

## Лабораторна робота №3

### Тема: Структурні шаблони

**Мета роботи:** навчитися реалізовувати структурні шаблони проєктування Адаптер, Декоратор, Міст, Компонувальник, Проксі, Легковаговик

### Хід роботи:

#### Завдання 0: Підготовка до виконання завдання

1. Створити директорію lab-3 в Вашому репозиторії з лабораторними.
2. Перейти до Завдання 1 😊

#### Завдання 1: Адаптер.

1. Створіть клас Logger, який буде мати методи `Log()`, `Error()`, `Warn()`, які виводять повідомлення в консоль різними кольорами (зеленим, червоним і оранжевим відповідно).
2. Створіть клас FileWriter з методами `Write()`, `WriteLine()`.
3. За допомогою шаблону Адаптер створіть файловий логер.
4. Покажіть правильність роботи свого коду запустивши його в головному методі програми

### Лістинг програми:

#### Logger.php:

```
<?php
class Logger
{
    public function log($message) {
        echo '<p style="color: green;">' . htmlspecialchars($message) . '</p>';
    }

    public function error($message) {
        echo '<p style="color: red;">' . htmlspecialchars($message) . '</p>';
    }

    public function warn($message) {
        echo '<p style="color: orange;">' . htmlspecialchars($message) . '</p>';
    }
}
```

					ДУ «Житомирська політехніка».22.121.16.000 – Лр3			
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи №3	Лім.	Арк.	Аркуші
Розроб.		Кохан Т.О						
Перевір.		Фант М. О.					1	18
Керівник						ФІКТ Гр. ІПЗ-22-З[1]		
Н. контр.								
Зав. каф.		Морозов А.В.						

## FileLogger.php:

```
<?php

class FileLogger
{
    private $logger;
    private $fileWriter;

    public function __construct(Logger $logger, FileWriter $fileWriter) {
        $this->logger = $logger;
        $this->fileWriter = $fileWriter;
    }

    public function log($message) {
        $this->logger->log($message);
        $this->fileWriter->writeLine("[LOG] " . $message);
    }

    public function error($message) {
        $this->logger->error($message);
        $this->fileWriter->writeLine("[ERROR] " . $message);
    }

    public function warn($message) {
        $this->logger->warn($message);
        $this->fileWriter->writeLine("[WARN] " . $message);
    }
}
```

## FileWriter.php:

```
<?php

class FileWriter
{
    private $fileHandle;

    public function __construct($filePath) {
        $this->fileHandle = fopen($filePath, 'a');
    }

    public function write($message) {
        fwrite($this->fileHandle, $message);
    }

    public function writeLine($message) {
        $this->write($message . PHP_EOL);
    }

    public function __destruct() {
        fclose($this->fileHandle);
    }
}
```

## index.php:

```
<?php
require_once 'Logger.php';
require_once 'FileWriter.php';
require_once 'FileLogger.php';
```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – ЛрЗ	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```
function main() {
    $logger = new Logger();
    $fileWriter = new FileWriter('log.txt');
    $fileLogger = new FileLogger($logger, $fileWriter);

    $fileLogger->log("This is a log message.");
    $fileLogger->error("This is an error message.");
    $fileLogger->warn("This is a warning message.");
}

main();
```

**log.txt:**

```
[LOG] This is a log message.
[ERROR] This is an error message.
[WARN] This is a warning message.
```

**Результат виконання програми:**

This is a log message.

This is an error message.

This is a warning message.

## Завдання 2: Декоратор.

1. Ви розробляєте РПГ гру. Створіть класи героїв Warrior, Mage, Palladin.
2. Для героїв створіть інвентар (одяг, зброю, артефакти), який може підходити будь-якому типу героїв, у вигляді декораторів.
3. Важливою вимогою є можливість використання декількох екземплярів інвентаря на герої одночасно.
4. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

**Лістинг програми:**

**heroes.php:**

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – ЛрЗ	Арк.
		Фант М. О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<?php

interface Hero {
    public function getDescription(): string;
}

class BaseHero implements Hero {
    protected $name;

    public function __construct($name) {
        $this->name = $name;
    }

    public function getDescription(): string {
        return $this->name;
    }
}

class Warrior extends BaseHero {
    public function __construct() {
        parent::__construct("Warrior");
    }
}

class Mage extends BaseHero {
    public function __construct() {
        parent::__construct("Mage");
    }
}

class Paladin extends BaseHero {
    public function __construct() {
        parent::__construct("Paladin");
    }
}

```

## inventory.php:

```

<?php

interface Inventory extends Hero {
}

abstract class InventoryDecorator implements Inventory {
    protected $hero;

    public function __construct(Hero $hero) {
        $this->hero = $hero;
    }

    public function getDescription(): string {
        return $this->hero->getDescription();
    }
}

class Weapon extends InventoryDecorator {
    public function getDescription(): string {
        return parent::getDescription() . ", Weapon";
    }
}

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

    }
}

class Armor extends InventoryDecorator {
    public function getDescription(): string {
        return parent::getDescription() . ", Armor";
    }
}

class Artifact extends InventoryDecorator {
    public function getDescription(): string {
        return parent::getDescription() . ", Artifact";
    }
}

