл-шаблон, в який будемо виводити
- 3) Для пояснення сутності різних підходів до виведення даних розглянемо простий приклад: На основі наданих значень довжин сторін прямокутника знайти його периметр та площу і вивести на екран початкові дані і результати. Рішення даної задачі в мові програмування php принципово не відрізняється від рішення даної задачі в будь-якій мові програмування. Звичайний лінійний алгоритм.

<?php

...

```
$a=7;  
$b=8;  
$S=$a*$b;  
$P=2*($a+$b);
```

...

?>

Далі постає питання, куди виводити результат. Виявляється, що для виведення результату треба:

- 1) сформулювати гіпертекстову сторінку з якимось (може навіть найпростішим дизайном)
- 2) в даній сторінці визначити місце для виведення результатів рішення задачі
- 3) вивести отримані результати в визначене місце на гіпертекстовій сторінці з використанням HTML тегів для форматування даних, що виводяться.

Для остаточного рішення нашої задачі-прикладу візьмемо базову гіпертекстову сторінку наступної структури (Рис. 1)

Перший спосіб виведення результатів - це пряме виведення. Результируючий файл представлено на рисунку (Рис. 2)

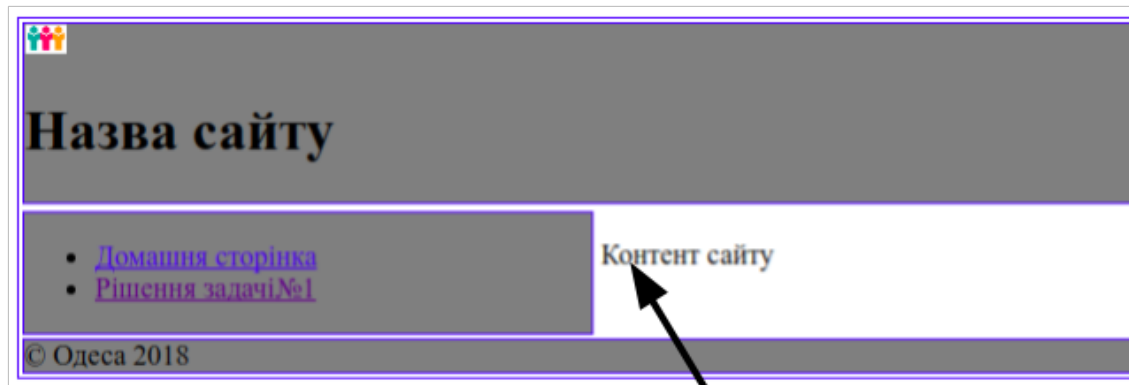
В чому постає проблема при даному варіанті реалізації виведення результатів:

1. При форматуванні нам приходится об'єднувати теги HTML форматування з php кодом, а значить програмна частина перетинається з дизайном сторінки
2. Виведення даних можливе саме у вказане місце, якщо виникає необхідність щось змінювати в іншій частині гіпертекстового документу, то засобів немає

Інший підхід - це розмежування частини контролеру та дизайну на основі MVC архітектури. Побачити як виглядає програмна реалізація даного приклада можна в моделі на рисунку (рис. 3)

Переглядаючи дві схеми (рис.2) та (рис.3) ми можемо бачити те, що в першому варіанті виведення даних шаблон гіпертекстової сторінки поділяється на 2 частини, а в моделі MVC в шаблоні виділяється варіативний іменований блок. Такі іменовані блоки можуть бути в довільному місці гіпертекстового документа і їх кількість не обмежена.

Подальше на практиці ми будемо користуватися моделлю MVC для виведення результатів роботи php скриптів. В файлі-шаблоні ми будемо користуватися синтаксисом бібліотеки Twig.



```

<!doctype html>
<html>
<head>
<title>Гіпертекстовий документ</title>
<meta charset="UTF-8">
<link rel="stylesheet" href="style.css">
</head>
<body>
<div id="b1">
<div id="b2">

<h1>Назва сайту</h1>
</div>
<div id="b3">
<ul>
<li><a href="index.html">Домашня сторінка</a></li>
<li><a href="lab1.html">Рішення задачі№1</a></li>
</ul>
</div>
<div id="b4">
<p>Контент сайту</p>
</div>
<div id="b5">
&copy; Одеса 2018
</div>
</div>
</body>
</html>

```

Місце для
виведення
результатів
рішення
задачі

Рис. 1. Визначення місця для виведення результатів рішення php задачі

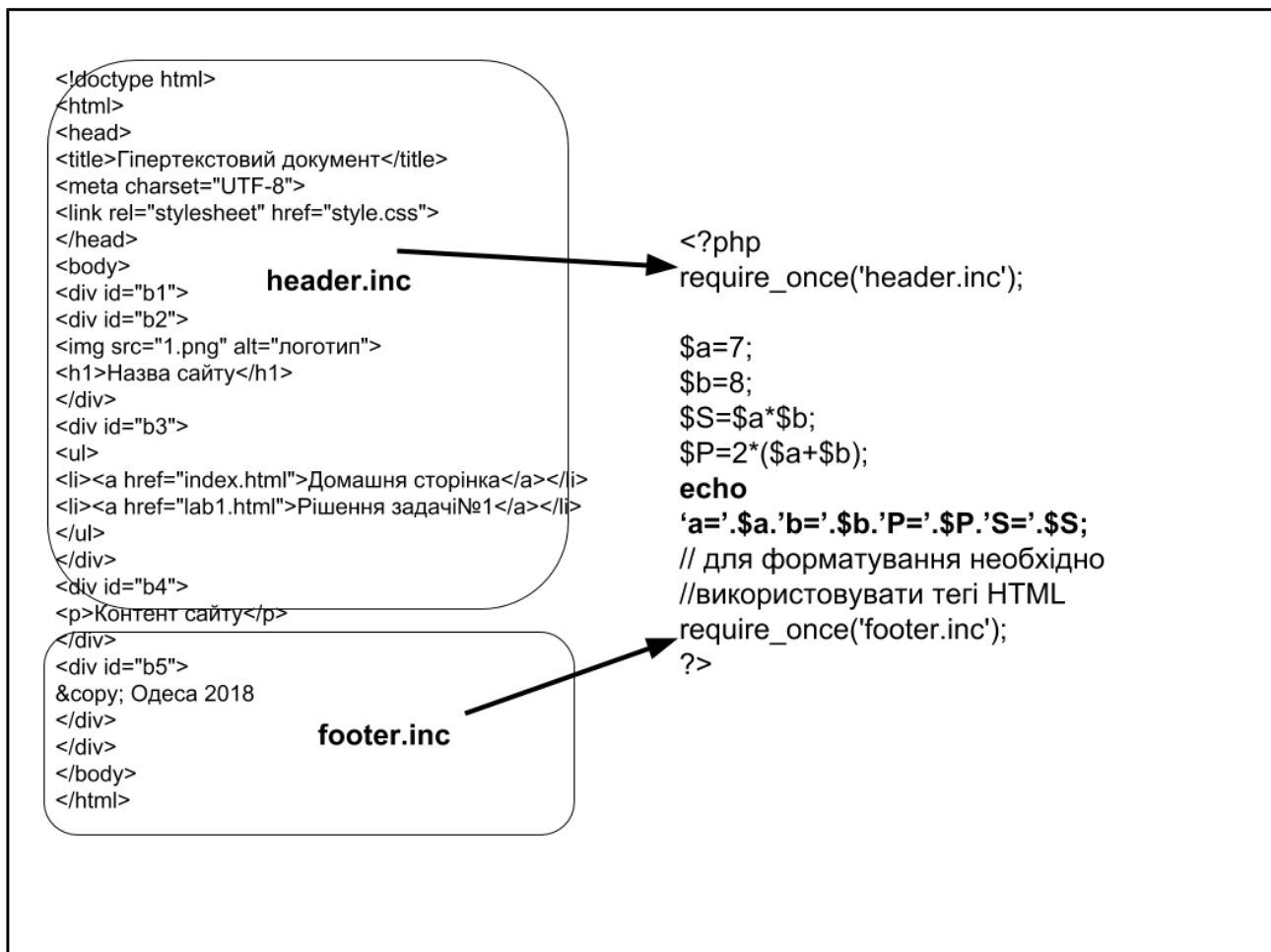


Рис. 2. Виведення результатів в визначене місце в шаблон гіпертекстової сторінки

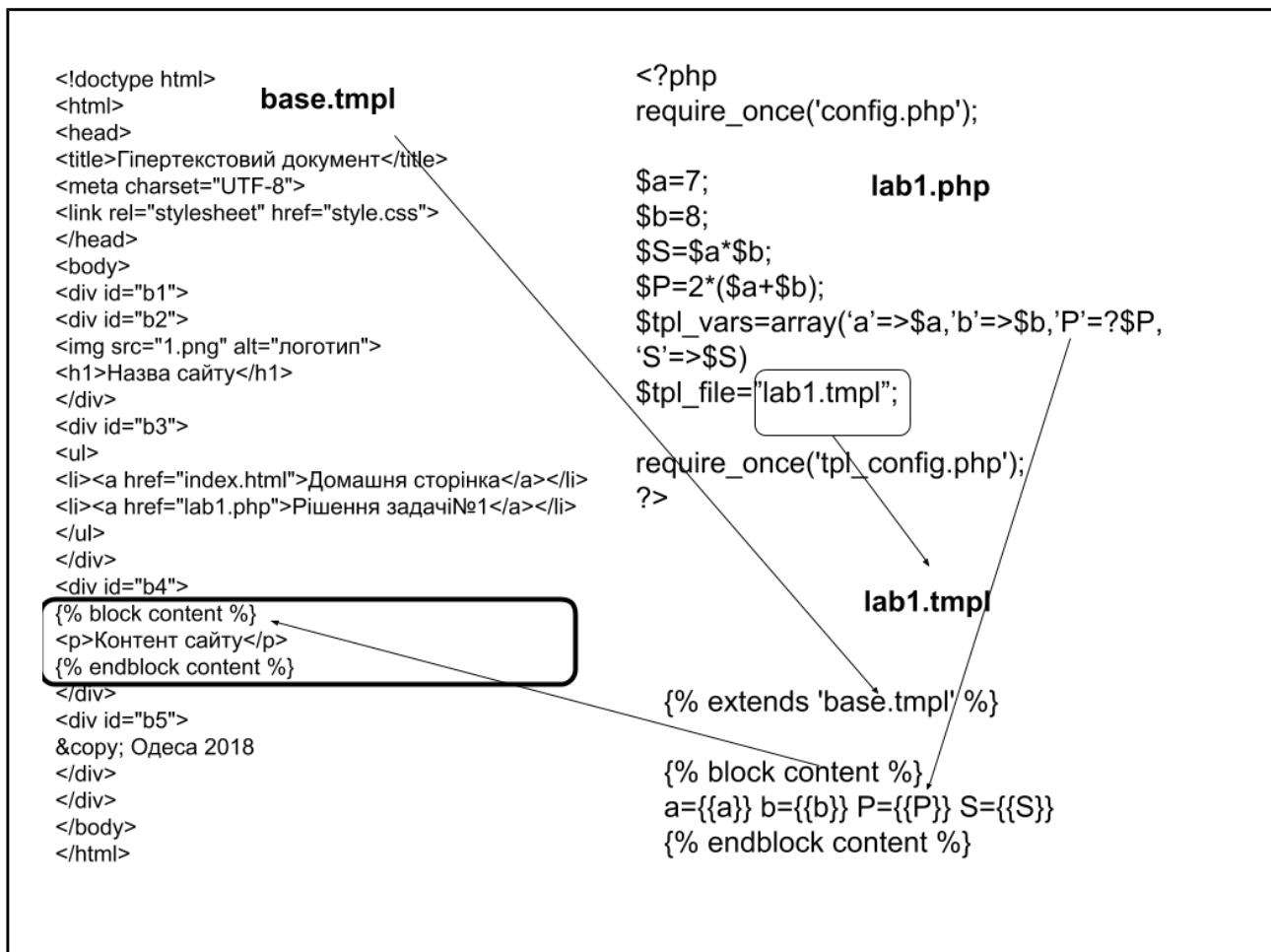


Рис. 3. Виведення результатів в іменованій варіативний блок

1.г. Модель MVC. Шаблонізатор Twig.

Стандартна схема архітектури MVC «Модель-Вид-Контролер» Model View Controller зображена на наступному рисунку (рис. 4).

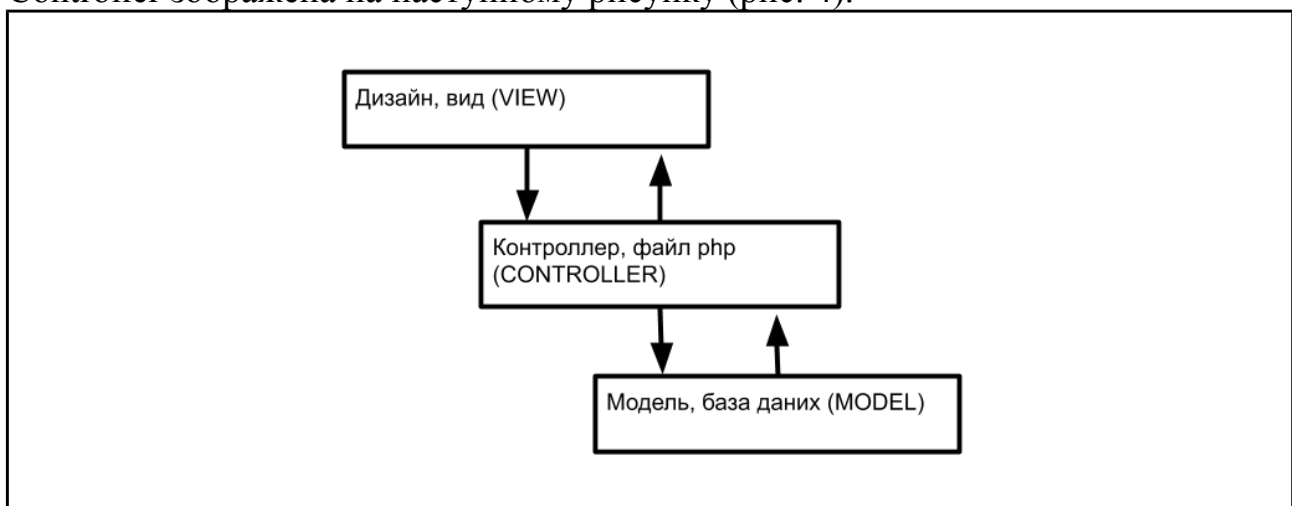


Рис. 4. Схема архітектури MVC

Розберемо по пунктах дану схему.

У моделі MVC, як впливає з назви, є три основних компоненти: Модель, Дизайн, і Контролер.

Дизайн (вид) відповідає за відображення інформації, що надходить із системи або в систему.

Модель є «суттю» системи і відповідає за дані системи, її інформаційну частину (як правило це база даних).

Контроллер є сполучною ланкою між «дизайном» і «моделлю» системи, за допомогою якого і існує можливість провести поділ між ними. Контроллер отримує дані від користувача і передає їх в «модель». Крім того, він отримує повідомлення від моделі, і передає їх в «дизайн».

Стосовно до інтернет-додатків здається, що частини контроллер і дизайн об'єднані, тому що за відображення і одночасно за введення інформації відповідає браузер.

Роз'єднати ці дві частини дозволяє сучасна технологія використання шаблонизатора.

Шаблонизатор дозволяють домогтися відділення прикладної логіки і даних від логіки представлення. Це дуже зручно в ситуаціях, коли програміст і верстальник шаблону - різні люди. Шаблонизатор сприяє виконанню концепції поділу.

Довгий час для розробки сайтів на PHP використовував шаблонизатор Smarty. Досить зручний шаблонизатор, так би мовити де-факто для PHP. Паралельно розвивалися інші шаблонизатор. Розглянемо наступну порівняльну таблицю шаблонизаторів за швидкодією [2]:

Бібліотека	Час (сек)	Пам'ять (Кб)
Мова PHP	2.4	114
Twig	3	383
PHPTAL	3.8	598
Dwoo	6.9	1 645
Smarty 2	12.9	610
Smarty 3	14.9	799
Calypso	34.3	614
eZ Templates	53	2 783

Twig шаблонизатор для PHP є одним з найбільш зручних і швидкодійних.

2. Розгортаємо базові файли та папки системи “MVC-старт”

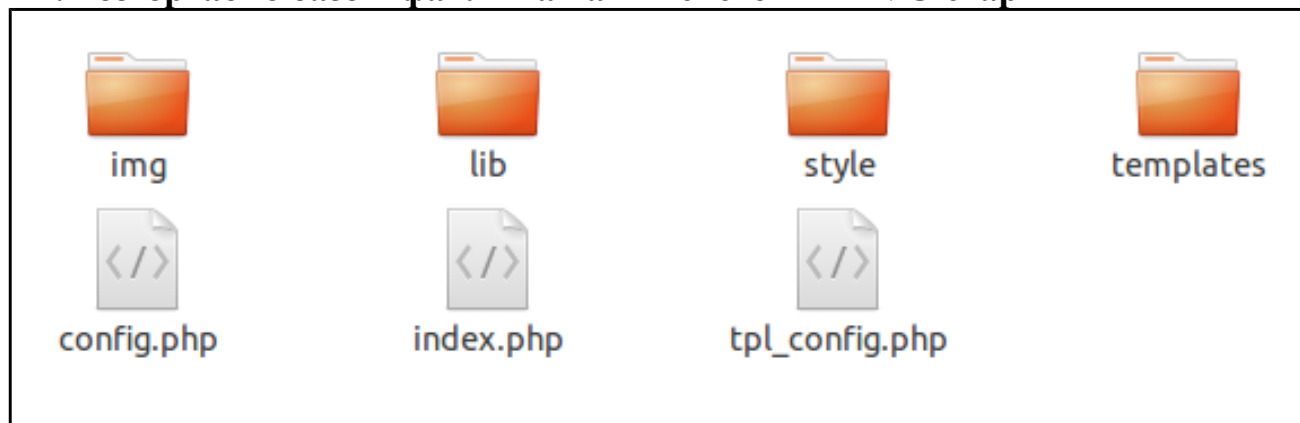
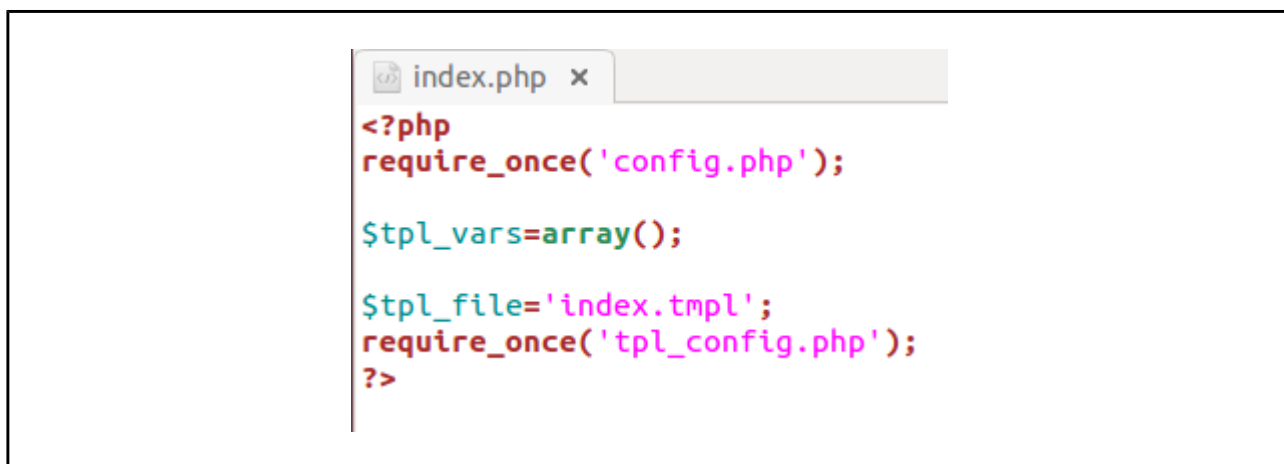


Рис. 5. Файли та папки системи “MVC-старт”

Розглядаючи підсумкову структуру файлів та папок системи “MVC-старт” (Рис. 5) визначимо їх структурні компоненти:

1 - "index.php" це приклад файлу системи, що відповідає за рішення задачі мовою php, формування масиву результатів, що мають виводитись в гіпертекстовому інтерфейсі та визначає файл-шаблон, в якому буде виведено дані (Рис.6)



```
index.php x
<?php
require_once('config.php');

$tpl_vars=array();

$tpl_file='index.tpl';
require_once('tpl_config.php');
?>
```

Рис 6. Зміст файлу index.php

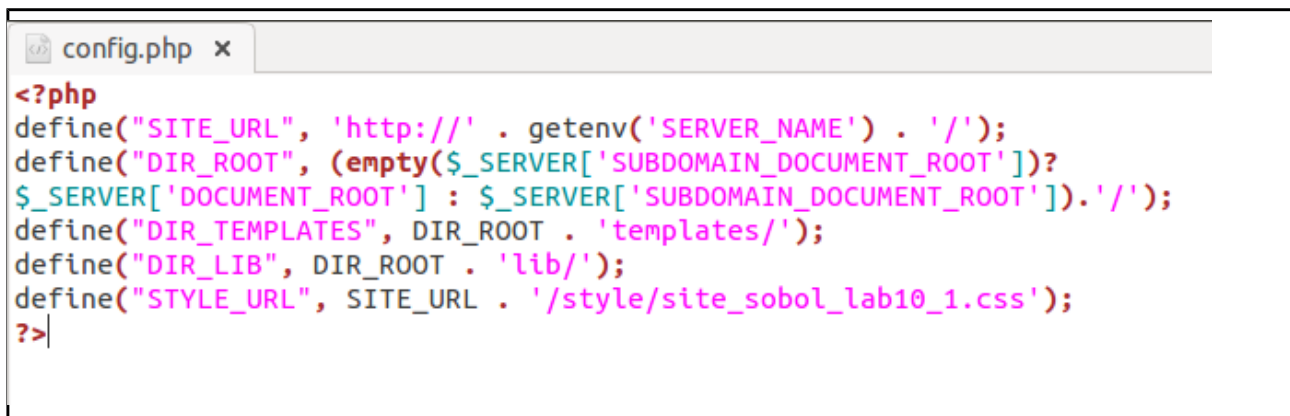
```
// Підключення конфігураційного файлу
require_once ('config.php');
/* Нижче визначаємо масив змінних доступних в шаблоні. У першому прикладі
даний масив порожній*/
$tpl_vars = array ();
/* Нижче визначаємо ім'я файлу-шаблону, який знаходиться в папці templates */
$tpl_file = 'index.tpl';
// Запускаємо шаблонизатор і виводимо відповідну сторінку
require_once ('tpl_config.php');
В папці templates повинен бути створений файл index.tpl
```

2 - "config.php" - визначення констант, які пов'язані зі шляхами і адресами сайту, при перенесенні сайту на новий хостинг (Рис.7)

Першим кроком на шляху побудови професійного сайту є створення файлу config.php, в котрому ми налаштовуємо автоматичне визначення URL адреси сайту, що дозволяє без змін в налаштуваннях переносити сайт на новий хостинг. Всі константи, що визначаються в конфігураційному файлі інтуїтивно зрозумілі за своїми назвами. Для їх визначення використано стандартні функції php.

getenv('SERVER_NAME') повертає ім'я хоста з URL адреси сайту.

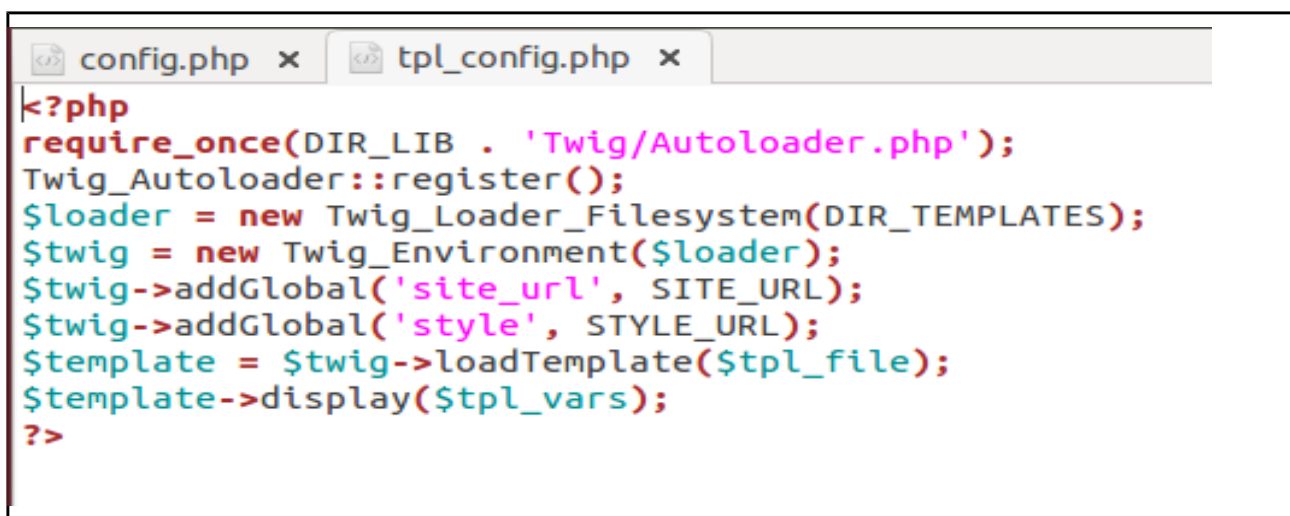
\$_SERVER['SUBDOMAIN_DOCUMENT_ROOT'] з масиву змінних оточення визначаємо повний шлях на сервер до кореневого каталогу сайту (в залежності від налаштувань Web серверу це може бути \$_SERVER['DOCUMENT_ROOT']). У випадку розташування сайту у вкладених папках на хостингу, необхідно дописувати шлях до основної деректорії сайту.

A screenshot of a code editor window with a single tab titled 'config.php'. The code is in PHP and defines several constants for site configuration. The code is as follows:

```
<?php
define("SITE_URL", 'http://' . getenv('SERVER_NAME') . '/');
define("DIR_ROOT", (empty($_SERVER['SUBDOMAIN_DOCUMENT_ROOT'])?
$_SERVER['DOCUMENT_ROOT'] : $_SERVER['SUBDOMAIN_DOCUMENT_ROOT']).'/');
define("DIR_TEMPLATES", DIR_ROOT . 'templates/');
define("DIR_LIB", DIR_ROOT . 'lib/');
define("STYLE_URL", SITE_URL . '/style/site_sobol_lab10_1.css');
?>
```

Рис. 7. Стартова структура файлу config.php

3 - "tpl_config.php" - визначає базові команди по підключенню бібліотеки "Twig", які повинні бути виконані в кожному файлі частині контролю (Рис.8)

A screenshot of a code editor window with two tabs: 'config.php' and 'tpl_config.php'. The 'tpl_config.php' tab is active, showing PHP code that initializes the Twig templating engine. The code is as follows:

```
<?php
require_once(DIR_LIB . 'Twig/Autoloader.php');
Twig_Autoloader::register();
$loader = new Twig_Loader_Filesystem(DIR_TEMPLATES);
$twig = new Twig_Environment($loader);
$twig->addGlobal('site_url', SITE_URL);
$twig->addGlobal('style', STYLE_URL);
$template = $twig->loadTemplate($tpl_file);
$template->display($tpl_vars);
?>
```

Рис.8. Файл tpl_config.php

Нижче наведено пояснення до команд у файлі tpl_config.php

```
// Підключаємо необхідні бібліотечні файли
require_once(DIR_LIB . 'Twig/Autoloader.php');
//Запускаємо Twig
Twig_Autoloader::register();
//Визначаємо путь до папки з шаблонами
$loader = new Twig_Loader_Filesystem(DIR_TEMPLATES);
/*Визначаємо шлях до папки cache (тимчасові файли), яка має бути відкрита на
запис. Саме в цій папці формуються php файли з шаблонів при першому
запуску.*
$twig = new Twig_Environment($loader);
/*Два наступних рядка роблять доступними відповідні глобальні константи
конфігураційного файлу у кожному шаблоні*/
$twig->addGlobal('site_url', SITE_URL);
$twig->addGlobal('style', STYLE_URL);
```

```
/*Далі визначаємо ім'я змінної, що зберігає шлях до файла-шаблону відносно папки templates. Змінна $tpl_file має визначатися в кожному php файлі системи */
```

```
$template = $twig->loadTemplate($tpl_file);
```

```
/*Далі виводимо на екран результати, що передано через масив $tpl_vars. Масив $tpl_vars має бути визначений в кожному php файлі.*/
```

```
$template->display($tpl_vars);
```

4 - "lib" - це папка, що містить файли бібліотеки "Twig".

На сайті Twig (twig.symfony.com) — вказано, що для роботи необхідним є **PHP 5.2.4** і вище.

Далі необхідно встановити бібліотеку Twig, останньою на момент публікації даних матеріалів є версія 1.10.3 (<http://twig.sensiolabs.org/>)[3]. З архіву нам необхідна папка **Twig**, що знаходиться в папці **lib**. Її ми копіюємо в свою папку **lib**. Отримуємо на хостингу:

lib/-

|_ Twig

Необхідно розташувати в корні власного сайту папку lib і можна починати роботу з шаблонизатором **Twig**. Папка lib має знаходитись в кореневому каталозі сайту.

5 - "templates" - це папка, що містить базовий шаблон і шаблони всіх сторінок системи. На старті тут має бути файл base.tpl та шаблон index.tpl сторінки index.php (Рис. 9).



Рис. 9. Стартова структура папки з шаблонами

На рисунку (рис. 10) наведено правила взаємодії файлів частини дизайну і правила підключення дизайну до файлу частини контроллера (index.php)

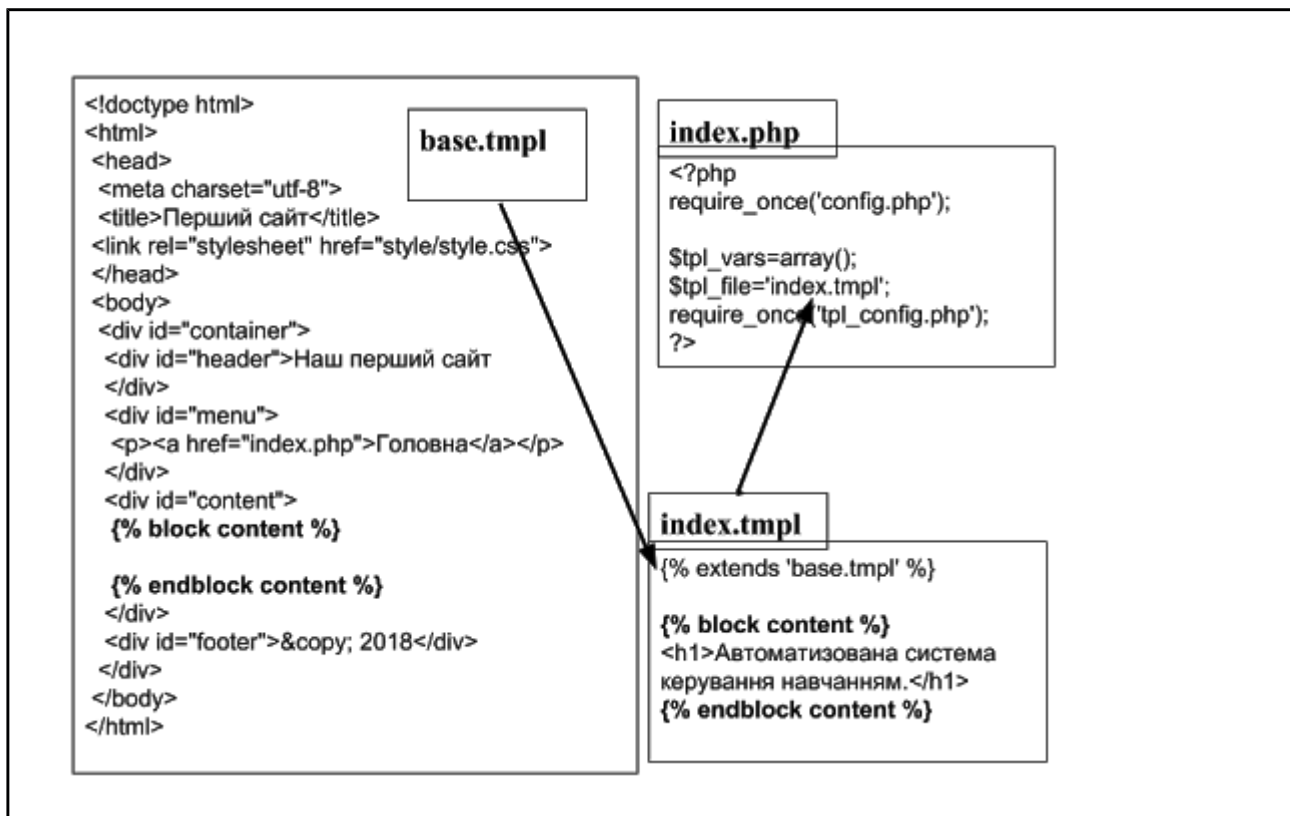


Рис. 10. Правила взаємодії файлів частини дизайну і правила підключення дизайну до файлу частини контролеру

7 - Папки "img" і "style" без зміни переносяться зі статичного варіанту сайту. Для стартової системи MVC-старт папка img пуста, а в папці "style" знаходиться файл style.css (рис. 11)

3. Вивчаємо правила підключення нового шаблону до системи і налаштування варіативного блоку "content" (рис. 10)

Шаблон **base.tpl**—це базовий шаблон. На його основі будуть розроблятися шаблони до всіх php файлів сайту

Для переведення сайту до роботи в моделі MVC файл base.tpl розробляється на основі файла base.html з додаванням наступних змін:

1. Підключення файла зі стилями в файлі base.html

```
<link rel="stylesheet" href="style/style.css" >
```

перетворюється до вигляду

```
<link rel="stylesheet" href="{{style}}" >
```

Зверніть увагу {{style}} - це виведення глобальної змінної, яка визначена в файлі tpl_config.php

(\$twig->addGlobal('style', STYLE_URL);) (Рис. 7)

STYLE_URL – це константа, що була визначена в файлі config.php
define("STYLE_URL", SITE_URL . '/style/style.css') (Рис. 6)

2. В блоці "content"

```
<div id="content">
```

```
</div>
```

необхідно додати варіативний блок шаблонизатора Twig:

```
<div id="content">
{% block content %}
```

```
{% endblock content %}
</div>
```

content – це ім'я блоку і воно може бути довільним часто його назва збігається з ім'ям зовнішнього блоку div

Кількість таких динамічних блоків в файлах-шаблонах може бути довільною.

Кожен такий блок може бути перевизначений в новому файлі-шаблоні, побудованому на основі шаблону base.tpl

```
#container
{
    width: 100%; /* Ширина сайту */
    margin: 0 auto; /* Вирівнювання по центру */
    background: #f0f0f0; /* Колір фону лівої колонки */
}
#header
{
    font-size: 2.2em; /* Розмір тексту */
    text-align: center; /* Вирівнювання по центру */
    padding: 5px; /* Відступи навколо тексту */
    background: #8fa09b; /* Колір фону шапки */
    color: #ffe; /* Колір тексту */
}
#menu
{
    margin-top: 10px;
    width: 20%; /* Ширина блоку */
    padding: 0 10px; /* Відступи навколо тексту */
    float: left; /* Обтікання по правому краю */
}
#content
{
    margin-left: 130px; /* Відступ зліва */
    padding: 80px; /* Поля навколо тексту */
    background: #fff; /* Колір фону блоку-контенту */
}
#footer
{
    background: #8fa09b; /* Колір фону футеру */
    color: #fff; /* Колір тексту */
    padding: 5px; /* Відступ навколо тексту */
    clear: left; /* Відміняємо дію float */
}
```

Рис. 11. Приклад структури файлу style.css

Індивідуальне завдання

1. Зареєструвати власний хостинг (реальний або віртуальний, безкоштовний або платний). Вивчити правила роботи з даним хостингом.
2. Завантажити стартові файли та папки системи “MVC-старт” на хостинг. Посилання для завантаження архіву системи “MVC-старт”:
<http://pdpu.edu.ua/doc/mvc/mvc.zip>
3. Підготувати 5 нових сторінок в системі, що мають імена lab1-1.php - lab1-5.php. Для кожної сторінки налаштувати індивідуальний шаблон, на якому інформацію про себе структуровано з використанням різних об'єктів HTML.
lab1-1.tmpl використати об'єкти заголовок (h1-h6) та абзац (p)
lab1-2.tmpl використати об'єкт список (ul, ol) та рисунок (img)
lab1-3.tmpl використати об'єкт таблиця (table)
lab1-4.tmpl використати об'єкти заголовок (h1-h6), рисунок (img) та & послідовність
lab1-5.tmpl використати об'єкти заголовок (h1-h6) та таблиця (table)

Лабораторна робота №3

Мова PHP. Побудова лінійних алгоритмів

Ціль: Рішення задач на побудову лінійних алгоритмів в мові php. Типи даних мови php, приведення типів, вирази.

Зміст

- 1) Основи мови PHP.
 - a. Вступ до мови PHP.
 - b. Константи.
 - c. Змінні. Приведення типів.
 - d. Оператори
 - e. Перевірка вмісту змінної
- 2) Передача скалярних значень з частини контролера в частину дизайну (шаблону) MVC моделі.
- 3) Виведення значень, що було передано, в шаблоні шаблонизатору Twig
- 4) Індивідуальні завдання.

Хід роботи.

1) Основи мови PHP.

PHP - мова створення сценаріїв. PHP - це аббревіатура від слів Personal Home Page. Перша версія PHP була створена Расмусом Лердорфом в 1994 році і представляла собою набір інструментів для відстеження відвідувачів Web-сторінки. Згодом PHP з набору інструментів перетворився в повноцінний мову програмування, а його назва була змінена PHP HyperText Preprocessor (препроцесор гіпертексту PHP).

PHP - це серверна мова створення сценаріїв. Конструкції PHP, вставлені в HTML-текст, виконуються сервером при кожному відвідуванні сторінки. Результат їх обробки разом зі звичайним HTML-текстом передається браузеру.

В даний час основними версіями PHP є 5 + та 7+.

Розробку PHP-програми можна відокремити від власне розробки Web-сторінки, що спростить життя і програмісту, і дизайнеру.

PHP є абсолютно безкоштовним.

Один і той же PHP-код можна використовувати на різних платформах.

PHP-програми складаються з простого тексту, тому набирати їх можна в будь-якому текстовому редакторі. Популярні HTML-редактори мають вбудовану підтримку для редагування PHP-програм.

Розширення файлів PHP-програм за замовчуванням - .php. На підставі цього розширення сервер розпізнає файл як PHP-програму і запускає інтерпретатор.

PHP-програма повинна бути відокремлена `<?php` `?>`

ЗАПУСК ПРОГРАМИ НА МОВІ PHP ВИКОНУЄТЬСЯ З СЕРВЕРА (з URL адреси)

Для поділу операторів (по аналогії з Cі) використовують крапку з комою.

Коментарі в PHP-програмою можуть бути трьох стилів:

/ * Багаторядковий коментар

в стилі класичного Cі * /

// Однорядковий коментар в стилі C ++

Однорядковий коментар в стилі Perl

Константи визначаються в PHP-програмою за допомогою функції `define()`.

Наприклад:

```
define("PI", 3.1415927);
```

Після визначення константи не може бути змінена. В імені константи зазвичай використовуються лише великими літерами

Змінні. Ім'я будь-якої змінної в PHP починається зі знака \$. Імена змінних чутливі до регістру символів.

Тип змінної не потрібно задавати спеціально. Конкретний тип змінної встановлюється і змінюється в ході виконання програми.

PHP підтримує вісім типів даних:

- логічний (приймає значення `true` або `false`);
- ціле число;
- дійсне число з плаваючою точкою;
- рядок;
- об'єкт;
- масив;
- ресурс (спеціальний тип);
- `null` (спеціальний тип).

Тип змінної можна перевірити за допомогою функції `gettype()`. В якості аргумента надається змінна, функція повертає її тип. Наприклад:

```
$type1=gettype($t);
```

Тип змінної можна змінити з допомогою функції **`settype()`**. В якості аргументів вказується змінна, новий тип. Наприклад:

```
settype($t,"integer");
```

Приведення типу змінної здійснюється за допомогою операторів:

- *(bool)* - до логічного типу;
- *(int)* - до цілого числа;
- *(double)* - до матеріального числа;
- *(string)* - до рядка;
- *(array)* - до масиву;
- *(object)* - до об'єкту.

Відмінність від зміни типу за допомогою функції `settype()` полягає в тому, що оператор приведення створює тимчасову копію нового типу, залишаючи саму змінну без змін.

У PHP можливе звернення до однієї і тієї ж змінної з використанням різних імен. Для реалізації цього використовуються посилання. Посилання дозволяють двом або більшій кількості змінних посилатися на одну і ту ж область пам'яті.

```
$t1 = 96;
```

```
$t2 = &$t1;  
$t1 = 315;//автоматично змінено значення $t2
```

Оператори РНР нагадують загальновідомі оператори мови Сі.

унарні оператори

- Зміна знака на протилежний
! Доповнення. Використовується для реверсування значення логічних змінних
++ Збільшення значення змінної. Може застосовуватися і як префікс, і як суфікс
- Зменшення значення змінної. Може застосовуватися і як префікс, і як суфікс

Арифметичні оператори (бінарні оператори)

- віднімання
- + додавання
 - * множення
 - / ділення
 - % остача від цілочисельного ділення

Оператор конкатенації (бінарний оператор)

Оператор конкатенації "." приєднує праву рядок до лівої.

1 Оператори присвоєння

- = Присвоїте
- += Додавання (\$n += 777; аналогічно \$n = \$n + 777;)
- = Віднімання (\$n -= 777; аналогічно \$n = \$n - 777;)
- *= Множення
- /= Ділення
- %= Остача від цілочисельного ділення
- .= Конкатенація (\$n .= "777"; аналогічно \$n = \$n."777";)

2 Бітові оператори

Бітові оператори дозволяють змінити окремі біти цілих чисел.

- & І
- | АБО
- ^ Виключаюче АБО
- ~ Інверсія
- >> Здвиг праворуч
- << Здвиг ліворуч

3 Оператори порівняння

- > Більше
- >= Більше або рівно
- < Менше
- <= Менше або рівно
- == Рівно
- === Ідентично (Однакові значення і типи двох змінних?)
- != , Не рівно
- <> Не ідентично

4 Логічні оператори

Логічні оператори відрізняються від бітових тим, що опрацьовують не числа, а значення: TRUE и FALSE.

and	I
or	АБО
xor	Виключаюче АБО
!	Інверсія
>>	Здвиг праворуч
<<	Здвиг ліворуч
&&	I
	АБО

Логічні оператори "I" і "АБО" мають два формати. Це не синоніми. Справа в тому, що оператор or має пріоритет нижче, ніж ||, а and - нижче, ніж &&. Таким чином, при побудові складних умовних виразів можна обійтися без дужок. Однак, в такому способі вказівки порядку простіше і заплутатися.

Перевірка вмісту змінної

Іноді необхідно перевірити, чи існує змінна або яке вона має значення. Нижче наведені функції, що дозволяють виконати такі дії.

`isset($ім'я_змінної)` # Істина, якщо змінна оголошена навіть без надання значення.

`empty($ім'я_змінної)` # Істина, якщо значення змінної дорівнює нулю або порожньому рядку, або змінна не оголошена.

PHP також дозволяє перевірити тип змінної. Наприклад, для того щоб перевірити, чи є змінна цілочисленною, слід скористатися функцією

`is_int($number)`

Результатом виконання цієї функції є TRUE, якщо змінна \$ number має тип Завдання. Розглянемо подібні функції.

`is_array($var2)` перевіряє, чи є змінна \$var2 масивом.

`is_float($number)` перевіряє, чи є змінна \$number числом з плаваючою крапкою.

`is_null($var1)` перевіряє, чи дорівнює значення змінної \$var1 нулю

`is_numeric($string)` перевіряє, чи є змінна \$string числовий рядком.

`is_string($string)` перевіряє, чи є змінна \$string рядком.

Для перевірки зворотних умов слід скористатися символом вигуки (!). Наприклад, при обробці наступного виразу буде отримано значення TRUE, якщо змінна не оголошена: `!isset($ім'я_змінної)`

2) Передача скалярних значень з частини контроллера в частину дизайнера MVC моделі.

В масиві `tpl_vars` визначаються доступні в шаблоні змінні `name`, `username`, `password` (Рис. 1 Файл `lab1.php` ВАРІАНТ 1)

В змінній `$tpl_file` визначаємо шлях до файлу-шаблону `lab1.tpl` (Рис. 1 Файл `lab1.php` ВАРІАНТ 1)

На рисунку 1 файл lab1.php реалізований в другому варіанті для демонстрації можливого визначення значень змінних в мові php та подальше їх включення в масив tpl_vars (Рис. 1 Файл lab1.php ВАРИАНТ2)

2. Вивід значень, що передано в шаблон. Синтаксис шаблонизатора Twig.

Передані в шаблон змінні виводяться за допомогою наступної конструкції {{name}}, {{username}}, {{password}} (Рис. 1 Файл lab1.tpl).

Зверніть увагу на той факт, що при форматуванні даних використовуються теги HTML (Рис. 1 Файл lab1.tpl) та можливість підключення CSS стилів.



Рис. 1. Приклад передачі значень з частини контроллера в шаблон

4) Індивідуальні завдання.

1. Всі вхідні дані визначати конкретними значеннями змінних у файлі lab3.php (без інтерфейсу введення).
2. У шаблон lab3.tpl передати значення наступних змінних
 - строкову змінну з умовою завдання
 - значення всіх змінних, що визначають вхідні дані
 - змінні з результатами рішення задачі.

3. У шаблоні оформити виведення переданих значень з використанням тегів і стилів.

4. Рішити дві задачі. 1 задача - ваш порядковий номер в групі, 2 задача- номер в групі +15.

Задача1. Дана сторона квадрата a . Знайти його периметр $P = 4 \cdot a$. Вивести умову задачі, початкові значення та результуючі.

Задача2. Дана сторона квадрата a . Знайти його площу $S = a \cdot a$. Вивести умову задачі, початкові значення та результуючі.

Задача3. Дано сторони прямокутника a і b . Знайти його площу $S = a \cdot b$ і периметр $P = 2 \cdot (a + b)$. Вивести умову задачі, початкові значення та результуючі.

Задача4. Дан діаметр кола d . Знайти його довжину $L = \pi \cdot d$. Як значення π використовувати 3.14. Вивести умову задачі, початкові значення та результуючі.

Задача5. Дана довжина ребра куба a . Знайти об'єм куба $V = a \cdot a \cdot a$ і площу його поверхні $S = 6 \cdot a \cdot a$. Вивести умову задачі, початкові значення та результуючі.

Задача6. Дано довжини ребер a , b , c прямокутного паралелепіпеда. Знайти його об'єм $V = a \cdot b \cdot c$ і площа поверхні $S = 2 \cdot (a \cdot b + b \cdot c + a \cdot c)$. Вивести умову задачі, початкові значення та результуючі.

Задача7. Знайти довжину кола L і площу круга S заданого радіуса R :

$$L = 2 \cdot \pi \cdot R, S = \pi \cdot R \cdot R.$$

Як значення π використовувати 3.14. Вивести умову задачі, початкові значення та результуючі.