```

### create\_hero.php:

```

<?php

require_once 'heroes.php';
require_once 'inventory.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    $heroType = $_POST["hero"];
    $inventory = $_POST["inventory"] ?? [];

    switch ($heroType) {
        case 'Warrior':
            $hero = new Warrior();
            break;
        case 'Mage':
            $hero = new Mage();
            break;
        case 'Paladin':
            $hero = new Paladin();
            break;
        default:
            echo "Invalid hero type";
            exit();
    }

    $inventoryInfo = implode(", ", $inventory);

    echo "<h2>Your Hero:</h2>";
    echo "<p>Type: " . $hero->getDescription() . "</p>";
    echo "<p>Inventory: " . $inventoryInfo . "</p>";
} else {

    echo "Form data not submitted";
}

```

### index.php:

```

<!doctype html>
<html lang="en">

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Фант М. О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-scale=1.0,
maximum-scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Main</title>
</head>
<body>
<h1>Create Your RPG Hero</h1>
<form method="post" action="create_hero.php">
  <label for="hero">Choose your hero:</label>
  <select name="hero" id="hero">
    <option value="Warrior">Warrior</option>
    <option value="Mage">Mage</option>
    <option value="Paladin">Paladin</option>
  </select>
  <br><br>
  <label>Choose your inventory:</label><br>
  <input type="checkbox" id="weapon" name="inventory[]" value="Weapon">
  <label for="weapon">Weapon</label>
  <input type="checkbox" id="armor" name="inventory[]" value="Armor">
  <label for="armor">Armor</label>
  <input type="checkbox" id="artifact" name="inventory[]" value="Artifact">
  <label for="artifact">Artifact</label>
  <br><br>
  <input type="submit" value="Create Hero">
</form>
</body>
</html>

```

**Результат виконання програми:**

## Create Your RPG Hero

Choose your hero:

Choose your inventory:

☒ Weapon ☒ Armor ☐ Artifact

## Your Hero:

Type: Mage

Inventory: Weapon, Armor

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

### Завдання 3: Міст.

1. Ви працюєте над графічним редактором. Створіть базовий клас Shape.

2. Створіть дочірні до Shape класи, Circle, Square, Triangle.
3. За допомогою шаблону Міст додайте можливість рендерингу кожної з фігур як векторної або растрової графіки (вивівши відповідне повідомлення у консоль, наприклад "Drawing Triangle as pixels").
4. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

#### Лістинг програми:

##### shape.php:

```
<?php
abstract class Shape {
    protected $renderer;

    public function __construct(Renderer $renderer) {
        $this->renderer = $renderer;
    }

    abstract public function draw();
}

class Circle extends Shape {
    public function draw() {
        $this->renderer->renderCircle();
    }
}

class Square extends Shape {
    public function draw() {
        $this->renderer->renderSquare();
    }
}

class Triangle extends Shape
{
    public function draw()
    {
        $this->renderer->renderTriangle();
    }
}
```

## renderer.php:

```
<?php
interface Renderer {
    public function renderCircle();
    public function renderSquare();
    public function renderTriangle();
}

class VectorRenderer implements Renderer {
    public function renderCircle() {
        echo "Drawing Circle as vector\n";
    }

    public function renderSquare() {
        echo "Drawing Square as vector\n";
    }

    public function renderTriangle() {
        echo "Drawing Triangle as vector\n";
    }
}

class RasterRenderer implements Renderer {
    public function renderCircle() {
        echo "Drawing Circle as pixels\n";
    }

    public function renderSquare() {
        echo "Drawing Square as pixels\n";
    }

    public function renderTriangle() {
        echo "Drawing Triangle as pixels\n";
    }
}
```

## index.php:

```
<?php
require_once 'shape.php';
require_once 'renderer.php';

$vectorRenderer = new VectorRenderer();
$rasterRenderer = new RasterRenderer();

$shapes = [
    new Circle($vectorRenderer),
    new Square($rasterRenderer),
    new Triangle($vectorRenderer)
];

foreach ($shapes as $shape) {
    $shape->draw();
    echo "<br>";
}
```

## Результат виконання програми:

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – ЛрЗ	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		8



# Drawing Circle as vector

## Drawing Square as pixels

## Drawing Triangle as vector

### Завдання 4: Проксі.

1. Створіть клас SmartTextReader, який вмє читати вміст текстового файлу і перетворювати його на двовірний масив якому зовнїшній масив вїдповїдає рядкам тексту, а вккладенї масиви вїдповїдають символам у вїдповїдному рядку.
2. Створїть проксі для SmartTextReader з логуванням SmartTextChecker, який буде виводити вїнформацїю про успїшне вїдкриття, прочитання і закриття файлу, а також буде виводити загальну кїлькїсть рядкїв і символїв у прочитаному тексті.
3. Створїть проксі для SmartTextReader з обмеженням доступу до певних файлїв SmartTextReaderLocker. Цей клас в конструкторї приймає регулярний вираз, по якому лїмітується доступ до певної групи файлїв. Якщо клїєнт викликатиме метод для прочитання такого лїмітованого файлу, замїсть прочитання файлу в консоль має виводитися повїдомлення "Access denied!".
4. Покажїть правильнїсть роботи свого коду запустивши його в головному методї програми.