Задача8. Дано два числа a і b . Знайти їх середнє арифметичне: $(a + b) / 2$. Вивести умову задачі, початкові значення та результуючі.

Задача9. Дано два ненульових числа. Знайти суму, різницю, добуток і частку їх квадратів. Вивести умову задачі, початкові значення та результуючі.

Задача10. Дано два кола із загальним центром і радіусами $R1$ і $R2$ ($R1 > R2$). Знайти площі цих кіл $S1$ і $S2$, а також площа $S3$ кільця, зовнішній радіус якого дорівнює $R1$, а внутрішній радіус дорівнює $R2$:

$$S1 = \pi \cdot R1 \cdot R1, S2 = \pi \cdot R2 \cdot R2, S3 = S1 - S2.$$

Як значення π використовувати 3.14. Вивести умову задачі, початкові значення та результуючі.

Задача11. Дана довжина L кола. Знайти його радіус R і площу S круга, обмеженого цим колом, враховуючи, що $L = 2 \cdot \pi \cdot R$, $S = \pi \cdot R \cdot R$. Як значення π використовувати 3.14. Вивести умову задачі, початкові значення та результуючі.

Задача12. Дана площа S круга. Знайти його діаметр D і довжину L кола, що обмежує круг, з огляду на, що $L = \pi \cdot D$, $S = \pi \cdot (D \cdot D) / 4$. Як значення π використовувати 3.14. Вивести умову задачі, початкові значення та результуючі.

Задача13. Дано три точки A , B , C числової осі. Знайти довжини відрізків AC і BC і їх суму. Вивести умову задачі, початкові значення та результуючі.

Задача14. Дано три точки A , B , C числової осі. Точка C розташована між точками A і B . Знайти довжини відрізків AC і BC . Вивести умову задачі, початкові значення та результуючі.

Задача15. Дано координати двох протилежних вершин прямокутника: (x_1, y_1) , (x_2, y_2) . Сторони прямокутника паралельні осям координат. Знайти периметр і площу даного прямокутника. Вивести умову задачі, початкові значення та результуючі.

Задача16. Дано відстань L в сантиметрах. Використовуючи операцію ділення націло, знайти кількість повних метрів в ньому (1 метр = 100 см). Вивести умову задачі, початкові значення та результуючі.

Задача17. Дана маса M в кілограмах. Використовуючи операцію ділення націло, знайти кількість повних тонн в ній (1 тонна = 1000 кг). Вивести умову задачі, початкові значення та результуючі.

Задача18. Дан розмір файлу в байтах. Використовуючи операцію ділення націло, знайти кількість повних кілобайт, які займає даний файл (1 кілобайт = 1024 байта). Вивести умову задачі, початкові значення та результуючі.

Задача19. Дано цілі додатні числа A і B ($A > B$). На відрізку довжиною A розміщено максимально можлива кількість відрізків довжиною B (без накладень). Використовуючи операцію ділення націло, знайти кількість відрізків B , розміщених на відрізку A . Вивести умову задачі, початкові значення та результуючі.

Задача20. Дано цілі додатні числа A і B ($A > B$). На відрізку довжиною A розміщено максимально можлива кількість відрізків довжиною B (без накладень). Використовуючи операцію взяття залишку від ділення без остачі, знайти довжину незайнятої частини відрізка A . Вивести умову задачі, початкові значення та результуючі.

Задача21. Дано двозначне число. Вивести спочатку його ліву цифру (десятки), а потім - його праву цифру (одиниці). Для знаходження десятків використати операцію ділення націло, для знаходження одиниць - операцію взяття залишку від ділення націло. Вивести умову задачі, початкові значення та результуючі.

Задача22. Дано двозначне число. Знайти суму і добуток його цифр. Вивести умову задачі, початкові значення та результуючі.

Задача23. Дано двозначне число. Вивести число, отримане при перестановці цифр вхідного числа. Вивести умову задачі, початкові значення та результуючі.

Задача24. Дано тризначне число. Використовуючи одну операцію ділення націло, вивести першу цифру даного числа (сотні). Вивести умову задачі, початкові значення та результуючі.

Задача25. Дано тризначне число. Вивести спочатку його останню цифру (Одиниці), а потім - його середню цифру (десятки). Вивести умову задачі, початкові значення та результуючі.

Задача26. Дано тризначне число. Знайти суму і добуток його цифр. Вивести умову задачі, початкові значення та результуючі.

Задача27. Дано тризначне число. Вивести число, отримане при прочитанні вхідного числа справа наліво. Вивести умову задачі, початкові значення та результуючі.

Задача28. Дано тризначне число. У ньому закреслили першу зліва цифру і приписали її справа. Вивести отримане число. Вивести умову задачі, початкові значення та результуючі.

Задача29. Дано тризначне число. У ньому закреслили першу праворуч цифру і приписали її зліва. Вивести отримане число. Вивести умову задачі, початкові значення та результуючі.

Задача30. Дано тризначне число. Вивести число, отримане при перестановці цифр сотень і десятків вхідного числа (наприклад, 123 перейде в 213). Вивести умову задачі, початкові значення та результуючі.

Лабораторна робота №4

Алгоритми розгалуження в мові php.

Ціль: Рішення задач на побудову алгоритмів розгалуження. Використання конструкції розгалуження в шаблонизаторі Twig

Зміст

- 1) Керуючі структури. Оператор розгалуження у мові PHP
- 2) Оператор розгалуження в шаблонизаторі Twig. Умовні вирази шаблонизатора Twig.
- 3) Приклад рішення задачі.
- 4) Індивідуальні завдання.

Хід роботи.

- 1) Керуючі структури. Оператор розгалуження у мові PHP

Умовні оператори

У мові PHP два умовних оператора: if і ?.

Існує три типи оператора if. Перший тип - базовий умовний оператор. Приклад 1

```
if ($a > $b)
{
    $result = "А більше Б";
}
```

Альтернативний синтаксис оператора **if**:

```
if ($a > $b):
    $result = "А більше Б";
endif;
```

Другий тип - умовний оператор **if-else**.

Приклад 2

```
if ($a > $b)
{
    $result = "А більше Б";
}
else
{
    $result = "А не більше Б";
}
```

Альтернативний синтаксис оператора **if-else**:

```
if ($a > $b):
    $result = "А більше Б";
else:
    $result = "А не більше Б";
endif;
```

Третій тип - умовний оператор **if-elseif**.

Приклад 3

```
if ($a > $b)
{
    $result = "А більше Б";
}
elseif ($a == $b)
{
    $result = "А дорівнює Б";
}
else
{
    $result = "А менше Б";
}
```

Альтернативний синтаксис оператора **if-elseif**:

```
if ($a > $b):
    $result = "А більше Б";
elseif ($a == $b):
    $result = "А дорівнює Б";
else:
    $result = "А менше Б";
endif;
```

Умовний оператор **?** повертає одне з двох значень, розділених двокрапкою. Використання оператора **?** може зробити більш компактним текст програми.

Приклад 4

```
$result = ($a == $b) ? "А дорівнює Б" : "А не дорівнює Б";
```

2) Оператор розгалуження в шаблонизаторі Twig. Умовні вирази шаблонизатора Twig.

Побудова умовних виразів в шаблонизаторі Twig синтаксично подібна до мови C++

Правила логічних значень в шаблонизаторі Twig:

Значення	Булевський еквівалент
----------	-----------------------

Пуста строка	false
Число нуль	false
Пробільний рядок	true
і непустий рядок	
Пустий масив	false
null	false

Непустий	true
масив	
object	true

Приклади використання умовного оператора в шаблонизаторі Twig

```
{%if login1==false%}  
<p> Введено некоректний логін. Спробуйте ще раз!</p>  
{%endif%}
```

```
{%if a>b%}  
<p>А більше Б</p>  
{%else%}  
<p>А менше або дорівнює Б</p>  
{%endif%}
```

```
{%if a>b%}  
<p>А більше Б</p>  
{%elseif a==b%}  
<p>А дорівнює Б</p>  
{%else%}  
<p>А менше Б</p>  
{%endif%}
```

3) Приклад рішення задачі.

Умова задачі: Визначити псевдовипадкове число, перевірити чи є воно парне чи непарне, вивести число і інформацію про парність на екран.

Перший спосіб рішення - використання умовного оператора в шаблоні

Формируємо файл **lab4.php** (Рис. 1), в якому визначаємо псевдовипадкове ціле число в діапазоні 0-30 (`$num=rand(0,30);`), знаходимо остачу від цілочисленного ділення на 2 (`$div=$num%2;`), передаємо значення `$num` та `$div` в шаблон (`$tpl_vars=array('num'=>$num,'div'=>$div);`).

В шаблоні **lab4.tmpl** (Рис. 2) перевіряємо значення переданої змінної `div`, якщо воно дорівнює 0, то виводимо інформацію про парність числа `num`, в протилежному випадку виводимо дані про непарність числа `num`.

Результат роботи програми наведено на рисунку (Рис.3)

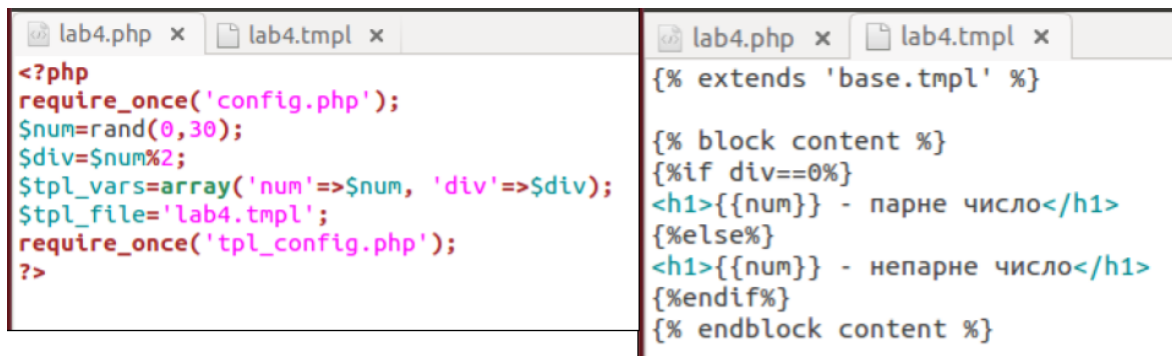


Рис. 1. Файл lab4.php

Рис. 2. Файл lab4.tpl

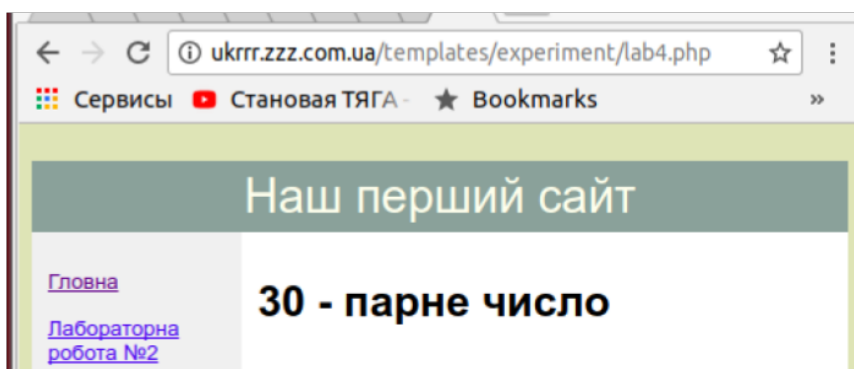


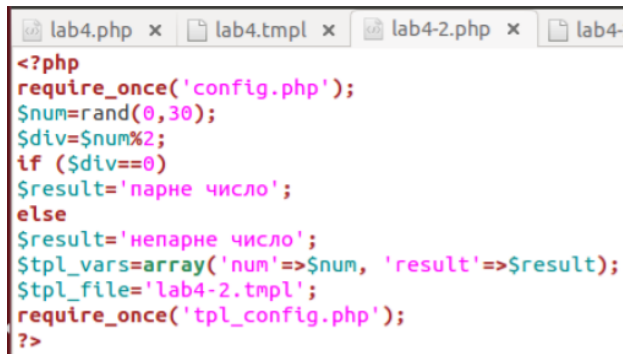
Рис. 3. Результат

Другий спосіб рішення - використання умовного оператора в файлі частини контролера (в php файлі)

Формуємо файл **lab4.php** (Рис. 4), в якому визначаємо псевдовипадкове ціле число в діапазоні 0-30 ($\$num=rand(0,30);$), знаходимо остачу від цілочисленного ділення на 2 ($\$div=\$num\%2;$), перевіряємо на рівність нулю остачу від цілочислового ділення і формуємо рядок-результат $\$result$ з використанням конструкції розгалуження, передаємо значення $\$num$ та $\$result$ в шаблон ($\$tpl_vars=array('num'=>\$num, 'result'=>\$result);$).

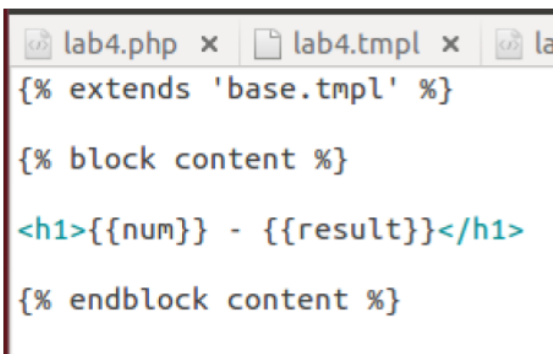
В шаблоні **lab4.tpl** (Рис. 5) виводимо значення змінних num та $result$.

Результат роботи програми наведено на рисунку (Рис.6) він є аналогічним до результату рішення даної задачі першим способом.



```
<?php
require_once('config.php');
$num=rand(0,30);
$div=$num%2;
if ($div==0)
$result='парне число';
else
$result='непарне число';
$tpl_vars=array('num'=>$num, 'result'=>$result);
$tpl_file='lab4-2.tpl';
require_once('tpl_config.php');
?>
```

Рис. 4. Файл lab4.php, варіант 2



```
{% extends 'base.tpl' %}

{% block content %}

<h1>{{num}} - {{result}}</h1>

{% endblock content %}
```

Рис. 5. Файл lab4.tpl, варіант 2

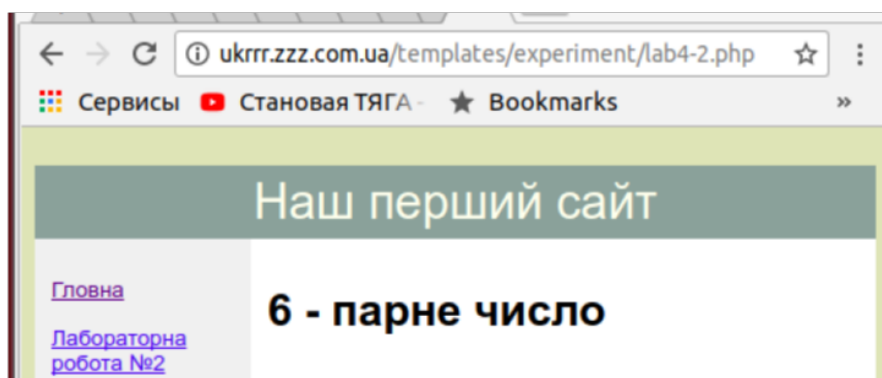


Рис. 6. Результат ідентичний до першого

4) Індивідуальні завдання.

1. Всі вхідні дані визначати конкретними значеннями змінних у файлі lab4.php (без інтерфейсу введення).

2. У шаблон lab4.tpl передати значення наступних змінних

- строкову змінну з умовою завдання
- значення всіх змінних, що визначають вхідні дані
- змінні з результатами рішення задачі.

3. У шаблоні оформити виведення переданих значень з використанням тегів і стилів.

4. Рішити дві задачі. 1 задача - ваш порядковий номер в групі, 2 задача- номер в групі +15.

5. Для одної з двох задач реалізувати 2 спосіб рішення, що передбачає використання умовної конструкції в шаблоні.

Задача1. Дано ціле число. Якщо воно є додатнім, то додати до нього 1; в іншому разі не змінювати його. Вивести умову задачі, початкові значення та результуючі.

Задача2. Дано ціле число. Якщо воно є додатнім, то додати до нього 1; в іншому випадку відняти від нього 2. Вивести умову задачі, початкові значення та результуючі.

Задача3. Дано ціле число. Якщо воно є додатнім, то додати до нього 1; якщо від'ємним, то відняти від нього 2; якщо нульовим, то замінити його на 10. Вивести умову задачі, початкові значення та результуючі.

Задача4 °. Дано три цілих числа. Знайти кількість додатніх чисел в результуючому наборі. Вивести умову задачі, початкові значення та результуючі.

Задача5. Дано три цілих числа. Знайти кількість додатніх і кількість від'ємних чисел в початковому наборі. Вивести умову задачі, початкові значення та результуючі.

Задача6. Дано два числа. Знайти більше з них. Вивести умову задачі, початкові значення та результуючі.

Задача7. Дано два числа. Вивести порядковий номер меншого з них. Вивести умову задачі, початкові значення та результуючі.

Задача8. Дано два числа. Вивести спочатку більше, а потім менше з них. Вивести умову задачі, початкові значення та результуючі.

Задача9. Дано дві змінні дійсного типу: А, В. Перерозподілити значення даних змінних так, щоб в А виявилось менше зі значень, а в В - більше. Вивести нові значення змінних А і В. Вивести умову задачі, початкові значення та результуючі.

Задача10. Дано дві змінні цілого типу: А і В. Якщо їх значення не рівні, то присвоїти кожній змінній суму цих значень, а якщо рівні, то присвоїть змінним нульові значення. Вивести нові значення змінних А і В. Вивести умову задачі, початкові значення та результуючі.

Задача11. Дано дві змінні цілого типу: А і В. Якщо їх значення не рівні, то присвоїти кожній змінній більше з цих значень, а якщо рівні, то привласнити змінним нульові значення. Вивести нові значення змінних А і В. Вивести умову задачі, початкові значення та результуючі.

Задача12. Дано три числа. Знайти найменше з них. Вивести умову задачі, початкові значення та результуючі.

Задача13. Дано три числа. Знайти середнє з них (тобто число, розташоване між найменшим і найбільшим). Вивести умову задачі, початкові значення та результуючі.

Задача14. Дано три числа. Вивести спочатку найменше, а потім найбільше з даних чисел. Вивести умову задачі, початкові значення та результуючі.

Задача15. Дано три числа. Знайти суму двох найбільших з них. Вивести умову задачі, початкові значення та результуючі.

Задача16. Дано три змінні дійсного типу: А, В, С. Якщо їх значення впорядковані за зростанням, то подвоїти їх; в іншому випадку замінити значення кожної змінної на протилежне. Вивести умову задачі, початкові значення та результуючі.

Задача17. Дано три змінні дійсного типу: А, В, С. Якщо їх значення впорядковані за зростанням або спаданням, то подвоїти їх; в іншому випадку замінити значення кожної змінної на протилежне. Вивести умову задачі, початкові значення та результуючі.

Задача18. Дано три цілих числа, одне з яких відмінно від двох інших, рівних між собою. Визначити порядковий номер числа, відмінного від остальных. Вивести умову задачі, початкові значення та результуючі.

Задача19. Дано чотири цілих числа, одне з яких відмінно від трьох інших, рівних між собою. Визначити порядковий номер числа, відмінного від інших. Вивести умову задачі, початкові значення та результуючі.

Задача20. На числовій осі розташовані три точки: А, В, С. Визначити, яка з двох останніх точок (В або С) розташована ближче до А, і вивести цю точку і її відстань від точки А. Вивести умову задачі, початкові значення та результуючі.

Задача21. Дано цілочисельні координати точки на площині. Якщо точка збігається з початком координат, то вивести 0. Якщо точка не збігається з початком координат, але лежить на осі ОХ або ОУ, то вивести відповідно 1 або 2. Якщо точка не лежить на координатних осях, то вивести 3. Вивести умову задачі, початкові значення та результуючі.

Задача22. Дано координати точки, що не лежить на координатних осях ОХ та ОУ. Визначити номер координатної чверті, в якій знаходиться дана крапка. Вивести умову задачі, початкові значення та результуючі.

Задача23. Дано цілочисельні координати трьох вершин прямокутника, сторони

якого паралельні координатним осях. Знайти координати його четвертої вершини. Вивести умову задачі, початкові значення та результуючі.

Задача24. Для даного дійсного x знайти значення наступної функції f , для якої виконуються наступні умови:

$$f(x) = 2 \cdot \sin(x), \text{ якщо } x > 0,$$

$$f(x) = 6 - x, \text{ якщо } x \leq 0.$$

Вивести умову задачі, початкові значення та результуючі.

Задача25. Для даного цілого x знайти значення наступної функції f , для якої виконуються наступні умови:

$$f(x) = 2 \cdot x, \text{ якщо } x < -2 \text{ або } x > 2,$$

$$f(x) = 3 \cdot x, \text{ в іншому випадку.}$$

Вивести умову задачі, початкові значення та результуючі.

Задача26. Для даного дійсного x знайти значення наступної функції f , для якої виконуються наступні умови:

$$f(x) = -x, \text{ якщо } x \leq 0,$$

$$f(x) = x, \text{ якщо } 0 < x < 2,$$

$$f(x) = 4, \text{ якщо } x \geq 2.$$

Вивести умову задачі, початкові значення та результуючі.

Задача27. Для даного дійсного x знайти значення наступної функції f , для якої виконуються наступні умови:

$$f(x) = 0, \text{ якщо } x < 0,$$

$$f(x) = 1, \text{ якщо } x \text{ належить } [0, 1), [2, 3), \dots,$$

$$f(x) = -1, \text{ якщо } x \text{ належить } [1, 2), [3, 4), \dots$$

Вивести умову задачі, початкові значення та результуючі.

Задача28. Дано номер року (додатне ціле число). Визначити кількість днів в цьому році, враховуючи, що звичайний рік нараховує 365 днів, а високосний - 366 днів. Високосним вважається рік, що ділиться на 4, за винятком тих років, які діляться на 100 і не діляться на 400 (наприклад, роки 300 1300 і 1900 не є високосними, а 1200 і 2000 - являють). Вивести умову задачі, початкові значення та результуючі.

Задача29. Дано ціле число. Вивести його рядок-опис виду «від'ємне парне число», «нульове число», «додатне непарне число» і т. д. Вивести умову задачі, початкові значення та результуючі.

Задача30. Дано ціле число, яке лежить в діапазоні 1-999. Вивести його строку-опис виду «парне двозначне число», «непарне тризначне число» і т.д. Вивести умову задачі, початкові значення та результуючі.

Лабораторна робота №5

Алгоритми розгалуження, що реалізуються з використанням оператора вибору в мові php.

Ціль: Рішення задач на побудову алгоритмів розгалуження. Використання оператора вибору.

Зміст

- 1) Оператор вибору у мові PHP.
 - 2) Приклад рішення задачі.
 - 3) Індивідуальні завдання.
- Хід роботи.

1) Оператор вибору у мові PHP.

Оператор вибору switch оцінює один вислів і в залежності від його значення виконує один з декількох блоків програми. Вираз в операторі switch найчастіше буває простою змінною. Наприклад:

Приклад 1

```
switch ( $a )
{
    case 1:
        $result= "А дорівнює 1";
        break;
    case 2:
        $result= "А дорівнює 2";
        break;
    case 3:
        $result= "А дорівнює 3";
        break;
    default:
        $result= "А не дорівнює ні 1, ні 2, ні 3";
}
```

Альтернативний синтаксис оператора **switch**:

```
switch ( $a ):
    case 1:
        $result= "А дорівнює 1";
        break;
    case 2:
        $result= "А дорівнює 2";
        break;
    case 3:
        $result= "А дорівнює 3";
        break;
```

default:

```
$result= "А не дорівнює ні 1, ні 2, ні 3";  
endswitch;
```

2) Приклад рішення задачі.

В прикладі рішаємо задачу: “За наведеним натуральним номером вивести назву місяця, якщо номер більше 12, то вивести повідомлення про помилку”. В файлі lab5.php ми визначаємо довільне значення натурального числа n. Формуємо масив результатів, що включає три значення: text, n, result, де text - текст задачі, n - введене число, result - отримана назва місяця (Рис.1). В файли-шаблони виводимо в варіативний блок content значення (Рис.2) і отримуємо остаточний результа (Рис. 3)

```
<?php  
require once('config.php');  
$text="За наведеним цілим номером вивести назву  
місяця, якщо номер більше 12, то вивести  
повідомлення про помилку";  
$n=5;  
switch ($n)  
{  
case 1: $result="січень";break;  
case 2: $result="лютий";break;  
case 3: $result="березень";break;  
case 4: $result="квітень";break;  
case 5: $result="травень";break;  
case 6: $result="червень";break;  
case 7: $result="липень";break;  
case 8: $result="серпень";break;  
case 9: $result="вересень";break;  
case 10: $result="жовтень";break;  
case 11: $result="листопад";break;  
case 12: $result="грудень";break;  
default: $result="Номер місяця введено не вірно";  
}  
$tpl_vars=array('text'=>$text, 'n'=>$n, 'result'=>$result);  
$tpl_file='lab5.tpl';  
require_once('tpl_config.php');  
?>
```

Рис. 1. Файл lab5.php

```
{% extends 'base.tpl' %}  
  
{% block content %}  
<h1>Лабораторна робота №5</h1>  
<h3>{{text}}</h3>  
<h2>Введено число {{n}}</h2>  
<h2>Введеному числу відповідає місяць - {{result}}</b></h2>  
{% endblock content %}
```

Рис. 2. Файл lab5.tpl

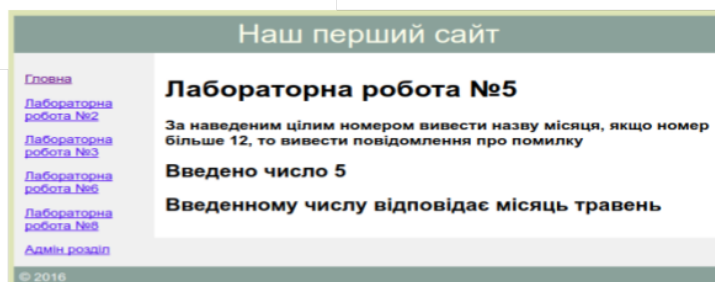


Рис. 3. Результат

3) Індивідуальні завдання

1. Всі вхідні дані визначати конкретними значеннями змінних у файлі lab5.php (без інтерфейсу введення).

2. У шаблон lab5.tpl передати значення наступних змінних

- строкову змінну з умовою завдання
- значення всіх змінних, що визначають вхідні дані
- змінні з результатами рішення задачі.

3. У шаблоні оформити виведення переданих значень з використанням тегів і стилів.

4. Рішити дві задачі. 1 задача - ваш порядковий номер в групі, 2 задача- номер в групі +10.

Задача1. Дано ціле число в діапазоні 1-7. Вивести рядок - назва дня тижня, що відповідає даному числу (1 - «понеділок», 2 - «вівто рок» і т. д.). Вивести умову задачі, початкові значення та результуючі.

Задача2. Дано ціле число K. Вивести рядок-опис оцінки, що відповідає числу K (1 - «погано», 2 - «незадовільно», 3 - «задовільно», 4 - «добре», 5 - «відмінно»). Якщо K не лежить в діапазоні 1-5, то вивести рядок «помилка». Вивести умову задачі, початкові значення та результуючі.

Задача3. Дано номер місяця - ціле число в діапазоні 1-12 (1 - січень, 2 - лютий і т. д.). Вивести назву відповідної пори року («зима», «весна», «літо», «осінь»). Вивести умову задачі, початкові значення та результуючі.

Задача4. Дано номер місяця - ціле число в діапазоні 1-12 (1 - січень, 2 - лютий і т. д.). Визначити кількість днів у цьому місяці для високосного року. Вивести умову задачі, початкові значення та результуючі.

Задача5. Арифметичні дії над числами пронумеровані в такий чином: 1 - складання, 2 - віднімання, 3 - множення, 4 - ділення. Дано номер дії N (ціле число в діапазоні 1-4) і дійсні числа A і B (B не дорівнює 0). Виконати над числами вказану дію і вивести результат. Вивести умову задачі, початкові значення та результуючі.

Задача6. Одиниці довжини пронумеровані в такий спосіб: 1 - дециметр, 2 - кілометр, 3 - метр, 4 - міліметр, 5 - сантиметр. Дано номер одиниці довжини (ціле число в діапазоні 1-5) і довжина відрізка в цих одиницях (дійсне число). Знайти довжину відрізка в метрах. Вивести умову задачі, початкові значення та результуючі.

Задача7. Одиниці маси пронумеровані в такий спосіб: 1 - кілограм, 2 - міліграм, 3 - грам, 4 - тона, 5 - центнер. Дано номер одиниці маси (ціле число в діапазоні 1-5) і маса тіла в цих одиницях (дійсне число). Знайти масу тіла в кілограмах. Вивести умову задачі, початкові значення та результуючі.

Задача8. Дано два цілих числа: D (день) і M (місяць), що визначають правильну дату невисокосного року. Вивести значення D і M для дати, що передує зазначеній. Вивести умову задачі, початкові значення та результуючі.

Задача9. Дано два цілих числа: D (день) і M (місяць), що визначають правильну дату невисокосного року. Вивести значення D і M для дати, наступної за вказаною. Вивести умову задачі, початкові значення та результуючі.

Задача10. Робот може переміщатися в чотирьох напрямках («N» - північ, «E» - захід, «S» - південь, «V» - схід) і приймати три цифрові команди: 0 - продовжувати рух, 1 - поворот наліво, -1 - поворот направо. Дан символ «N» - вихідний напрямок робота і ціле число (-1, 1) - послана йому команда. Вивести напрямок робота після виконання отриманої команди. Вивести умову задачі, початкові значення та результуючі.

Задача11. Локатор орієнтований на одну зі сторін світла («N» - північ, «E» - захід, «S» - південь, «V» - схід) і може приймати три цифрові команди повороту: 1 - поворот наліво, -1 - поворот направо, 2 - поворот на 180 °. Дан символ «N» - вихідна орієнтація локатора і цілі числа N1 і N2 - дві послані команди. Вивести орієнтацію локатора після виконання цих команд. Вивести умову задачі, початкові значення та результуючі.

Задача12. Елементи кола пронумеровані в такий спосіб: 1 - радіус R, 2 - діаметр $D = 2 \cdot R$, 3 - довжина $L = 2 \cdot \pi \cdot R$, 4 - площа кола $S = \pi \cdot R^2$. Дано номер одного з цих елементів і його значення. Вивести значення інших елементів даного кола (в тому ж порядку). Як значення π вважати рівним 3.14. Вивести умову задачі, початкові значення та результуючі.

Задача13. Елементи рівнобедреного прямокутного трикутника пронумеровані в такий спосіб: 1 - катет a, 2 - гіпотенуза $c = a^2$, 3 - висота h, опущена на гіпотенузу ($h = c / 2$), 4 - площа $S = c \cdot h / 2$. Дано номер одного з цих елементів і його значення. Вивести значення інших елементів даного трикутника (в тому ж порядку). Вивести умову задачі, початкові значення та результуючі.

Задача14. Елементи рівностороннього трикутника пронумеровані в такий чином: 1 - сторона a, 2 - радіус R1 вписаного кола ($R1 = A / 3$), 3 - радіус R2 описаного кола ($R2 = 2 \cdot R1$), 4 - площа $S = 2a^2 / 3$. Дано номер одного з цих елементів і його значення. вивести значення інших елементів даного трикутника (в тому ж порядку). Вивести умову задачі, початкові значення та результуючі.

Задача15. Мастям гральних карт присвоєно порядкові номери: 1 - піки, 2 - трефи, 3 - бубни, 4 - черви. Значенням карт, старших десятки, привласнені номери: 11 - валет, 12 - дама, 13 - король, 14 - туз. Дано

два цілих числа: N - значення карти ($6 \leq N \leq 14$) і M - масть карти ($1 \leq M \leq 4$). Вивести назву відповідної карти виду «шістка бубни», «дама черви», «туз трефи» і т. п. Вивести умову задачі, початкові значення та результуючі.

Задача16. Дано ціле число в діапазоні 20-69, що визначає вік (в роках). Вивести рядок-опис зазначеного віку, забезпечивши правильне узгодження числа з словом «рік», наприклад: 20 - «двадцять років», 32 - «тридцять два роки», 41 - «сорок один рік». Вивести умову задачі, початкові значення та результуючі.

Задача17. Дано ціле число в діапазоні 10-40, що визначає кількість навчальних завдань по деякій темі. Вивести рядок-опис зазначеної кількості завдань, забезпечивши правильне узгодження числа зі словами «Навчальне завдання», наприклад: 18 - «вісімнадцять навчальних завдань», 23 - «двадцять три навчальних завдань», 31 - «тридцять одне навчальне завдання». Вивести умову задачі, початкові значення та результуючі.

Задача18. Дано ціле число в діапазоні 100-999. Вивести рядок-опис даного числа, наприклад: 256 - "двісті п'ятдесят шість", 814 - "вісімсот чотирнадцять". Вивести умову задачі, початкові значення та результуючі.

Задача19. У східному календарі прийнятий 60-річний цикл, що складається з 12-річних підциклів, що позначаються назвами кольорів: зелений, червоний, жовтий, білий і чорний. У кожному підциклі роки носять назви тварин: щури, корови, тигра, зайця, дракона, змії, коні, вівці, мавпи, курки, собаки і свині. За номером року визначити його назву, якщо 1984 рік - початок циклу: «рік зеленого щура». Вивести умову задачі, початкові значення та результуючі.

Задача20. Дано два цілих числа: D (день) і M (місяць), що визначають правильну дату. Вивести знак Зодіаку, що відповідає цій даті: «Водолій» (20.1-18.2), «Риби» (19.2-20.3), «Овен» (21.3-19.4), «Телець» (20.4-20.5), «Близнюки» (21.5-21.6), «Рак» (22.6-22.7), «Лев» (23.7-22.8), «Діва» (23.8-22.9), «Терези» (23.9-22.10), «Скорпіон» (23.10-22.11), «Стрілець» (23.11-21.12), «Козеріг» (22.12-19.1). Вивести умову задачі, початкові значення та результуючі.

Лабораторна робота №6

Циклічні алгоритми в мові php.

Ціль: Рішення задач на побудову циклічних алгоритмів. Використання циклу **for**, **while**.

Зміст

- 1) Оператори циклу **for**, **while** у мові PHP.
 - 2) Приклад рішення задачі.
 - 3) Індивідуальні завдання.
Хід роботи.
- 1) **while** - цикл з передумовою. Розглянемо на прикладі рішення задачі: знайти суму ряду $(2*i+21)$, де i приймає значення від 1 до 100 $((2*1+21)+(2*2+21)+(2*3+21)+\dots+(2*100+21))$

```
$i=1; $s=0;
while($i<=100)
{
    $s+=2*i+21;
    $i++;
}
```

Альтернативний синтаксис оператора **while**

```
$i=1; $s=0;
while($i<=100):
    $s+=2*i+21;
    $i++;
endwhile;
```

for - цикл з параметром. Розглянемо на прикладі рішення задачі: знайти суму ряду $(2*i+21)$, де i приймає значення від 1 до 100 $((2*1+21)+(2*2+21)+(2*3+21)+\dots+(2*100+21))$

```
$s=0;
for($i=1;$i<=100;$i++)
{
    $s+=2*i+21;
}
```

Альтернативний синтаксис оператора **for**

```
$s=0;
for($i=1;$i<=100;$i++):
```

```
$s+=2*i+21;  
endfor;
```

Цикл **do...while** безумовно виконує один раз всі дії тіла циклу - це його принципово відрізняє від циклу **while**

```
$i=1; $s=0;  
do {  
$s+=2*i+21;  
$i++;}  
while($i<=100)
```

В організації циклу можуть брати участь ще два оператора: break (вихід з циклу) і continue (перехід на наступний крок).

2) Приклад рішення задачі

Умова задачі: Дано натуральне число А. Знайти кількість послідовних натуральних чисел, сума яких не більша, чим А. Вивести умову задачі, початкові значення та результуючі.

Рішення задачі представлено на рисунку (Рис.1). В файлі lab6.php визначаються змінні \$text з текстом задачі, \$a - це введенне натуральне число, \$s - це змінна для накопичення суми послідовних натуральних чисел, \$n - це чергове натуральне число. При формуванні масива результуючих значень, ми зменшуємо значення суми і кількості натуральних чисел до попереднього для виведення останнього правильного значення. В файлі lab6.tpl формується виведення результатів, що передано з частини контроллера (Рис.2). Загальний вигляд рішення задачі підпорядковано загальному дизайну (Рис.3).

```
<?php
require_once('config.php');
$text="Дано натуральне число А.
Знайти кількість послідовних натуральних чисел,
сума яких не більша, чим А. Вивести умову задачі,
початкові значення та результуючі.";
$a=345;//дане число А
$n=0;// кількість послідовних натуральних чисел
$s=0;// сума нат. чисел
while($s<=$a)
{
    $n+=1;
    $s+=$n;
}

$tpl_vars = array(
    'text' => $text,
    'a' => $a,
    's' => $s-$n,
    'n' => $n-1);

$tpl file = 'lab6.tpl';
require_once('tpl_config.php');
?>
```

Рис.1. lab6.php

```
{% extends 'base.tpl' %}

{% block content %}

<h1>Лабораторна робота №6</h1>
<h3>{{ text }}</h3>
<h3>A={{ a }}</h3>
<h3>S={{ s }}</h3>
<h3>N={{ n }}</h3>
{% endblock content %}
```

Рис.2. lab6.tpl

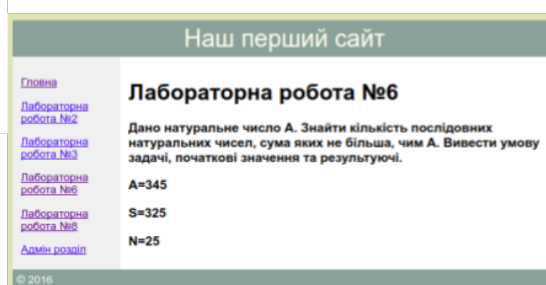


Рис.3. Результат

3) Індивідуальні завдання

1. Всі вхідні дані визначати конкретними значеннями змінних у файлі lab6.php (без інтерфейсу введення).

2. У шаблон lab6.tpl передати значення наступних змінних

- строкову змінну з умовою завдання
- значення всіх змінних, що визначають вхідні дані
- змінні з результатами рішення задачі.

3. У шаблоні оформити виведення переданих значень з використанням тегів і стилів.

4. Розв'язати дві задачі. 1 задача - ваш порядковий номер в групі, 2 задача - номер в групі +15.

Задача1. Дано ціле число $N (> 0)$. знайти суму

$N^2 + (N + 1)^2 + (N + 2)^2 + \dots + (2 \cdot N)^2$. Вивести умову задачі, початкові значення та результуючі.

Задача2. Дано ціле число $N (> 0)$. знайти добуток

$1.1 \cdot 1.2 \cdot 1.3 \cdot \dots$

$(N \text{ множників})$. Вивести умову задачі, початкові значення та результуючі.

Задача3. Дано ціле число $N (> 0)$. Знайти значення виразу

$$1.1 - 1.2 + 1.3 - \dots$$

(N доданків, знаки чергуються). Умовний оператор не використовувати. Вивести умову задачі, початкові значення та результуючі.

Задача4. Дано ціле число $N (> 0)$. Знайти квадрат даного числа, використовуючи для

його обчислення наступну формулу:

$$N^2 = 1 + 3 + 5 + \dots + (2 \cdot N - 1).$$

Після додавання до суми кожного доданка виводити поточне значення суми (в результаті будуть виведені квадрати всіх цілих чисел від 1 до N). Вивести умову задачі, початкові значення та результуючі.

Задача5 . Дано дійсне число A і ціле число $N (> 0)$. Знайти A в ступеню N :

$$A^N = A \cdot A \cdot \dots \cdot A$$

(Числа A перемножуються N разів). Вивести умову задачі, початкові значення та результуючі.

Задача6. Дано дійсне число A і ціле число $N (> 0)$. Використовуючи один цикл, вивести всі цілі ступені числа A від 1 до N . Вивести умову задачі, початкові значення та результуючі.

Задача7. Дано дійсне число A і ціле число $N (> 0)$. Використовуючи один цикл, знайти суму $1 + A + A^2 + A^3 + \dots + A^N$. Вивести умову задачі, початкові значення та результуючі.

Задача8. Дано дійсне число $X (|X| < 1)$ і ціле число $N (> 0)$. Знайти значення виразу

$$1 - A^X - X^2/2 + X^3/3 - \dots + (-1)^{(N-1)} \cdot X^N/N.$$

Отримане число є наближеним значенням функції $\ln$ в точці $1 + X$. Вивести умову задачі, початкові значення та результуючі.

Задача9. Дано дійсне число $X (|X| < 1)$ і ціле число $N (> 0)$. Знайти значення виразу

$$X - X^3/3 + X^5/5 - \dots + (-1)^N \cdot X^{(2 \cdot N + 1)} / (2 \cdot N + 1).$$

Отримане число є наближеним значенням функції $\arctg$ в точці X . Вивести умову задачі, початкові значення та результуючі.

Задача10. Дано дійсне число $X (|X| < 1)$ і ціле число $N (> 0)$. Знайти значення виразу

$$X + 1 \cdot X^3 / (2 \cdot 3) + 1 \cdot 3 \cdot X^5 / (2 \cdot 4 \cdot 5) + \dots + 1 \cdot 3 \cdot \dots \cdot (2 \cdot N - 1) \cdot X^{(2 \cdot N + 1)} / (2 \cdot 4 \cdot \dots \cdot (2 \cdot N) \cdot (2 \cdot N + 1)).$$

Отримане число є наближеним значенням функції $\arcsin$ в точці X . Вивести умову задачі, початкові значення та результуючі.

Задача11. Дано дійсне число X ($|X| < 1$) і ціле число N (> 0). Знайти значення виразу

$1 + X/2 - 1 \cdot X^2/(2 \cdot 4) + 1 \cdot 3 \cdot X^3/(2 \cdot 4 \cdot 6) - \dots + (-1)^{N-1} \cdot 1 \cdot 3 \cdot \dots \cdot (2 \cdot N-3) \cdot X^N/(2 \cdot 4 \cdot \dots \cdot (2 \cdot N))$. Отримане число є наближеним значенням функції корінь квадратний з $(1 + X)$. Вивести умову задачі, початкові значення та результуючі.

Задача12. Дано ціле число N (> 1) і дві дійсні точки на числовій осі:

A, B ($A < B$). Відрізок $[A, B]$ розбитий на N рівних відрізків. Вивести N - довжину кожного відрізка, а також набір точок

$A, A + H, A + 2 \cdot H, A + 3 \cdot H, \dots, B$, що утворює розбиття відрізка $[A, B]$. Вивести умову задачі, початкові значення та результуючі.

Задача13. Дано ціле число N (> 1) і дві дійсні точки на числовій осі:

A, B ($A < B$). Відрізок $[A, B]$ розбитий на N рівних відрізків. Вивести N - довжину кожного відрізка, а також значення функції $F(X) = 1 - \sin(X)$ в точках, що розбивають відрізок $[A, B]$: $F(A), F(A + H), F(A + 2 \cdot H), \dots, F(B)$. Вивести умову задачі, початкові значення та результуючі.

Задача14. Дано ціле число N (> 0). Послідовність дійсних чисел AK

визначається наступним чином:

$A_0 = 2, AK = 2 + 1 / AK-1, K = 1, 2, \dots$

Вивести елементи $A_1, A_2, \dots, A_N$. Вивести умову задачі, початкові значення та результуючі.

Задача15. Дано ціле число N (> 0). Послідовність дійсних чисел AK

визначається наступним чином:

$A_0 = 1, AK = (AK-1 + 1) / K, K = 1, 2, \dots$

Вивести елементи $A_1, A_2, \dots, A_N$. Вивести умову задачі, початкові значення та результуючі.

Цикл While

Задача1. Дано дійсне число A . Перший член арифметичної прогресії X та різниця D . Знайти номер і значення найбільшого члена цієї прогресії, що не перебільшує число A . Вивести умову задачі, початкові значення та результуючі.

Задача2. Дано дійсне число A . Перший член арифметичної прогресії X та різниця D . Знайти кількість членів цієї прогресії, сума яких не перебільшує число A . Вивести умову задачі, початкові значення та результуючі.

Задача3. Дано дійсне число A . Перший член геометричної прогресії X та знаменник D . Знайти кількість членів цієї прогресії, сума яких не перебільшує число A . Вивести умову задачі, початкові значення та результуючі.

Задача4. Дано дійсне число A . Перший член геометричної прогресії X та знаменник D . Знайти номер і значення найбільшого члена цієї прогресії, що не перебільшує число A . Вивести умову задачі, початкові значення та результуючі.

Задача5. Дано дійсне число A . Знайти кількість чисел Фібоначчі, сума яких не перебільшує число A . Вивести умову задачі, початкові значення та результуючі.

Задача6. Дано дійсне число A . Знайти номер і значення найбільшого числа Фібоначчі, що не перебільшує число A . Вивести умову задачі, початкові значення та результуючі.

Задача7. Дано дійсне число A . Знайти кількість підряд ідучих простих чисел, сума яких не перебільшує число A . Вивести умову задачі, початкові значення та результуючі.

Задача8. Дано дійсне число A . Знайти номер і значення найбільшого простого числа, що не перебільшує число A . Вивести умову задачі, початкові значення та результуючі.

Задача9. Дано дійсне число A . Знайти кількість підряд ідучих натуральних чисел, що кратні 3, сума яких не перебільшує число A . Вивести умову задачі, початкові значення та результуючі.

Задача10. Дано дійсне число A . Знайти номер і значення найбільшого натурального числа, що кратне 3, що не перебільшує число A . Вивести умову задачі, початкові значення та результуючі.

Задача11. Дано дійсне число A . Знайти кількість підряд ідучих непарних натуральних чисел, що кратні 5, сума яких не перебільшує число A . Вивести умову задачі, початкові значення та результуючі.

Задача12. Дано дійсне число A . Знайти номер і значення найбільшого непарного натурального числа, що кратне 5, що не перебільшує число A . Вивести умову задачі, початкові значення та результуючі.

Лабораторна робота №7

Масиви в мові php. Спеціальні цикли по елементах масиву в мові php та в шаблонизаторі Twig.

Ціль: Рішення задач на опрацювання масивів даних. Виведення масивів даних в шаблоні засобами шаблонизатору Twig.

Зміст

- 1) Масиви в мові PHP.
- 2) foreach - цикл за елементами масиву у мові PHP.
- 3) Цикл за елементами масиву в шаблонизаторі Twig.
- 4) Приклад рішення задачі.
- 5) Індивідуальні завдання.

1) Масиви в мові PHP.

Основне призначення масивів в PHP - організація груп пов'язаних значень. Кожен елемент масиву має індекс (ключ) і значення. Індекс елемента масиву вказується в квадратних дужках після імені масиву. Для того, щоб звернутися до п'ятого елемента масиву `$array1`, треба написати:

```
$array1[5]
```

Пам'ятайте, що за замовчуванням масив починається не з першого елемента, а з нульового.

Індекс може бути як числом, так і текстовим рядком. Масив із строковими індексами називають асоціативним, а самі індекси - іменами елементів. Наприклад, ціни на товари зберігаються в асоціативному масиві `$prices`, індексами якого є найменування товарів.

Щоб отримати значення ціни на конкретний товар, треба написати:

```
$prices["картопля"]
```

Значення елемента масиву може мати будь-який тип. Можлива організація багатовимірних масивів, так як елемент масиву може в свою чергу бути масивом. Приклади звернень до елементів багатовимірних масивів:

```
$array3 [0][4][1]  
$prices["овочі"]["помідор"]  
$list1[1]["прізвище"]
```

Створення масиву

Масив можна створити за допомогою функції `array()`, параметри якої і складають масив. Параметри можуть задаватися парами "ключ => значення". Якщо при створенні масиву ключ не вказується, то індекс визначається положенням елемента в масиві (починаючи з 0). наприклад:

```
$mass1 = array (174, 181, 166); // Масив з індексацією, що починається з нуля
$price = array ( "помідори" => 15, "огірки" => 12); // Асоціативний масив
$list1 = array (
    "Іванов" => array ( "ріст" => 174, "вага" => 68),
    "Петров" => array ( "ріст" => 181, "вага" => 90),
    "Сидоров" => array ( "ріст" => 166, "вага" => 73)); // Двовірний масив
```

Перебір елементів масиву в циклі, `foreach` - цикл за елементами масиву у мові PHP

Для ітераційного перегляду вмісту масиву служить спеціальний цикл по елементах масиву `foreach`. З його допомогою можна переглянути і простий (проіндексований числами) масив, і асоціативний, і багатовимірний.

```
$fruit = array ( "яблуко", "груша", "слива", "персик", "груша");
foreach          ($fruit          as          $value)
{
    $str .= $value."&nbsp;";
}
```

Фрагмент наведеного прикладу формує строку **"яблуко груша слива персик груша"**. Між елементами масиву в строчці встановлюється пробільний символ (` `).

Другий варіант синтаксису циклу по елементам масиву `foreach` включає можливість виводу ключа кожного елемента.

```
$fruit = array ( "яблуко", "груша", "слива", "персик", "груша");
foreach      (      $fruit      as      $key      =>      $value      )
{
    $str .= $key . ">". $value . "&nbsp;";
}
```

Фрагмент наведеного прикладу формує строку **"0=>яблуко 1=>груша 2=>слива 3=>персик 4=>груша"**. Між елементами масиву в строчці встановлюється пробільний символ (` `).

Переглянемо як аналогічний приклад буде реалізовано для асоціативного масиву

```
$price = array ( "яблуко"=>"30грн", "груша"=>"40грн", "слива"=>"50грн",
"персик"=>"40грн", "груша"=>"35грн");
foreach ($fruit as $value)
{
$str .= $value."&nbsp;";
}
```

Фрагмент наведеного прикладу формує строку **"30грн 40грн 50грн 40грн 35грн"**. Між елементами масиву в строчці встановлюється пробільний символ ().

Другий варіант синтаксису циклу по елементам масиву foreach включає можливість виводу ключа кожного елемента.

Переглянемо як аналогічний приклад буде реалізовано для асоціативного масиву

```
$price = array ( "яблуко"=>"30грн", "груша"=>"40грн", "слива"=>"50грн",
"персик"=>"40грн", "груша"=>"35грн");
foreach ($fruit as $key => $value)
{
$str .= $key ."=>". $value."&nbsp;";
}
```

Фрагмент наведеного прикладу формує строку **"яблуко=>30грн груша=>40грн слива=>50грн персик=>40грн груша=>35грн"**. Між ключем і значенням кожного елемента масиву встановлено символ =>, а між елементами масиву в строчці встановлюється пробільний символ ().

3) Цикл за елементами масиву в шаблонизаторі Twig.

Розглянемо ситуацію, коли в шаблон з частини контроллера передано два одномірних масива \$fruit, \$price

```
$fruit = array ( "яблуко", "груша", "слива", "персик", "груша");
$price = array ( "яблуко"=>"30грн", "груша"=>"40грн", "слива"=>"50грн",
"персик"=>"40грн", "груша"=>"35грн");
$tpl_vars= array('fruit'=>$fruit, 'price'=>$price);
```

В шаблоні данні масиви доступні через імена fruit, price.

Розглянемо синтаксис циклу по елементах масиву, що дозволить вивести на екран його елементи:

```
<ul>
{% for key, value in fruit %}
<li>{{key}}=>{{value}}</li>
{% endfor %}
</ul>
<ul>
{% for key, value in price %}
<li>{{key}}=>{{value}}</li>
```

```
{% endfor %}  
</ul>
```

Коли значення ключа не потрібно виводити, то ми отримуємо наступний скорочений синтаксис виведення результатів

```
<ul>  
  {% for value in fruit %}  
  <li>{{value}}</li>  
  {% endfor %}  
</ul>  
  
<ul>  
  {% for value in price %}  
  <li>{{value}}</li>  
  {% endfor %}  
</ul>
```

Дуже важливим для побудови Web орієнтованих інформаційних систем стає вміння виводити на екран в шаблоні значення з двовимірною асоціативною масивою тому, що результати SELECT запитів формуються в частині контролера у такому вигляді. Розглянемо приклад.

id	name	description	data
1	Новина №1	Детальна інформація №1	2018-09-22
2	Новина №2	Детальна інформація №2	2018-09-11
3	Новина №3	Детальна інформація №3	2012-11-19
4	Новина №4	Детальна інформація №4	2012-11-13
5	Новина №5	Детальна інформація №5	2012-11-01

Рис. 1. Приклад таблиці бази даних

Результат виборки всіх значень з таблиці (Рис.1) представляється в частині контролера у вигляді наступного асоціативного масиву:

```
$result=array(  
array("id"=>1,"name"=>"Новина №1","description"=>"Детальна інформація №1","data"=>"2018-09-22"),  
array("id"=>2,"name"=>"Новина №2","description"=>"Детальна інформація №2","data"=>"2018-09-11"),  
array("id"=>3,"name"=>"Новина №3","description"=>"Детальна інформація №3","data"=>"2012-11-19"),
```

```
array("id"=>4,"name"=>"Новина №4","description"=>"Детальна інформація №4","data"=>"2012-11-13"),
array("id"=>5,"name"=>"Новина №5","description"=>"Детальна інформація №5","data"=>"2012-11-01")
);
$tpl_vars=array("result"=>$result);
```

Розглянемо принцип виведення значень з масиву result у форматі таблиці в шаблоні:

```
<table border="1">
{% for element in result %}
<tr>
{% for value in element %}
<td>{{value}}</td>
{% endfor %}
</tr>
{% endfor %}
</table>
```

3) Приклад рішення задачі.

Задача полягає в визначенні двох масивів (\$menu-одновимірний, \$_user-двовимірний) в файлі lab7.php частини контролера та передачі даних масивів в частину дизайну (Рис. 2). Виведення одновимірного масиву зроблено у вигляді нумерованого списку, а двовимірного масиву у вигляді таблиці (Рис. 3). В результаті отримано наступну сторінку результуючого сайту (Рис. 4)

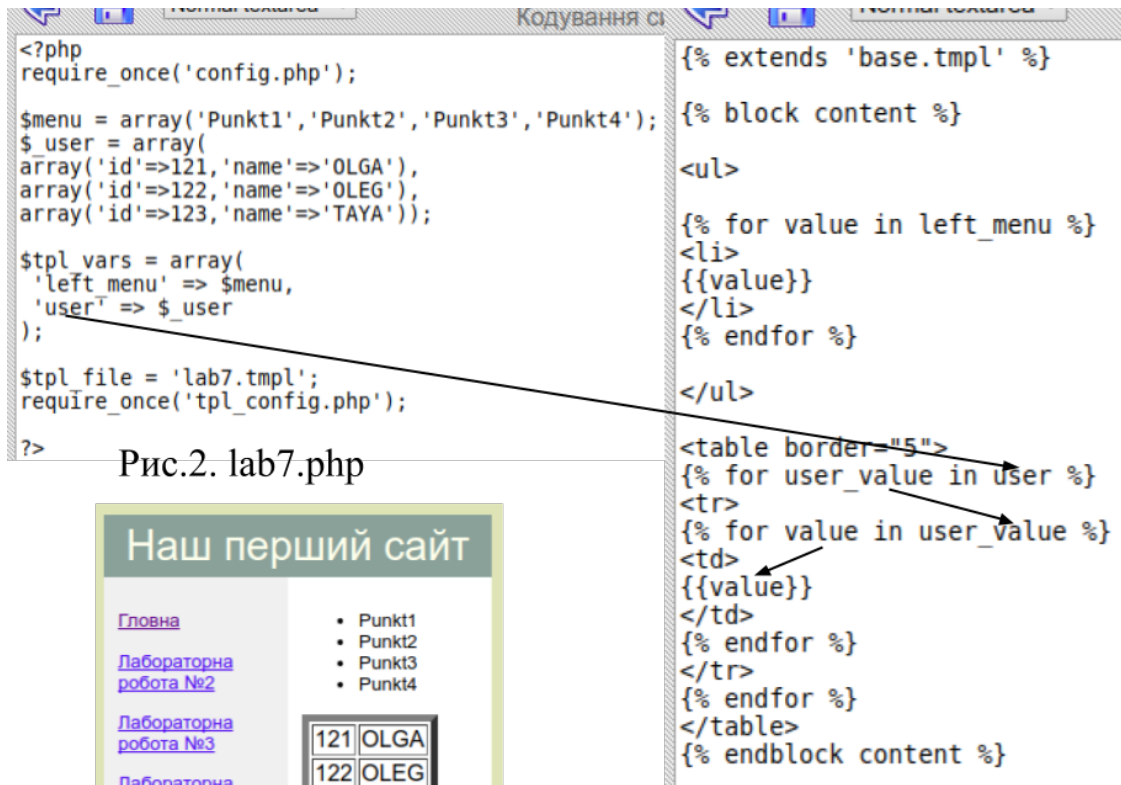


Рис.2. lab7.php



Рис.4. Результат

Рис.3. lab7.tpl

3) Індивідуальні завдання

1. Всі вхідні дані визначати конкретними значеннями змінних у файлі lab7.php (без інтерфейсу введення).

2. У шаблон lab7.tpl передати значення наступних змінних

- строкову змінну з умовою завдання
- значення всіх змінних, що визначають вхідні дані
- змінні з результатами рішення задачі.

3. У шаблоні оформити виведення переданих значень з використанням тегів і стилів.

4. Рішити одну задачу. 1 задача - ваш порядковий номер в групі.

Задача1. Дано ціле число $N (> 0)$. Сформувати і вивести цілочисельний масив розміру N , що містить N перших додатніх непарних чисел: 1, 3, 5, Вивести умову задачі, початкові значення та результуючі.

Задача2. Дано ціле число $N (> 0)$. Сформувати і вивести цілочисельний масив розміру N , що містить ступені двійки від першої до N -ої: 2, 4, 8, 16, Вивести умову задачі, початкові значення та результуючі.

Задача3. Дано ціле число $N (> 1)$, а також перший член A і різниця D арифметичної прогресії. Сформувати і вивести масив розміру N , що складається з N перших членів даної прогресії:

$A, A + D, A + 2 \cdot D, A + 3 \cdot D, \dots$. Вивести умову задачі, початкові значення та результуючі.

Задача4. Дано ціле число $N (> 1)$, а також перший член A і знаменник D геометричної прогресії. Сформувати і вивести масив розміру N , що складається з N перших членів даної прогресії:

$A, A \cdot D, A \cdot D^2, A \cdot D^3, \dots$. Вивести умову задачі, початкові значення та результуючі.

Задача5. Дано ціле число $N (> 2)$. Сформувати і вивести цілочисельний масив розміру N , що містить N перших елементів послідовності чисел Фібоначчі FK : $F1 = 1, F2 = 1, FK = FK-2 + FK-1, K = 3, 4, \dots$. Вивести умову задачі, початкові значення та результуючі.

Задача6. Дано цілі числа $N (> 2)$, A і B . Сформувати і вивести цілочислений масив розміру N , перший елемент якого дорівнює A , другий дорівнює B , а кожний наступний елемент дорівнює сумі всіх попередніх. Вивести умову задачі, початкові значення та результуючі.

Задача7. Дан масив розміру N . Вивести його елементи в зворотному порядку. Вивести умову задачі, початкові значення та результуючі.

Задача8. Дан цілочисельний масив розміру N . Вивести всі непарні числа, що містяться в даному масиві в порядку зростання їх індексів, а також їх кількість K . Вивести умову задачі, початкові значення та результуючі.

Задача9. Дан цілочисельний масив розміру N . Вивести всі парні числа, що містяться в даному масиві в порядку спадання їх індексів, а також їх кількість K . Вивести умову задачі, початкові значення та результуючі.

Задача10. Дан цілочисельний масив розміру N . Вивести спочатку всі парні числа в порядку зростання їх індексів, а потім - всі непарні числа в порядку спадання їх індексів. Вивести умову задачі, початкові значення та результуючі.

Задача11. Дан масив A розміру N і ціле число $K (1 \leq K \leq N)$. Вивести елементи масиву з порядковими номерами, кратними K : $A_K, A_{2 \cdot K}, A_{3 \cdot K}, \dots$. Умовний оператор не використовувати. Вивести умову задачі, початкові значення та результуючі.

Задача12. Дано масив А розміру N (N - парне число). Вивести його елементи з парними номерами в порядку зростання номерів: A2, A4, A6, ..., AN. Умовний оператор не використовувати. Вивести умову задачі, початкові значення та результуючі.

Задача13. Дан масив А розміру N (N - непарне число). Вивести його елементи з непарними номерами в порядку спадання номерів: AN, AN-2, AN-4, ..., A1. Умовний оператор не використовувати. Вивести умову задачі, початкові значення та результуючі.

Задача14. Дан масив А розміру N. Вивести спочатку його елементи з парними номерами (в порядку зростання номерів), а потім - елементи з непарними номерами (також в порядку зростання номерів):

A2, A4, A6, ..., A1, A3, A5,

Умовний оператор не використовувати. Вивести умову задачі, початкові значення та результуючі.

Задача15. Дан масив А розміру N. Вивести спочатку його елементи з непарними номерами в порядку зростання номерів, а потім - елементи з парними номерами в порядку спадання номерів.

A1, A3, A5, ..., A6, A4, A2.

Умовний оператор не використовувати. Вивести умову задачі, початкові значення та результуючі.

Задача16. Дан масив А розміру N. Вивести його елементи в наступному порядку:

A1, AN, A2, AN-1, A3, AN-2, Вивести умову задачі, початкові значення та результуючі.

Задача17. Дан масив А розміру N. Вивести його елементи в наступному порядку:

A1, A2, AN, AN-1, A3, A4, AN-2, AN-3, Вивести умову задачі, початкові значення та результуючі.

Лабораторна робота №8

Створення графічного інтерфейсу для введення даних. Методи опрацювання форм в мові PHP.

Ціль: Вивчення правил структурування і опрацювання графічного інтерфейсу введення даних. Стани запуску PHP файла.

Зміст

- 1) Користувацькі функції
- 2) Правила опрацювання даних форми. Асоціативні масиви \$_GET, \$_POST
- 3) Приклад рішення задачі.
- 4) Стани, з яких запускаються файли частини контролеру (php файли) і їх критерії
- 5) Індивідуальні завдання.

Хід роботи:

1) Користувацькі функції

Для визначення користувальницької функції використовується ключове слово function.

Приклад 1

```
function f()  
{  
$x=5;  
return $x;  
}
```

f - це ім'я функції. Визначається як ідентифікатор.

Ідентифікатором називається послідовність літер, цифр і символів підкреслення (_). Ідентифікатори виступають в якості імен змінних, функцій, властивостей об'єкта, і т. д. При виборі ідентифікатора необхідно враховувати наступні правила:

- Ідентифікатори не можуть збігатися ні з одним з ключових слів PHP, вони мають спеціальне значення для інтерпретатора.
- Ідентифікатори можуть складатися з одного і більше символів.
- Ідентифікатори повинні починатися з букви або символу підкреслення. Далі можуть слідувати будь-які літери, цифри і символи підкреслення в будь-якій кількості.

Після імені функції в дужках визначаються формальні параметри. Цей список може бути пустим.

Викликається функція в залежності від того, які дії вона виконує по-різному:

- 1) Якщо функція повертає одне значення, то виклик передбачає використання цього значення

Приклад 2

```
function f()
{
$x=5;
return $x;
}
$y=f(); //В $y покладено значення 5
```

Функцію з Прикладу 1 можна реалізувати без використання оператора return. Тоді передача результатів функції відбувається через формальний параметр-посилання.

Приклад 3

```
function f(&$x)
{
$x=5;
}
f($y); //В $y покладено значення 5, виклик функції - окремий оператор
```

В тіло функції можна передавати значення з основної програми через фактичні і формальні параметри.

Приклад 4

```
function f($a,$b,&$x)//($a,$b,&$x - формальні параметри)
{
$x=$a+$b;
}
f(5,7,$y); //В $y покладено значення 12. (5,7,$y - фактичні параметри)
$c=54; $d=48;
f($c,$d,$y); //В $y покладено значення 102($c,$d,$y - фактичні параметри)
```

Глобальні змінні

Щоб змінна, створена поза функцією, була доступна всередині функції, треба оголосити її глобальною за допомогою інструкції global.

Приклад 5

```
global $y;
function f($a,$b)
{
$y=$a+$b;
}
f(5,7); //В $y покладено значення 12, це значення доступне тут
$c=54; $d=48;
f($c,$d); //В $y покладено значення 102, це значення доступне тут
```

2) Правила опрацювання даних форми. Асоціативні масиви \$_GET, \$_POST

Суперглобальні змінні - суперглобальні змінні - це вбудовані змінні, які завжди доступні у всіх областях видимості

Деякі зумовлені змінні в PHP є "суперглобального", що означає, що вони доступні в будь-якому місці скрипта. Немає необхідності використовувати синтаксис `global $ variable`; для доступу до них у функціях і методах.

Суперглобальними змінними є:

```
$GLOBALS
$_SERVER
$_GET
$_POST
$_FILES
$_COOKIE
$_SESSION
$_REQUEST
$_ENV
```

`$_GET` - асоціативний масив змінних, що передаються скрипту через параметри URL (відомі також як рядок запиту).

Приклад 1

URL/?t=1&b=2

В файлі, що відкривається з URL (наприклад lab8.php) доступні

```
$t=$_GET['t'];//має значення 1
```

```
$b=$_GET['b'];//має значення 2
```

Якщо форма гіпертекстового документа пересилає данні методом `get`, то програма-обробник їх отримує через параметри URL адреси, що зчитуються в масиві `$_GET`

```
<form method="get" action="lab8.php">
<input type="text" name="t" value="1">
<input type="text" name="b" value="2">
</form>
```

Тоді в файлі lab8.php доступні

```
$t=$_GET['t'];//має значення 1
```

```
$b=$_GET['b'];//має значення 2
```

`$_POST` - асоціативний масив змінних, що передаються скрипту через стандартний вхідний потік.

Приклад 2

Якщо форма гіпертекстового документа пересилає данні методом `post`, то програма-обробник їх отримує через стандартний вхідний потік, з якого зчитуються дані в масиві `$_POST`

```
<form method="post" action="lab8.php">
<input type="text" name="t" value="1">
<input type="text" name="b" value="2">
</form>
```

Тоді в файлі lab8.php доступні

```
$t=$_POST['t'];//має значення 1  
$b=$_POST['b'];//має значення 2
```

3) Приклад рішення задачі.

Розглянемо приклад рішення задачі: Дано радіус кола. Обчислити його довжину. Оформити введення і виведення даних у форму.

Модель №1 рішення даної задачі

1. Створюємо файл lab8.php, який відображає пусту форму для введення початкових даних (Рис.1). Дана форма буде оброблятися файлом lab8_act.php

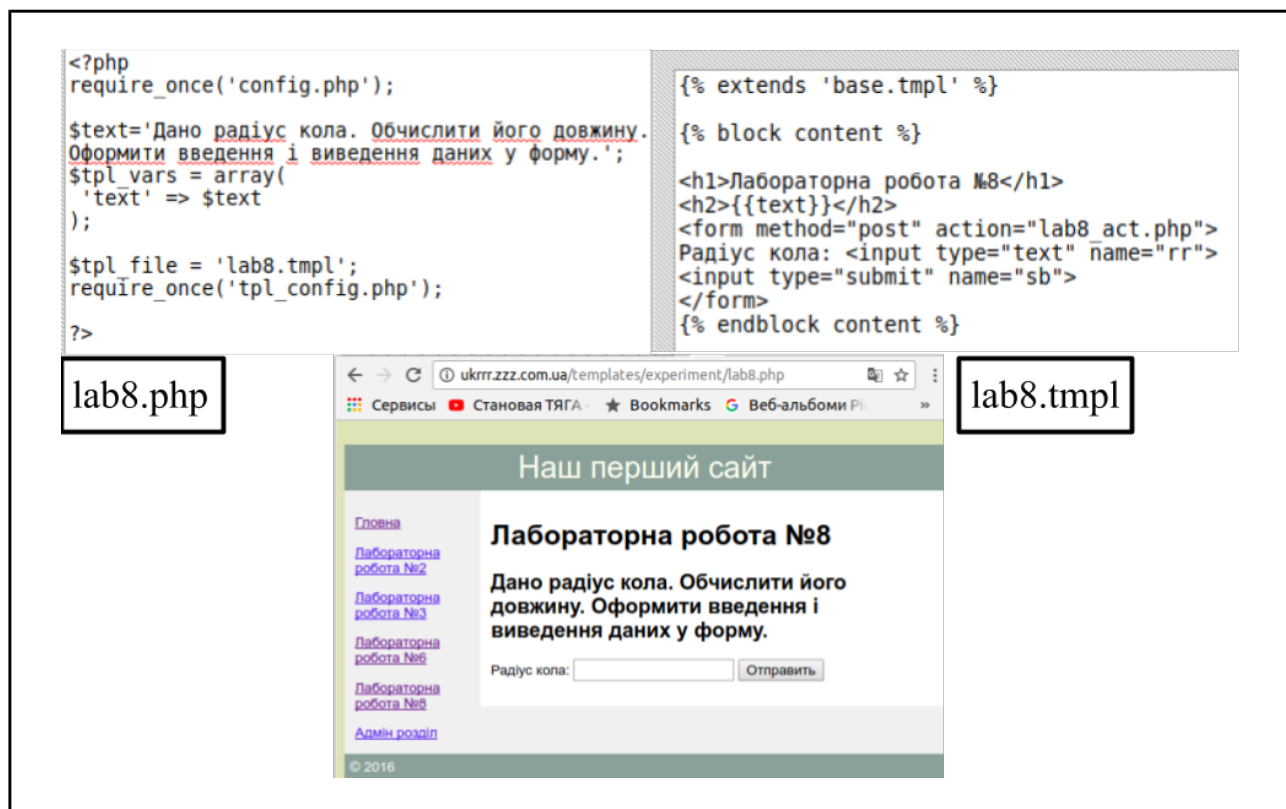


Рис.1. Створюємо форму для введення початкових даних

2. В файлі lab8_act.php опрацьовуємо дані, що введено і виводимо результати на екран (Рис.2).

Модель №1 рішення даної задачі зрозуміла, але доволі складна за реалізацією. Вона вимагає створення 2 файлів частини контролеру та 2 файлів частини дизайну.



Рис. 2. Формування файлів lab8_act.php, що опрацьовує вхідні дані форми, та lab8_act.tpl, що виводить форму з результатами

4) Стани, з яких запускаються файли частини контроллера (php файли) і їх критерії

На практиці існує поняття станів запуску файлів частини контроллера. Так перший стан - це стан, коли ніяка інформація з форми не приходила. Універсальним критерієм такого стану може бути невизначеність змінної sb з асоціативного масиву \$_POST (isset(\$_POST['sb']) вертає хибу). В цьому стані ми повинні відкрити форму для введення початкових даних. Другий стан - це стан, який характеризується наявністю стартових даних. У цьому стані ми рішаємо задачу і готуємо список результатів для відображення у формі. Узагальнюючи пояснення сформуємо Модель №2 рішення задачі:

1. Створимо файл частини контроллера, що формує список змінних для двох станів lab8.php (Рис. 2).
2. Створимо у відповідність до частини контроллера файл частини дизайну lab8.tpl (Рис.2).

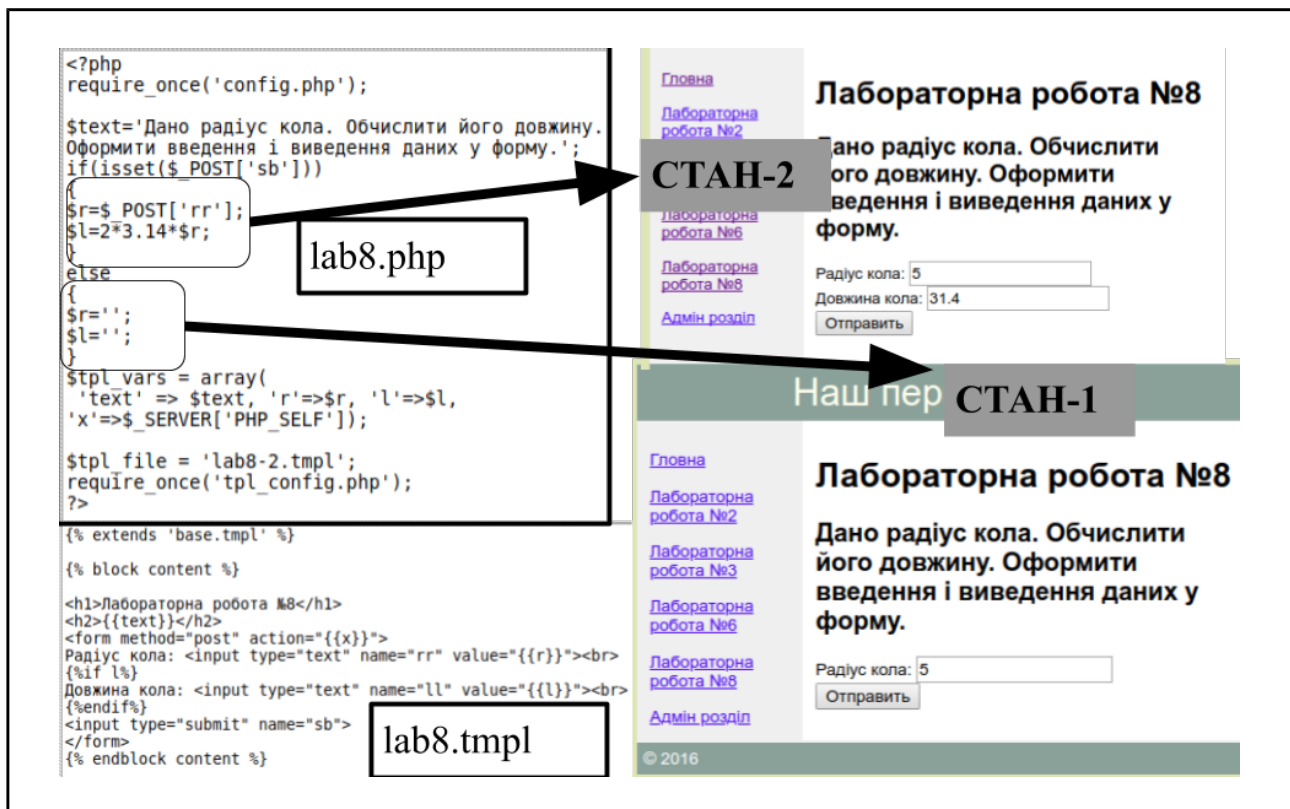


Рис.2. Рішення задачі з використанням станів

Розглядаючи більш детально файл lab8.php (Рис.2), ми бачимо використання `$_SERVER['PHP_SELF']`. Ця глобальна змінна грамотно передає керування форми в поточний файл частини контролера (в шаблоні використано як значення атрибута action тега form). Рішення задачі з організацією інтерфейсу введення та виведення інформації має охоплювати два стани php файла частини контролеру.

5) Індивідуальні завдання

1. Підготувати форми для введення та виведення початкових даних та результатів для індивідуальних завдань лабораторних робіт №3-№7
2. Підготувати для кожної задачі файл частини контролера, що працює у двох станах (до передачі даних з форми та після)
3. Розробити для кожного файла частини контролера відповідний шаблон
4. Для одної з задач структурувати код частини контролера з використанням двох функцій. Перша функція викликається для опрацювання стану-1, а друга для опрацювання стану-2 (Рис.2). Реалізувати різні способи передачі значень з функції в основну програму (1-через формальні параметри-посилання, 2-через список значень, що вертається оператором return, 3-через визначення глобальних змінних)
4. Підключити посилання на результати до головного меню сайта-звіта.

Лабораторна робота №9

Об'єктно-орієнтовані основи мови php.

Ціль: Опанувати базові підходи до побудови класів в мові PHP. Правила визначення властивостей та методів. Визначення екземплярів класів (об'єктів). Виклик властивостей та методів об'єктів.

Зміст

- 1) Об'єктно-орієнтовані основи мови php.
- 2) Приклад рішення задачі.
- 3) Індивідуальні завдання.

Хід роботи.

1) Об'єктно-орієнтоване програмування (ООП) на PHP

Об'єкт (екземпляр класу) має унікальну ідентифікацію (ім'я), для нього в відповідному класі визначається набір спеціальних змінних - властивостей і спеціальних функцій - методів. Те, що в процедурному програмуванні називається змінною - в ООП називається властивість. Те, що в процедурному програмуванні називається функцією - в ООП називається методом класу. Створені на основі класу об'єкти називаються екземплярами класу або просто об'єкти.

2) Приклад рішення задачі.

Розглянемо прості правила побудови власного класу на прикладі класу Person.

Для об'єктів даного класу має визначатись дві властивості:

name (повне ім'я персони)

year (рік народження персони)

Для об'єктів даного класу має визначатись один метод:

PersonToString (повертає строкову версію об'єкта)

Приклад 1.

```
class Person
{
public $name; // - це властивість
public $year; // - це властивість
function __construct( $name="Хтось", $year=2000 )
{
    $this->name = $name;
    $this->year = $year;
}
function PersonToString() // - це метод класу
{
    return $this->name.' - рік народження:'.$this->year;
}
}
$petrenko = new Person('Петренко Петро Петрович',1990);
$soltes = new Person('Солтис Степан Степанович',1960);
```

```
$tpl_vars = array(
'petrenko' => $petrenko->PersonToString(),
'soltes' => $soltes->PersonToString(),
'name1' => $petrenko->name,
'year1' => $petrenko->year
);
```

Розберемо код прикладу:

```
class Person {} //визначення імені класу
```

```
public $name; // - це властивість
public $year; // - це властивість
function __construct( $name="Хтось", $year=2000 )
{
    $this->name = $name;
    $this->year = $year;
}
```

В даному фрагменті визначено властивості name та year. Стандартна конструкція function__construct дає можливість визначити список формальних параметрів класу, які замінюються фактичними при визначенні екземпляру класу (об'єкту)

```
function PersonToString() // - це метод класу
{
    return $this->name.' - рік народження:'.$this->year;
}
```

Визначення методу PersonToString()

```
$petrenko = new Person('Петренко Петро Петрович',1990);
$soltes = new Person('Солтис Степан Степанович',1960);
Визначення об'єктів класу Person
```

```
$tpl_vars = array(
'petrenko' => $petrenko->PersonToString(),
'soltes' => $soltes->PersonToString(),
'name1' => $petrenko->name,
'year1' => $petrenko->year
);
```

Передавання в шаблон результатів виклику методів та властивостей об'єктів класу Person

Звернення з методу до властивостей тільки через службове слово `$this`: `$this->name`; (Зверніть увагу на відсутність знака долара перед `name`)
Звернення всередині методу до іншого методу теж через `$this`: `$this->foo()`; Для доступу до властивостей і методів об'єкта служить оператор `"->"`: `$this->name`; (Зверніть увагу на відсутність знака долара перед `name`)

Звернення всередині методу до іншого методу теж через `$this`: `$this->foo()`;
Об'єкт створюється за допомогою оператора `new` на підставі шаблону, званого класом. Клас визначається ключовим словом `class`.

Обробка винятків в ООП

Для обробки некритичних помилок використовуються виключення (`Exception`).
Це важливо саме при роботі з базою даних.

Приклад 2. Використання оператора `try-catch`

```
try {  
    $a = 1;  
    $b = 0;  
    if ($b == 0)  
        throw new Exception ( "поділ на нуль!");  
    $c = $a / $b;  
} catch (Exception $e) {  
    echo $e->getMessage ();  
    echo $e->getLine ();  
}
```

`Exception` - вбудований клас. Якщо потрапили в `throw`, то код нижче не виконується і здійснюється перехід до блоку `catch`.

Блок `try-catch` використовується як в процедурному, так і в ООП програмуванні.
Він використовується для вилову помилок - великий блок `try` з безліччю `throw` і все виловлюється в одному місці - блоці `catch`.

3) Індивідуальні завдання.

- 1) Створити в файлі частини контролеру `lab9.php` клас згідно з вашим індивідуальним завданням (вибрати згідно з вашим порядковим в журналі групи). Визначити два об'єкта створеного класу.
- 2) Передати в шаблон результат виклику методу до визначених об'єктів, та результат виклику властивостей до об'єктів.
- 3) Підключити результат до загального сайту.

Індивідуальні задачі

1. Створити клас `Fish`.

Для об'єктів даного класу має визначатись дві властивості:

`name` (повне ім'я риби, її вид)

`location` (регіони існування)

Для об'єктів даного класу має визначатись один метод:

`FishToString` (повертає строкову версію об'єкта)

2. Створити клас `Animal`.
Для об'єктів даного класу має визначатись дві властивості:
`vud` (повне ім'я тварини, її вид)
`clas` (клас тварини(наприклад-ссавці))
Для об'єктів даного класу має визначатись один метод:
`AnimalToString` (повертає строкову версію об'єкта)
3. Створити клас `Drib` (для визначення об'єктів типу звичайний дріб).
Для об'єктів даного класу має визначатись дві властивості:
`ch` (чисельник)
`zn` (знаменник)
Для об'єктів даного класу має визначатись один метод:
`DribToString` (повертає строкову версію об'єкта)
4. Створити клас `Complex` (для визначення об'єктів типу комплексні числа).
Для об'єктів даного класу має визначатись дві властивості:
`re` (дійсна частина)
`im` (уявна частина)
Для об'єктів даного класу має визначатись один метод:
`ComplexToString` (повертає строкову версію об'єкта)
5. Створити клас `Tovar`.
Для об'єктів даного класу має визначатись дві властивості:
`name` (повне ім'я товару)
`cc` (ціна товару)
Для об'єктів даного класу має визначатись один метод:
`TovarToString` (повертає строкову версію об'єкта)
6. Створити клас `Car`.
Для об'єктів даного класу має визначатись дві властивості:
`name` (повне ім'я машини)
`country` (країна-виробник)
Для об'єктів даного класу має визначатись один метод:
`CarToString` (повертає строкову версію об'єкта)
7. Створити клас `Country`.
Для об'єктів даного класу має визначатись дві властивості:
`name` (назва країни)
`capital` (назва столиці)
Для об'єктів даного класу має визначатись один метод:
`CountryToString` (повертає строкову версію об'єкта)
8. Створити клас `School`.
Для об'єктів даного класу має визначатись дві властивості:
`name` (повна назва школи)
`country` (країна)
Для об'єктів даного класу має визначатись один метод:
`SchoolToString` (повертає строкову версію об'єкта)
9. Створити клас `Sport`.
Для об'єктів даного класу має визначатись дві властивості:

- name (назва виду спорту)
location (країна - засновник цього виду спорту)
Для об'єктів даного класу має визначатись один метод:
SportToString (повертає строкову версію об'єкта)
10. Створити клас Theater.
Для об'єктів даного класу має визначатись дві властивості:
name (назва театру)
adres (адреса театру)
Для об'єктів даного класу має визначатись один метод:
TheaterToString (повертає строкову версію об'єкта)
11. Створити клас TV.
Для об'єктів даного класу має визначатись дві властивості:
name (компанія-виготівник телевізору)
characteristic (базові характеристики)
Для об'єктів даного класу має визначатись один метод:
TVToString (повертає строкову версію об'єкта)
12. Створити клас Capital.
Для об'єктів даного класу має визначатись дві властивості:
name (імя столиці будь-якої країни)
population (кількість мешканців даної столиці)
Для об'єктів даного класу має визначатись один метод:
CapitalToString (повертає строкову версію об'єкта)
13. Створити клас OS.
Для об'єктів даного класу має визначатись дві властивості:
name (імя операційної системи)
year (рік заснування)
Для об'єктів даного класу має визначатись один метод:
OSToString (повертає строкову версію об'єкта)
14. Створити клас Plane.
Для об'єктів даного класу має визначатись дві властивості:
name (повна назва літака)
country (країна-виробник)
Для об'єктів даного класу має визначатись один метод:
PlaneToString (повертає строкову версію об'єкта)
15. Створити клас University.
Для об'єктів даного класу має визначатись дві властивості:
name (повна назва університету)
location (країна)
Для об'єктів даного класу має визначатись один метод:
UniversityToString (повертає строкову версію об'єкта)

Лабораторна робота №10

Об'єктно-орієнтовані основи мови php. Клас PDO. Налаштування роботи з базою даних.

Ціль: Вивчення методів класа PDO для підключення до бази даних та опрацювання простих запитів до бази даних. Формування шаблону виведення результатів “Таблиця”

Зміст

- 1) Клас PDO для підключення до бази даних та опрацювання запитів до бази даних.
- 2) Приклад роботи з СУБД MySQL у середовищі PHP MyAdmin.
 - a) Створення БД та користувача.
 - b) Створення таблиць БД.
 - c) Заповнення таблиць даними.
- 3) Налаштування конфігураційного файлу для роботи з базою даних.
 - a) Константи конфігурування
 - b) Функція-методу підключення до БД
 - c) Методи опрацювання простих (без параметрів) запитів до БД
- 4) Формування шаблону виведення результатів SELECT запиту “Таблиця”.
- 5) Приклад рішення задачі.
- 6) Індивідуальні завдання.

Хід роботи:

- 1) **Клас PDO для підключення до бази даних та опрацювання запитів до бази даних**

PDO - це клас для роботи з СУБД.

У PDO спосіб з'єднання DSN.

Приклад 1. Налаштування з'єднання з базою даних

```
define("DB_HOST", "mysql.zzz.com.ua"); //визначаємо хост серверу БД
define("DB_USERNAME", "sasha123123"); //ім'я користувача бази даних
define("DB_PASSWORD", "Sasha123"); //пароль користувача бази даних
define("DB_NAME", "dimitrashko97"); //імя бази даних

$dbh = new PDO('mysql:host='.DB_HOST.';dbname='.DB_NAME,DB_USERNAME,DB_PASSWORD); //створюємо об'єкт класа PDO
$dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION); //налаштування опрацювання помилок
$dbh->exec("set names utf8"); //налаштування кодування об'єкта
```

Обробка винятків

Часто рекомендується при роботі з класом PDO користуватись конструкцією try-catch для оброблення винятків PDOException. Це можна опускати, PHP коректно виводить повідомлення про помилки, нам для роботи цього достатньо. Виконання запитів.

Для виконання запитів можна користуватися двома методами.

1. Якщо в запит не передаються ніякі змінні, то можна скористатися функцією query(). Вона виконає запит і поверне спеціальний об'єкт - PDO statement. Отримані дані будуть переведені у формат двовимірного асоціативного масиву методом fetchAll(PDO::FETCH_ASSOC).

Приклад 2. Якщо в SQL запит не передаються ніякі змінні

```
$stmt = $dbh-> query ( 'SELECT * FROM users');  
$result = $stmt->fetchAll(PDO::FETCH_ASSOC); //отримаємо масив  
//одновимірних масивів з ключами-іменами атрибутів таблиці users
```

2. Якщо в запит передаються змінні необхідно підготувати запит, а другою дією підставити підготовлені змінні. Отримані дані будуть переведені у формат двовимірного асоціативного масиву методом fetchAll(PDO::FETCH_ASSOC).

Приклад 3. Якщо в SQL запит передаються змінні

```
$stmt = $dbh-> prepare ( 'SELECT * FROM users WHERE id=:id and  
name=:name1');  
$stmt->execute(['id' => $id, 'name1' => $name1]); //визначаємо :id та :name1  
//та виконуємо SQL запит  
$result = $stmt->fetchAll(PDO::FETCH_ASSOC); //отримаємо масив  
//одновимірних масивів з ключами-іменами атрибутів таблиці users, що  
//відповідає умові SQL запиту
```

2) Приклад роботи з СУБД MySQL у середовищі PHP MyAdmin.

На хостингу в панелі керування необхідно створити базу даних, до неї користувача, що має ім'я та пароль і зафіксувати шлях до хоста баз даних (Рис.1).

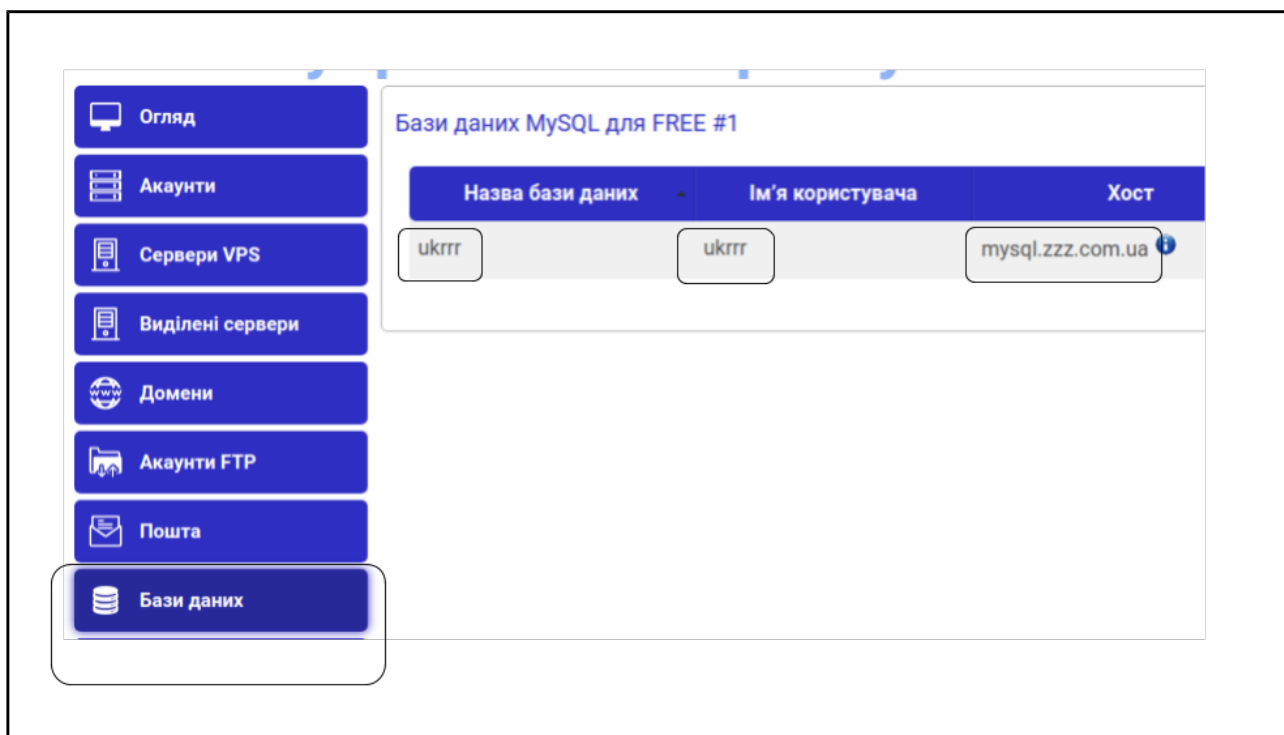


Рис.1. Заведення бази даних і користувача (Приклад для хостингу zzz.com.ua)
Дана інформація має бути відображена в відповідних константах конфігураційного файлу, що наведено у прикладі
Приклад 4. Налаштування констант конфігураційного файлу

```
define("DB_HOST", "mysql.zzz.com.ua");
define("DB_USERNAME", "sha123123");
define("DB_PASSWORD", "Sa6667896");
define("DB_NAME", "dimit");
```

Наступний етап - це створення таблиці бази даних та наповнення її щонайменше 5 записами. Для цього ми скористаємось SQL запитамі CREATE та INSERT

Приклад 5. Створюємо таблицю users з наповненням

```
CREATE TABLE `news` (
  `id` int(20) UNSIGNED NOT NULL,
  `name` varchar(255) NOT NULL,
  `description` text NOT NULL,
  `data` date NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=utf8;

INSERT INTO `news` (`id`, `name`, `description`, `data`) VALUES
(1, 'Новина №1', 'Детальна інформація №1', '2018-09-22'),
(2, 'Новина №2', 'Детальна інформація №2', '2018-09-11'),
(3, 'Новина №3', 'Детальна інформація №3', '2012-11-19'),
(4, 'Новина №4', 'Детальна інформація №4', '2012-11-13'),
(5, 'Новина №5', 'Детальна інформація №5', '2012-11-01');
```

Наведений у прикладі SQL код запитів запускаємо в середовищі PHP MyAdmin у розділі SQL (Рис. 2)

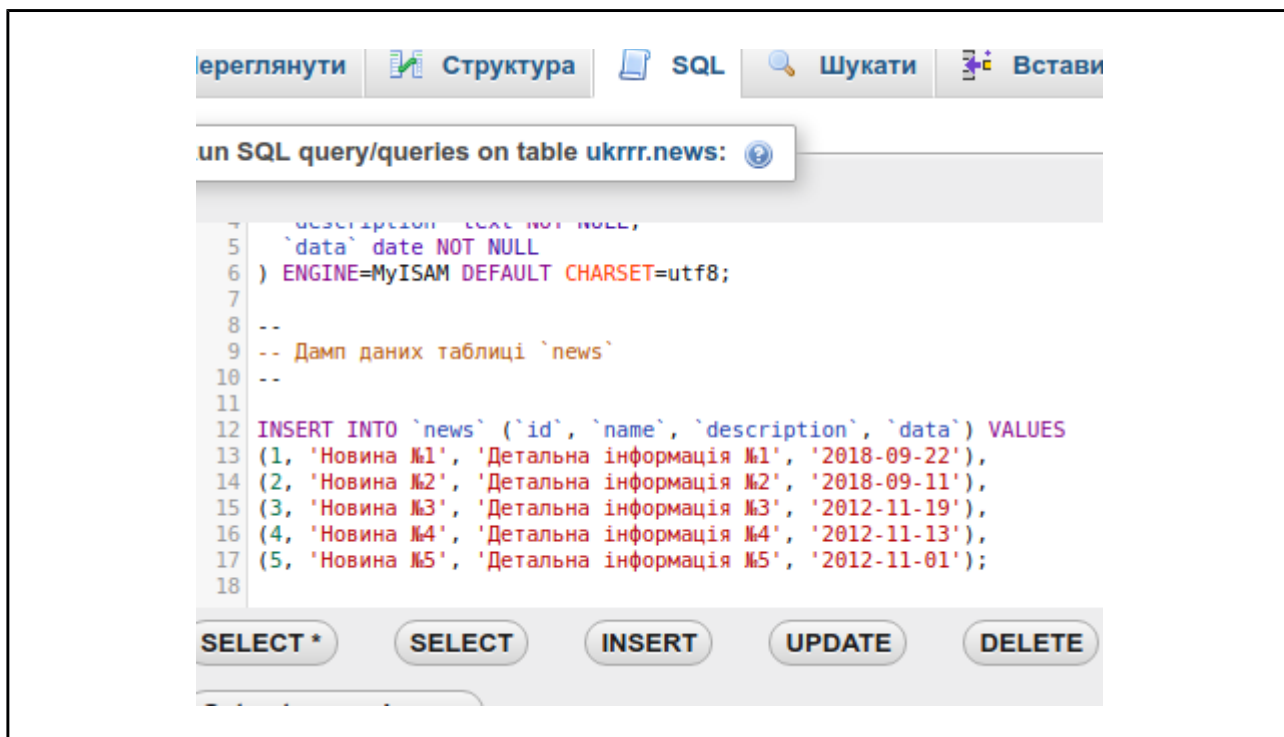


Рис.2. Виконання команд CREATE та INSERT в середовищі PHP MyAdmin
Після виконаних дій База даних з таблицею налаштована. Нам можна приступати до формування сторінок інформаційної системи на основі даних з таблиці бази даних.

3) Налаштування конфігураційного файлу для роботи з базою даних.

Першим етапом для організації роботи з базою даних в інформаційній системі стає налаштування конфігураційного файлу. Тут ми визначаємо константи, що забезпечують підключення до бази даних, а також функцію, в котрій формується об'єкт класу PDO, що опрацьовує роботу з базою даних. Єдиною різницею Прикладаб від Приклада1 стає оформлення з'єднання з базою даних в окрему функцію. Це необхідно тому, що конфігураційний файл підключається до кожного файлу частини контролеру, але не завжди ми потребуємо з'єднання з базою даних.

Приклад 6

```
define("DB_HOST", "mysql.zzz.com.ua");
define("DB_USERNAME", "sha123123");
define("DB_PASSWORD", "Sa6667896");
define("DB_NAME", "dimit");
function db_connect(){
    $dbch = new
PDO('mysql:host='.DB_HOST.';dbname='.DB_NAME,DB_USERNAME,
DB_PASSWORD);
    $dbch->setAttribute(PDO::ATTR_ERRMODE,
```

```
PDO::ERRMODE_EXCEPTION);
    $dbh->exec("set names utf8");
    return $dbh;
}
```

4) Приклад виконання індивідуального завдання.

Після налаштування конфігураційного файлу в наведений спосіб розглянемо приклад рішення задачі на вибірку всіх записів з таблиці news бази даних. Файл частини контролера lab10.php, в якому виконується виклик функції для підключення бази даних, опрацювання запиту SQL, формування асоціативного масиву-результату та передача результату, що отримано в шаблон.

Приклад 7 (lab10.php)

```
<?php
require_once('config.php');
$dbh = db_connect();//підключення до БД
$stmt = $dbh->query('SELECT * FROM news');// Виконується SQL запит
$result = $stmt->fetchAll(PDO::FETCH_ASSOC);// Результат в масив
$tpl_vars = array('result' => $result);
$tpl_file = 'lab10.tpl';
require_once('tpl_config.php');
?>
```

Відповідно сформуємо шаблон lab10.tpl для виведення результатів у вигляді таблиці. Тут використано два вкладених цикли. Зовнішній - по елементах масиву result, а внутрішній обробляє один кортеж - результат.

Приклад 7 (lab10.tpl)

```
{% extends 'base.tpl' %}
{% block content %}
<table border="5">
{% for user_value in result %} //result - переданий з контроллера результат
<tr>
{% for key, value in user_value %} //user_value - це черговий рядок таблиці
<td>
{{value}} //виводимо значення чергового атрибута в рядку таблиці news
</td>
{% endfor %} //кінець внутрішнього циклу
</tr>
{% endfor %} // кінець зовнішнього циклу
</table>
{% endblock content %}
```

У результаті отримаємо дані з таблиці news у табличному форматі (рис. 3)

Наш перший сайт			
Гловна Лабораторна робота №2 Лабораторна робота №3 Лабораторна робота №6 Лабораторна робота №8 Адмін розділ © 2016	1	Новина №1	Детальна інформація №1
	2	Новина №2	Детальна інформація №2
	3	Новина №3	Детальна інформація №3
	4	Новина №4	Детальна інформація №4
	5	Новина №5	Детальна інформація №5

Рис. 3. Результат вибірки з таблиці у табличному шаблоні

Важливим є той факт, що для отримання результату любого SQL запиту на вибірку інформації достатньо в частині контроллера поміняти SELECT запит.

5) Індивідуальні завдання.

1- За своїм порядковим номером у групі вибрати варіант індивідуального завдання з розділу 5

2- Побудувати таблиці бази даних предметної галузі за своїм варіантом, встановити відповідні зв'язки з урахуванням посильної цілісності та цілісності сутностей

3- Наповнити базу даних даними з відповідної предметної галузі. Щонайменше по 5 записів в кожну таблицю. При наповненні враховуйте посильну цілісність!

4- Створити 3 сторінки сайту, що з використанням одного шаблону представляють результат на вибірку всіх даних з кожної таблиці бази даних у табличному форматі.

5- Варіанти індивідуальної бази даних предметної галузі:

1. Аналіз динаміки показників фінансової звітності різних підприємств

Опис предметної галузі

Ви є керівником інформаційно-аналітичного центру великого холдингу. Вашим завданням є відстеження динаміки показників для підприємств Вашого холдингу.

У структуру холдингу входять кілька підприємств. Кожне підприємство має стандартні характеристики (назва, реквізити, телефон, контактна особа). Робота підприємства може бути оцінена наступним чином: на початку кожного звітного періоду на основі фінансової звітності обчислюється за деякими формулами певний набір показників. Прийняти, що важливість показників

характеризується деякими числовими константами. Значення кожного показника вимірюється в деякій системі одиниць.

Таблиці:

Показники (Код показника, Назва, Важливість, Одиниця виміру).

Підприємства (Код підприємства, Назва, Банківські реквізити, Телефон, Контактна особа).

Динаміка показників (Код, Код показника, Код підприємства, Дата, Значення).

2. Бібліотека

Опис предметної галузі

Ви є керівником бібліотеки. Ваша бібліотека вирішила заробляти гроші, видаючи напрокат деякі книги, наявні в невеликій кількості примірників. Вашим завданням є відстеження фінансових показників роботи бібліотеки.

У кожної книги, що видається в прокат, є назва, автор, жанр. Залежно від цінності книги Ви визначили для кожної з них заставну вартість (сума, яку вносить клієнтом при взятті книги напрокат) і вартість прокату (сума, яку клієнт платить при поверненні книги, отримуючи назад заставу). У бібліотеку звертаються читачі. Всі читачі реєструються в картотеці, яка містить стандартні анкетні дані (прізвище, ім'я, по батькові, адреса, телефон). Кожен читач може звертатися в бібліотеку кілька разів. Всі звернення читачів фіксуються, при цьому по кожному факту видачі книги запам'ятовуються дата видачі та очікувана дата повернення.

Таблиці:

Книги (Код книги, Назва, Автор, Заставна вартість, Вартість прокату, Жанр).

Читачі (Код читача, Прізвище, Ім'я, По батькові, адреса, телефон).

Видані книги (Код, Код книги, Код читача, Дата видачі, Дата повернення).

3. Центр Зайнятості

Опис предметної галузі.

Ви працюєте в бюро з працевлаштування. Вашим завданням є відстеження фінансової сторони роботи компанії.

Діяльність Вашого бюро організована таким чином: Ваше бюро готове шукати працівників для різних роботодавців і вакансії для тих, хто шукає роботу фахівців різного профілю. При зверненні до Вас клієнта-роботодавця, його стандартні дані (назва, вид діяльності, адреса, телефон) фіксуються в базі даних. При зверненні до Вас клієнта-здобувача, його стандартні дані (прізвище, ім'я, по батькові, кваліфікація, професія, інші дані) також фіксуються в базі даних. За кожним фактом задоволення інтересів обох сторін складається документ. У документі вказуються здобувач, роботодавець, посаду і комісійні (дохід бюро).

Таблиці:

Роботодавці (Код роботодавця, Назва, Вид діяльності, Адреса, Телефон).

Угоди (Код, Код здобувача, Код роботодавця, Посада, Комісійні).

Здобувачі (Код здобувача, Прізвище, Ім'я, По батькові, Кваліфікація, Вид діяльності, Інші дані, Передбачуваний розмір заробітної плати).

4. Контролювання замовлень

Опис предметної галузі

Ви працюєте в компанії, що займається оптовим продажем різних товарів. Вашим завданням є відстеження фінансової сторони роботи компанії.

Діяльність Вашої компанії організована в такий спосіб: Ваша компанія торгує товарами з певного спектру. Кожен з цих товарів характеризується ціною, довідковою інформацією і ознакою наявності або відсутності доставки. У Вашу компанію звертаються замовники. Для кожного з них Ви запам'ятовуєте в базі даних стандартні дані (найменування, адреса, телефон, контактна особа) і складаєте по кожній угоді документ, запам'ятовуючи поряд з замовником кількість купленого ним товару і дату покупки.

Таблиці:

Замовники (Код замовника, Найменування, адресу, телефон, Контактна особа).

Товари (Код товару, Ціна, Доставка, Опис).

Замовлення (Код, Код замовника, Код товару, Кількість, Дата).

5. Видача банком кредитів

Опис предметної галузі.

Ви є керівником інформаційно-аналітичного центру комерційного банку. Одним з істотних видів діяльності Вашого банку є видача кредитів юридичним особам. Вашим завданням є відстеження динаміки роботи кредитного відділу.

Залежно від умов отримання кредиту, процентної ставки і терміну повернення всі кредитні операції поділяються на кілька основних видів. Кожен з цих видів має свою назву. Кредит може отримати юридична особа (клієнт), при реєстрації надав такі відомості: назва, вид власності, адреса, телефон, контактна особа. Кожен факт видачі кредиту реєструється банком, при цьому фіксуються сума кредиту, клієнт і дата видачі.

Таблиці:

Види кредитів (Код виду, Назва, Умови отримання, Ставка, Термін).

Клієнти (Код клієнта, Назва, Вид власності, Адреса, Телефон, Контактна особа).

Кредити (Код, Код виду, Код клієнта, Сума, Дата видачі).

6. Готель

Опис предметної галузі.

Ви працюєте в готелі. Вашим завданням є відстеження фінансової сторони роботи готелю.

Ваша діяльність організована таким чином: готель надає номери клієнтам на певний термін. Кожен номер характеризується місткістю, комфортністю (люкс, напівлюкс, звичайний) і ціною. Вашими клієнтами є різні особи, про яких Ви збираєте певну інформацію (прізвище, ім'я, по батькові та деякий коментар). Здача номера клієнту проводиться при наявності вільних місць в номерах, що

підходять клієнту за вказаними вище параметрами. При поселенні фіксується дата поселення. При виїзді з готелю для кожного місця запам'ятовується дата звільнення.

Таблиці:

Клієнти (Код клієнта, Прізвище, Ім'я, По батькові, паспортні дані, Коментар).

Номери (Код номера, Номер, Кількість людей, Комфортність, Ціна).

Поселення (Код, Код клієнта, Код номера, Дата поселення, Дата звільнення, Примітка).

7. Вантажні перевезення

Опис предметної галузі.

Ви працюєте в компанії, що займається перевезеннями вантажів. Вашим завданням є відстеження вартості перевезень з урахуванням заробітної плати водіїв.

Ваша компанія здійснює перевезення по різних маршрутах. Для кожного маршруту Ви визначили деякий назву, вираховували приблизний відстань і встановили деяку оплату для водія. Інформація про водіїв включає прізвище, ім'я, по батькові та стаж. Для проведення розрахунків Ви зберігаєте повну інформацію про перевезення (маршрут, водій, дати відправлення і прибуття). За фактом деяких перевезень водіям виплачується премія.

Таблиці:

Маршрути (Код маршруту, Назва, Дальність, Кількість днів у шляху, Оплата).

Водії (Код водія, Прізвище, Ім'я, По батькові, Стаж).

Виконана робота (Код, Код маршруту, Код водія, Дата відправки, Дата повернення, Премія).

8. Зайнятість акторів театру

Опис предметної галузі.

Ви є комерційним директором театру, і в Ваші обов'язки входить вся організаційно-фінансова робота, пов'язана із залученням акторів і укладанням контрактів. Ви поставили справу таким чином: кожен рік театр здійснює постановку різних вистав. Кожен спектакль має певний бюджет. Для участі в конкретних постановках в певних ролях Ви залучаєте акторів. З кожним з акторів Ви укладаєте персональний контракт на певну суму. Кожен з акторів має певний стаж роботи, деякі з них удостоєні різних нагород і звань.

Таблиці:

Актори (Код актора, Прізвище, Ім'я, По батькові, Звання, Стаж).

Вистави (Код вистави, Назва, Рік постановки, Бюджет).

Зайнятість акторів у виставі (Код, Код актора, Код вистави, Роль, Вартість річного контракту).

9. Інтернет-магазин

Опис предметної галузі.

Ви є співробітником комерційного відділу компанії, що продає різні товари через Інтернет. Вашим завданням є відстеження фінансової складової роботи компанії.

Робота Вашої компанії організована в такий спосіб: на Інтернет-сайті компанії представлені (виставлені на продаж) деякі товари. Кожен з них має деякий назву, ціну і одиницю виміру (штуки, кілограми, літри). Для проведення досліджень і оптимізації роботи магазину Ви намагаєтеся збирати дані з Ваших клієнтів. При цьому для Вас визначальне значення мають стандартні анкетні дані, а також телефон та адреса електронної пошти для зв'язку. У разі придбання товарів на суму понад 5000р. клієнт переходить в категорію «постійних клієнтів» і отримує знижку на кожну покупку в розмірі 2%. За кожним фактом продажу Ви автоматично фіксуєте клієнта, товари, кількість, дату продажу, дату доставки.

Таблиці:

Товари (Код товару, Назва, Ціна, Одиниця виміру).

Клієнти (Код клієнта, Прізвище, Ім'я, По батькові, адреса, телефон, e-mail, Ознака постійного клієнта).

Продажі (Код, Код товару, Код клієнта, Дата продажу, Дата доставки, Кількість).

10. Інвестування вільних коштів

Опис предметної галузі.

Ви є керівником аналітичного центру інвестиційної компанії. Ваша компанія займається вкладенням коштів у цінні папери.

Ваші клієнти - підприємства, які довіряють Вам управляти їх вільними грошовими коштами на певний період. Вам необхідно вибрати вид цінних паперів, які дозволять отримати прибуток і Вам і Вашому клієнтові. При роботі з клієнтом для Вас дуже істотною є інформація про підприємство - назва, вид власності, адреса і телефон.

Таблиці:

Цінні папери (Код цінного паперу, Мінімальна сума угоди, Рейтинг, Прибутковість за минулий рік, Додаткова інформація).

Інвестиції (Код, Код цінного паперу, Код клієнта, Котирування, Дата покупки, Дата продажу).

Клієнти (Код клієнта, Назва, Вид власності, Адреса, Телефон).

11. Курси по підвищенню кваліфікації

Опис предметної області

Ви працюєте в навчальному закладі і займаєтеся організацією курсів підвищення кваліфікації.

У Вашому розпорядженні є відомості про сформовані групи студентів. Групи формуються в залежності від спеціальності та відділення. У кожній з них включено певну кількість студентів. Проведення занять забезпечує штат викладачів. Для кожного з них у Вас в базі даних зареєстровані стандартні анкетні дані (прізвище, ім'я, по батькові, телефон) і стаж роботи. В результаті

розподілу навантаження Ви отримуєте інформацію про те, скільки годин занять проводить кожен викладач з відповідними групами. Крім того, зберігаються також відомості про вид проведених занять (лекції, практика), предмет і оплата за 1 годину.

Таблиці:

Групи (Номер групи, Спеціальність, Відділення, Кількість студентів).

Викладачі (Код викладача, Прізвище, Ім'я, По батькові, Телефон, Стаж).

Навантаження (Код, Код викладача, Номер групи, Кількість годин, Предмет, Тип заняття, Оплата).

12. Ломбард

Опис предметної галузі

Ви працюєте в ломбарді. Вашим завданням є відстеження фінансової сторони роботи ломбарду.

Діяльність Вашої компанії організована в такий спосіб: до Вас звертаються різні особи з метою отримання грошових коштів під заставу певних товарів. У кожного що приходите до Вас клієнтів Ви запитуєте прізвище, ім'я, по батькові та інші паспортні дані. Після оцінювання вартості принесеного в якості застави товару Ви визначаєте суму, яку готові видати на руки клієнту, а також свої комісійні. Крім того, визначаєте термін повернення грошей. Якщо клієнт згоден, то Ваші домовленості фіксуються у вигляді документа, гроші видаються клієнту, а товар залишається у Вас. У разі якщо в зазначений термін не відбувається повернення грошей, товар переходить у Вашу власність.

таблиці

Клієнти (Код клієнта, Прізвище, Ім'я, По батькові, Номер паспорта, Серія паспорта, Дата видачі паспорта).

Категорії товарів (Код категорії товарів, Назва, Примітка).

Здача в ломбард (Код, Код категорії товарів, Код клієнта, Опис товару, Дата здачі, Дата повернення, Сума, Комісійні).

13. Нотаріальна контора

Опис предметної області

Ви працюєте в нотаріальній конторі. Вашим завданням є відстеження фінансової сторони роботи компанії.

Діяльність Вашої нотаріальної контори організована таким чином: Ваша фірма готова надати клієнтові певний комплекс послуг. Для наведення порядку Ви формалізували ці послуги, склавши їх список з описом кожної послуги. При зверненні до Вас клієнта, його стандартні дані (назва, вид діяльності, адреса, телефон) фіксуються в базі даних. За кожним фактом надання послуги клієнту складається документ. У документі вказуються послуга, сума угоди, комісійні (дохід контори), опис угоди.

таблиці

Клієнти (Код клієнта, Назва, Вид діяльності, Адреса, Телефон).

Угоди (Код, Код клієнта, Код послуги, Сума, Комісійні, Опис).

Послуги (Код послуги, Назва, Опис).

14. Визначення факультативів для студентів

Опис предметної галузі

Ви працюєте у вищому навчальному закладі і займаєтесь організацією факультативів.

У Вашому розпорядженні є відомості про студентів, які включають стандартні анкетні дані (прізвище, ім'я, по батькові, адреса, телефон). Викладачі Вашої кафедри повинні забезпечити проведення факультативних занять з деяких предметів. По кожному факультативу існує певна кількість годин і вид проведених занять (лекції, практика, лабораторні роботи). В результаті роботи зі студентами у Вас з'являється інформація про те, хто з них записався на які факультативи. Існує певний мінімальний обсяг факультативних предметів, які повинен прослухати кожен студент. По закінченню семестру Ви вносите інформацію про оцінки, отримані студентами на іспитах.

Таблиці:

Студенти (Код студента, Прізвище, Ім'я, По батькові, адресу, телефон).

Предмети (Код предмета, Назва, Обсяг лекцій, Обсяг практик, Обсяг лабораторних робіт).

Навчальний план (Код, Код студента, Код предмета, Оцінка).

15. Перукарня

Опис предметної галузі.

Ви працюєте в перукарні.

Ваша перукарня стриже клієнтів відповідно до їх побажань і деяким каталогом різних видів стрижки. Так, для кожної стрижки визначені назва, приналежність підлозі (чоловіча, жіноча), вартість роботи. Для наведення порядку Ви, у міру можливості, складаєте базу даних клієнтів, запам'ятовуючи їх анкетні дані (прізвище, ім'я, по батькові). Починаючи з 5-ої стрижки, клієнт переходить в категорію постійних і отримує знижку в 3% при кожній наступній стрижці. Після того, як закінчена чергова робота, в касі фіксуються стрижка, клієнт і дата виконання робіт.

Таблиці:

Стрижки (Код стрижки, Назва, Пол, Вартість).

Клієнти (Код клієнта, Прізвище, Ім'я, По батькові, Пол, Ознака постійного клієнта).

Робота (Код, Код стрижки, Код клієнта, Дата).

16. Платна поліклініка

Опис предметної галузі.

Ви є керівником служби планування у платній поліклініці. Вашим завданням є відстеження фінансових показників роботи поліклініки.

У поліклініці працюють лікарі різних спеціальностей, які мають різну кваліфікацію. Кожен день в поліклініку звертаються хворі. Всі хворі проходять

обов'язкову реєстрацію, при якій в базу даних заносяться стандартні анкетні дані (прізвище, ім'я, по батькові, рік народження). Кожен хворий може звертатися в поліклініку кілька разів, потребуючи різної медичної допомоги. Всі звернення хворих фіксуються, при цьому встановлюється діагноз, визначається вартість лікування, запам'ятовується дата звернення.

Таблиці:

Лікарі (Код лікаря, Прізвище, Ім'я, По батькові, Спеціальність, Категорія).

Пацієнти (Код пацієнта, Прізвище, Ім'я, По батькові, Рік народження).

Звернення (Код, Код лікаря, Код пацієнта, Дата звернення, Діагноз, Вартість лікування).

16. Платна поліклініка

Опис предметної області

Ві є керівником служби планування платній поліклініки. Вашим завдання є відстеження ФІНАНСОВИХ показників роботи поліклініки.

У поліклініці Працюють лікарі різних спеціальностей, Які мають різну кваліфікацію. Кожен день в поліклініку звертають Хворі. Всі Хворі проходять обов'язкову реєстрацію, при якій в базу Даних заносяться Стандартні анкетні дані (прізвище, ім'я, по батькові, рік народження). КОЖЕН хворий может звертатися в поліклініку кілька разів, потребуючи різної медичної допомоги. Всі Звернення Хворих фіксуються, при цьому встановлюється діагноз, визначається ВАРТІСТЬ лікування, запам'ятовується дата Звернення.

Таблиці:

Лікарі (Код лікаря, Прізвище, Ім'я, По батькові, Спеціальність, Категорія).

Пацієнти (Код пацієнта, Прізвище, Ім'я, По батькові, Рік народження).

Звернення (Код, Код лікаря, Код пацієнта, Дата Звернення, Діагноз, Стоимость лікування).

17. Прокат автомобілів

Опис предметної галузі.

Ви є керівником комерційної служби у фірмі, що займається прокатом автомобілів. Вашим завданням є відстеження фінансових показників роботи пункту прокату.

У Ваш автопарк входить деяка кількість автомобілів різних марок, вартостей і типів. Кожен автомобіль має свою вартість прокату. В пункт прокату звертаються клієнти. Всі клієнти проходять обов'язкову реєстрацію, при якій про них збирається стандартна інформація (прізвище, ім'я, по батькові, адреса, телефон). Кожен клієнт може звертатися в пункт прокату кілька разів. Всі звернення клієнтів фіксуються, при цьому по кожній угоді запам'ятовуються дата видачі та очікувана дата повернення.

таблиці

Автомобілі (Код автомобіля, Марка, Вартість, Вартість прокату, Тип).

Клієнти (Код клієнта, Прізвище, Ім'я, По батькові, адресу, телефон).

Видані автомобілі (Код, Код автомобіля, Код клієнта, Дата видачі, Дата повернення).

18. Розподіл додаткових обов'язків

Опис предметної галузі.

Ви працюєте в комерційній компанії і займаєтесь розподілом додаткових разових робіт. Вашим завданням є відстеження ходу виконання додаткових робіт.

Компанія має певний штат співробітників, кожен з яких отримує певний оклад. Час від часу, виникає потреба у виконанні деякої додаткової роботи, яка не входить в коло основних посадових обов'язків співробітників. Для наведення порядку в цій сфері діяльності Ви повинні класифікувати всі види додаткових робіт, визначившись з сумою оплати за фактом їх виконання. При виникненні додаткової роботи певного виду Ви призначаєте відповідального, фіксуючи дату початку. За фактом закінчення Ви фіксуєте дату і виплачуєте додаткову суму до зарплати з урахуванням Вашої класифікації.

Таблиці:

Працівники (Код співробітника, Прізвище, Ім'я, По батькові, Оклад).

Види робіт (Код виду, Опис, Оплата за день).

Роботи (Код, Код співробітника, Код виду, Дата початку, Дата закінчення).

19. Розподіл навчального навантаження

Опис предметної галузі.

Ви працюєте у вищому навчальному закладі і займаєтесь розподілом навантаження між викладачами кафедри.

У Вашому розпорядженні є відомості про викладачів кафедри, що включають поряд з анкетними даними відомості про їх наукового ступеня, займаної адміністративної посади та стаж роботи. Викладачі Вашої кафедри повинні забезпечити проведення занять з деяких предметів. По кожному з них існує певна кількість годин. В результаті розподілу навантаження у Вас повинні бути сформовані дані наступного роду: «Такий-то викладач проводить заняття за такою-то предмету з такою-то групою».

Таблиці:

Викладачі (Код викладача, Прізвище, Ім'я, По батькові, Вчений ступінь, посада, Стаж).

Предмети (Код предмета, Назва, Кількість годин).

Навантаження (Код, Код викладача, Код предмета, Номер групи).

20. Реалізація готової продукції

Опис предметної галузі.

Ви працюєте в компанії, що займається оптово-роздрібним продажем різних товарів. Вашим завданням є відстеження фінансової сторони роботи компанії.

Діяльність Вашої компанії організована в такий спосіб: Ваша компанія торгує товарами з певного спектру. Кожен з цих товарів характеризується найменуванням, оптовою ціною, роздрібною ціною і довідковою інформацією. У Вашу компанію звертаються покупці. Для кожного з них Ви запам'ятовуєте в

базі даних стандартні дані (найменування, адреса, телефон, контактна особа) і складаєте по кожній угоді документ, запам'ятовуючи поряд з покупцем кількість купленого ним товару і дату покупки.

Таблиці:

Товари (Код товару, Найменування, Оптова ціна, Роздрібна ціна, Опис).

Покупці (Код покупця, Телефон, Контактна особа, Адреса).

Угоди (Код, Код покупця, Код товару, Кількість, Дата угоди, Ознака оптового продажу).

21. Здача в оренду торгових площ

Опис предметної галузі.

Ви працюєте у великому торговому центрі, який здає в оренду комерсантам свої торгові площі.

Вашим завданням є наведення ладу у фінансовій стороні роботи торгового центру.

Роботи Вашого торгового центру побудована таким чином: в результаті планування Ви визначили кілька торгових точок в межах Вашої будівлі, які можуть здаватися в оренду. Для кожної з торгових точок важливими даними є поверх, площа, наявність кондиціонера і вартість оренди в день. З усіх потенційних клієнтів Ви збираєте стандартні дані (назва, адреса, телефон, реквізити, контактна особа). При появі потенційного клієнта Ви показуєте йому наявні вільні площі. При досягненні угоди Ви оформляєте договір, фіксуючи в базі даних торгову точку, клієнта, період (строк) оренди.

Таблиці:

Торгові точки (Код торгової точки, Поверх, Площа, Наявність кондиціонера, Вартість оренди в день).

Клієнти (Код клієнта, Назва, Реквізити, Адреса, Телефон, Контактна особа).

Оренда (Код, Код торгової точки, Код клієнта, Дата початку, Дата закінчення).

22. Страхова компанія

Опис предметної галузі.

Ви працюєте в страховій компанії. Вашим завданням є відстеження фінансової діяльності компанії.

Компанія має різні філії по всій країні. Кожна філія характеризується назвою, адресою та телефоном. Діяльність компанії організована в такий спосіб: до Вас звертаються різні особи з метою укладення договору про страхування. Залежно від прийнятих на страхування об'єктів і застрахованих ризиків, договір укладається за певним видом страхування (наприклад, страхування автотранспорту від викрадення, страхування домашнього майна, добровільне медичне страхування). При укладанні договору Ви фіксуєте дату укладення, страхову суму, вид страхування, тарифну ставку і філія, в якому укладався договір.

Таблиці:

Вид страхування (Код виду страхування, Найменування).

Філія (Код філії, Найменування філії, Адреса, Телефон).

Договори (Код, Код філії, Код виду страхування, Дата укладення, Страхова сума, Тарифна ставка).

23. Технічне обслуговування верстатів

Опис предметної галузі.

Ваше підприємство займається ремонтом верстатів і іншого промислового устаткування. Вашим завданням є відстеження фінансової сторони діяльності підприємства.

Клієнтами Вашої компанії є промислові підприємства, оснащені різним складним обладнанням. У разі поломок обладнання вони звертаються до Вас.

Ремонтні роботи у Вашій компанії організовані в такий спосіб: всі верстати прокласифицировать по країнам-виробникам, рокам випуску і маркам. Всі види ремонту відрізняються назвою, тривалістю в днях, вартістю. Виходячи з цих даних, по кожному факту ремонту Ви фіксуєте вид верстата і дату початку ремонту.

Таблиці:

Види верстатів (Код виду верстата, Країна, Рік випуску, Марка).

Види ремонту (Код ремонту, Назва, Тривалість, Вартість, Примітки).

Ремонт (Код, Код виду верстата, Код ремонту, Дата початку, Примітки).

24. Туристична фірма

Опис предметної області

Ви працюєте в туристичній компанії. Ваша компанія працює з клієнтами, продаючи їм путівки. Вашим завданням є відстеження фінансової сторони діяльності фірми.

Робота з клієнтами у Вашій компанії організована в такий спосіб: у кожного клієнта, який прийшов до Вас, збираються деякі стандартні дані - прізвище, ім'я, по батькові, адреса, телефон. Після цього Ваші співробітники з'ясовують у клієнта, куди він хотів би поїхати відпочивати. При цьому йому демонструються різні варіанти, що включають країну проживання, особливості місцевого клімату, наявні готелі різного класу. Поряд з цим, обговорюється можлива тривалість перебування і вартість путівки. У разі якщо вдалося домовитися, і знайти для клієнта прийнятний варіант, Ви реєструєте факт продажу путівки (або путівок, якщо клієнт купує відразу кілька путівок), фіксуючи дату відправлення. Іноді Ви вирішуєте надати клієнту певну знижку.

Таблиці:

Маршрути (Код маршруту, Країна, Клімат, Тривалість, Готель, Вартість).

Клієнти (Код клієнта, Прізвище, Ім'я, По батькові, адресу, телефон).

Путівки (Код, Код маршруту, Код клієнта, Дата відправлення, Кількість, Знижка).

25. Облік внутрішньо офісних витрат

Опис предметної галузі.

Ви працюєте в бухгалтерії приватної фірми. Співробітники фірми мають можливість здійснювати дрібні покупки для потреб фірми, надаючи в бухгалтерію товарний чек. Вашим завданням є відстеження внутрішньо офісних витрат.

Ваша фірма складається з відділів. Кожен відділ має назву. У кожному відділі працює певна кількість співробітників. Співробітники можуть здійснювати покупки відповідно до видів витрат. Кожен вид витрат має назву, деякий опис і граничну суму коштів, які можуть бути витрачені по даному виду витрат в місяць. При кожній покупці співробітник оформляє документ, де вказує вид витрат, дату, суму і відділ.

Таблиці:

Відділи (Код відділу, Назва, Кількість працівників).

Види витрат (Код виду, Назва, Опис, Гранична норма).

Витрати (Код, Код виду, Код відділу, Сума, Дата).

Лабораторна робота №11

Опрацювання більш складних запитів до бази даних. Повторення мови SQL.

Ціль: Повторення базової структури SELECT запиту. Виведення результатів вибірки у шаблон типу “Блог”.

Зміст

- 1) Мова SQL. SELECT запит. Повторення.
- 2) Формування шаблону для виведення результатів у форматі блогу.
- 3) Приклад рішення задачі.
- 4) Індивідуальні завдання.

Хід роботи:

- 1) Мова SQL. SELECT запит. Повторення

```
SELECT [STRAIGHT_JOIN]
        [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
        [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
[HIGH_PRIORITY]
        [DISTINCT | DISTINCTROW | ALL]
select_expression,...
[INTO {OUTFILE | DUMPFILE} 'file_name' export_options]
[FROM table_references
        [WHERE where_definition]
        [GROUP BY {unsigned_integer | col_name | formula} [ASC | DESC], ...]
        [HAVING where_definition]
        [ORDER BY {unsigned_integer | col_name | formula} [ASC | DESC], ...]
        [LIMIT [offset,] rows]
        [PROCEDURE procedure_name]
        [FOR UPDATE | LOCK IN SHARE MODE]]
```

1.1 Формування запитів до двох таблиць з використанням конструкції JOIN.

В даній лабораторній роботі в першу чергу необхідно згадати базові правила щодо формування запиту до 2-х таблиць. Представимо, що в таблиці, що має назву А первинний ключ - це атрибут, що має назву РК. Таблиця, що має назву В посилається на таблицю А за допомогою зовнішнього ключа на ім'я FK. За таких стартових умов надаються приклади вибірки перетину, лівого та правого об'єднання відношень. Відповідні SQL запити представлені на рисунку (рис.1).

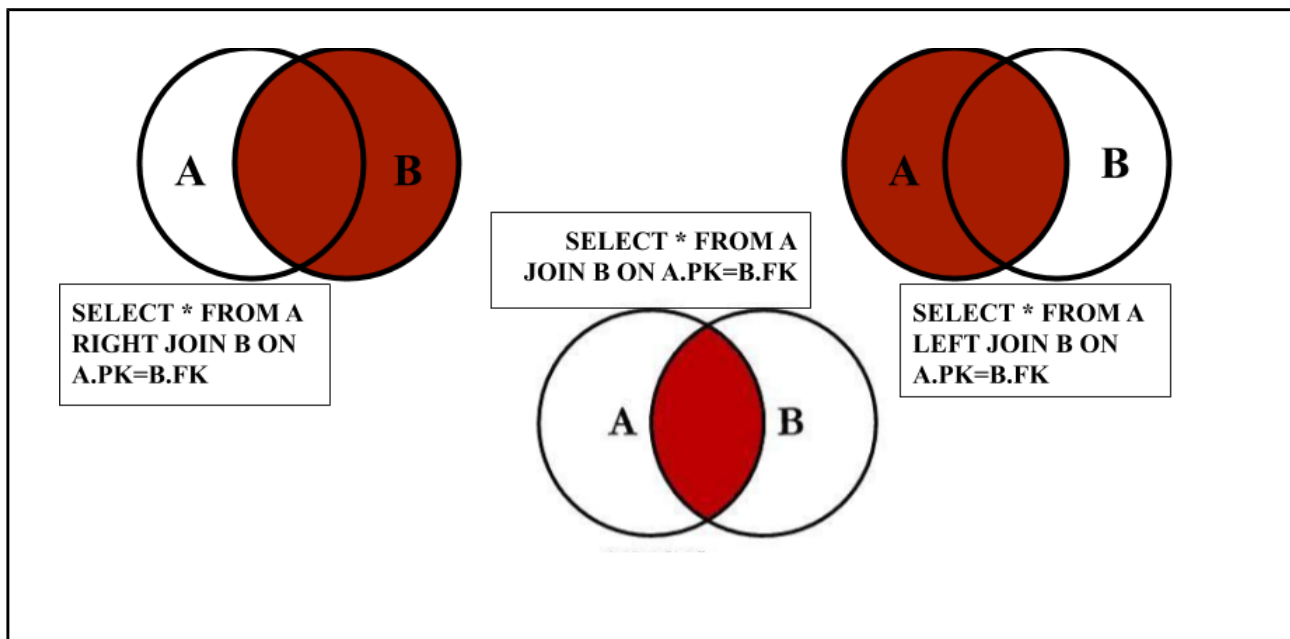


Рис. 1. Базові правила об'єднання відношень

1.2 Формування обмеження результуючих кортежей SELECT запиту за допомогою конструкції WHERE.

У простих умовах значення одного виразу (зазвичай значення відповідного атрибуту кожного кортежа) порівнюється із значенням іншого виразу (зазвичай конкретне значення). Прості умови відбору, після застосування до деякого атрибуту повертають для кожного кортежа значення TRUE, FALSE або NULL. За допомогою правил логіки ці прості умови можна об'єднувати в більш складні, використовуючи при цьому логічні операції AND, OR, NOT.

Операції для побудови простих запитів:

- Операції порівняння для числових даних: = , < > , < , <= , > , >=.
- Операції порівняння для даних типу дата: BETWEEN, NOT BETWEEN
- Операції порівняння для строкового типу: LIKE шаблон або ім'я стовпця NOT LIKE шаблон. Для побудови шаблонів використовуємо: % співпадає з будь-якою послідовністю з нуля чи більше символів; _ (символ підкреслення) співпадає з будь-яким окремим символом; ESCAPE символ пропуску.
- Операції порівняння значення атрибуту з NULL: ім'я стовпця IS NULL або ім'я стовпця IS NOT NULL.
- Перевірка приналежності до множини значень: вираз, що перевіряється IN (список констант відокремлених комами) або вираз, що перевіряється NOT IN (список констант відокремлених комами)

Для впорядкування результатів запиту використовується конструкція ORDER BY. ORDER BY ім'я_атрибуту ASC/DESC. При впорядкуванні можна обирати зростаючий (ASC) або спадний (DESC) порядок. За замовчуванням дані сортуються по зростанню. При наявності сортування по 2-м та більше атрибутам в першу чергу все відсортується за даними по першому атрибуту, а для однакових даних цього атрибуту буде виконано сортування по другому атрибуту і так далі.

1.3 Формування запитів на основі агрегатних функцій.

Для проведення підсумків по інформації, яка міститься в базі даних, застосовуються агрегатні функції. Агрегатна функція приймає в якості аргументу будь-який атрибут (стовпець даних), а повертає одне значення. Базові агрегатні функції: `sum(стовпець/вираз)` обчислює суму всіх значень, `avg(стовпець/вираз)` обчислює середнє всіх значень, `min(стовпець/вираз)` знаходить найменше серед всіх значень, `max(стовпець/вираз)` знаходить найбільше серед всіх значень, `count(стовпець)` підраховує кількість значень, що містяться в стовпці `count(*)` підраховує кількість рядків в таблиці - результату запиту

Приклад1

```
SELECT COUNT(*) FROM A//повертає число - кількість кортежей в A
```

Запити з групуванням (блок `GROUP BY`) Запит, який включає в себе блок `GROUP BY`, називається запитом з групуванням, оскільки він об'єднує рядки початкових таблиць в групи. Групування має бути виконано по всім атрибутам результату `SELECT` запиту за умови наявності одної або більше агрегатних функцій.

Приклад2

```
SELECT country, COUNT(*) FROM A GROUP BY (country)//повертає таблицю  
- назва country та кількість кортежей, з такою назвою country
```

Умови відбору груп (`HAVING`). Так само як `WHERE` використовується для відбору окремих рядків. Блок `HAVING` необхідно використовувати для формування умов, в котрих фігурують результати обчислень агрегатних функцій.

Приклад3

```
SELECT country, COUNT(*) FROM A  
GROUP BY (country)  
HAVING COUNT(*)>5;  
повертає таблицю - назви country, для яких кількість кортежей, з такою назвою  
country перевищує 5
```

2) Формування шаблону для виведення результатів `SELECT` запиту у форматі блогу.

Табличний вигляд результатів вибору інформації з бази даних - це інтуїтивно зрозумілий формат. Переходячи до розуміння формату блога наведемо приклад з професійних сайтів. Так на сайтах, що розробляються на основі CMS Joomla формат блогу зазвичай передбачає виведення заголовку матеріалу, анонсу матеріалу та кнопки “ДЕТАЛЬНІШЕ...” (рис.2)



Рис. 2. Приклад відображення даних у форматі блогу з сайту Університету Ушинського

Проаналізуємо, як сформувати шаблон типу блог для результатів SELECT запиту. З частини контролеру мають бути передані наступні дані:

- 1 - масив з результатом запиту **result**
- 2- масив з іменами атрибутів, дані по котрим будуть відображені для кожного кортежу **blog_atribut**
- 3- ім'я атрибуту, що є однозначною ідентифікацією кожного кортежа результату вибірки **pk**

Приклад шаблону блогу

```
{% extends 'base.tmpl' %}

{% block content %}
<ul>
{% for key, user_value in result %}
<!--на кожному кроці роботи циклу вибираємо один кортеж-масив-->
<li><ul>
{% for value1 in blog_atribut %}
<!--на кожному кроці роботи циклу вибираємо одне ім'я атрибуту, що має бути
виведено з кожного кортежу-масива-->
<li>
```

```

{{user_value[value1]}}//виводимо значення
</li>
{% endfor %}
<li>
<a href="{{x}}/?pk={{user_value[pk]}}">ДЕТАЛЬНІШЕ...</a>
<!--Формуємо кнопку “ДЕТАЛЬНІШЕ...”, її оброблення буде розглянуто в
наступній лабораторній роботі-->
</li>
</ul></li>
{% endfor %}
</ul>
</table>
{% endblock content %}

```

Даний шаблон відображає результат довільного запиту на вибірку інформації у форматі блогу за умови передачі саме перелічених вище змінних з частини контроллера

3) Приклад рішення задачі. Розглянемо рішення задачі на виведення результатів у форматі блогу.

Приклад файлу частини контроллера

```

<?php
require_once('config.php');

if(isset($_GET['pk']))
{
//Даний стан буде опрацьовано в наступній лабораторній роботі
$tpl_vars = array();
$tpl_file = 'lab11_1.tpl';
}
else
{
$dbh = db_connect();
$stmt = $dbh->query('SELECT * FROM news');
$result = $stmt->fetchAll(PDO::FETCH_ASSOC);
$blog_atribut = array('name','data');
$tpl_vars = array(
'result' => $result,
'x'=>$_SERVER['PHP_SELF'],
'blog_atribut'=>$blog_atribut,
'pk'=>'id');

$tpl_file = 'lab11.tpl';

```

```

}
require_once('tpl_config.php');
?>

```

Остаточний вигляд результату представлено на рисунку (Рис. 3)

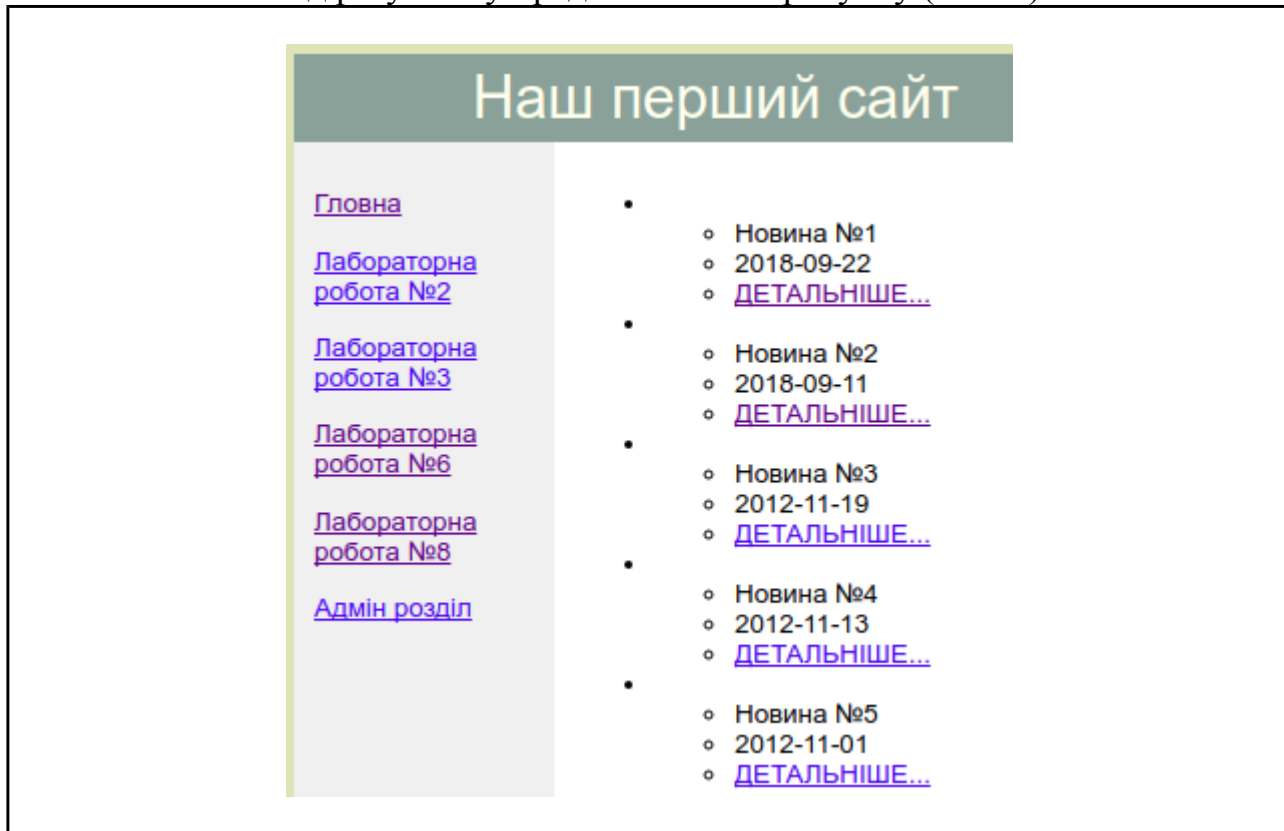


Рис. 3. Вигляд шаблону “Блог” для таблиці news

2) Індивідуальне завдання

- До бази даних індивідуального завдання десятої лабораторної роботи формуємо запити на вибірку інформації наступного характеру:

1. (Три запити). З включенням в запит всіх трьох таблиць сформувані три SQL запити SELECT, що реалізують використання відповідно конструкцій Left Join, Join, Right Join. Сформувані три запити, що надають різні результати (для отримання різних результатів виконати відповідні зміни в заповненні бази даних)

2. (Один запит). На основі любого одного запиту з попереднього підпункту (підпункт №1) сформувані 1 новий запит шляхом додавання обмеження Where, Order by. Сформулювати для даного запиту текстове пояснення, що відповідає на питання: “Що саме вибрано з бази даних предметної галузі?”

3. (Три запити). На основі любого одного запиту з підпункту №1 сформувані 3 нових запит шляхом використання 1- агрегатної функції, 2- агрегатної функції та конструкції Group by, 3-агрегатної функції та конструкції Having. Сформулювати для даних запитів текстове пояснення, що відповідає на питання: “Що саме вибрано з бази даних предметної галузі?”

4. Побудувати 7 сторінок інформаційної системи з використанням шаблону табличного виведення інформації.
5. Для одного запиту. Створити спрощений шаблон для виведення інформації в форматі блогу (Виводимо дані того атрибуту, ім'я якого передане в шаблон, формуємо кнопку “ДЕТАЛЬНІШЕ”. При натисненні на цю кнопку повна інформація обраного кортежу-результату поки що не відображається).

Список використаних джерел

1. Влад Мержевич. URL: <http://htmlbook.ru/> (дата звернення 23.11.2016)
2. Fabien Potencier. Шаблонизаторы в PHP. URL: <https://habrahabr.ru/post/75901/> (дата звернення 23.11.2016)
3. Twig: The flexible, fast, and secure template engine for PHP. URL: <http://twig.sensiolabs.org/> (дата звернення 23.11.2016)
4. Введение в MVC для интернет-разработок. URL: <http://bourabai.kz/dbt/mvc.htm> (дата звернення 23.11.2016)
5. Влад Мержевич. Пример 2. Макет из двух колонок. URL: <http://htmlbook.ru/layout/fiksirovannaya-shirina-navigatsiya-sleva> (дата звернення 23.11.2016)
6. Web-технологии. Основы языка PHP. URL: <https://htmlweb.ru/php/php2.php> (дата звернення 22.11.2016)
7. Абромьян М.Э. 10000 задач по программированию. Часть 1. URL: <http://www.ict.edu.ru/ft/004732/abr-1.pdf> (дата звернення 14.05.2018)
8. Довідник з MySQL. URL: <http://www.mysql.ru/docs/man/SELECT.html> (дата звернення 22.05.2019)
9. Мулеса О.Ю. Інформаційні системи та реляційні бази даних. Навч.посібник. – Електронне видання, 2018. – 118 с. URL: <https://dspace.uzhnu.edu.ua/jspui/handle/lib/19776> (дата звернення 22.05.2016)