### Лїстинг програми:

#### SmartTextReader.php:

```
<?php
class SmartTextReader {
    private $filename;

    public function __construct($filename) {
        $this->filename = $filename;
    }

    public function readText() {
        return file($this->filename);
    }
}
```

#### SmartTextChecker.php:

```
<?php
class SmartTextChecker {
    private $reader;

    public function __construct($filename) {
        $this->reader = new SmartTextReader($filename);
    }
}
```

		Кохан Т.О.			ДУ «Житомирська полїтехнїка».22.121.16.000 – ЛрЗ	Арк.
		Фант М. О.				9
Змн.	Арк.	№ докум.	Пїдпис	Дата		

```

        public function readText() {
            $lines = $this->reader->readText();
            echo "File opened and read successfully\n";
            echo "-----\n";
            echo "Total lines: " . count($lines) . "\n";
            $totalChars = 0;
            foreach ($lines as $line) {
                $totalChars += strlen($line);
            }
            echo "Total characters: $totalChars\n";
            echo "-----\n";
            return $lines;
        }
    }
}

```

## SmartTextReaderLocker.php:

```

<?php

class SmartTextReaderLocker {
    private $reader;
    private $allowedPattern;

    public function __construct($filename, $allowedPattern) {
        $this->reader = new SmartTextReader($filename);
        $this->allowedPattern = $allowedPattern;
    }

    public function readText() {
        $text = implode("", $this->reader->readText());
        if (preg_match($this->allowedPattern, $text)) {
            echo "Access denied!\n";
            echo "-----\n";
        } else {
            echo "File opened and read successfully\n";
            echo "-----\n";
            echo "Text content:\n";
            echo $text . "\n";
            echo "-----\n";
        }
    }
}

```

## index.php:

```

<?php
require_once 'SmartTextChecker.php';
require_once 'SmartTextReader.php';
require_once 'SmartTextReaderLocker.php';

echo "<pre>";

$reader = new SmartTextReader('example.txt');
echo "Перший тест - звичайне читання файлу:\n";
$text = $reader->readText();
print_r($text);
echo "\n";

$checker = new SmartTextChecker('example.txt');
echo "Другий тест - логуювання читання файлу:\n";

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – ЛрЗ	Арк.
		Фант М. О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```
$text = $checker->readText();
echo "\n";

$locker = new SmartTextReaderLocker('restricted_example.txt', '/restricted/');
echo "Третій тест - обмеження доступу до файлу за певним шаблоном:\n";
$text = $locker->readText();

echo "</pre>";
```

### example.txt:

```
Рядок для перевірки роботи коду.
Рядок для перевірки роботи коду.
Рядок для перевірки роботи коду.
```

### restricted\_example.php:

```
Цей файл містить обмежену інформацію.
```

### Результат виконання програми:

```
Перший тест - звичайне читання файлу:
Array
(
    [0] => Рядок для перевірки роботи коду.
    [1] => Рядок для перевірки роботи коду.
    [2] => Рядок для перевірки роботи коду.
)

Другий тест - логування читання файлу:
File opened and read successfully
-----
Total lines: 3
Total characters: 181
-----

Третій тест - обмеження доступу до файлу за певним шаблоном:
File opened and read successfully
-----
Text content:
Цей файл містить обмежену інформацію.
-----
```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Фант М. О.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

### Завдання 5: Компонувальник.

1. Вам потрібно створити власну мову розмітки LightHTML.
2. Кожен елемент розмітки має наслідувати клас LightNode.
3. Створіть два дочірніх класи від LightNode: LightElementNode, LightTextNode.
4. LightTextNode може містити лише текст.
5. LightElementNode може містити будь-які LightNode. LightElementNode повинен мати інформацію про назву тега, його тип відображення (блочний чи рядковий), тип закриття (одиничний тег, як `<img/>` чи з закриваючим тегом) список CSS класів,

кількість дочірніх елементів, а також має бути можливість виводити на екран його outerHTML і innerHTML.

6. За допомогою своєї мови розмітки виведіть в консоль елемент сторінки на Ваш вибір (наприклад якусь таблицю, список тощо).

7. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

### Лістинг програми:

#### LightHTML.php:

```
<?php

abstract class LightNode
{
    abstract public function getOuterHTML();

    abstract public function getInnerHTML();
}

class LightTextNode extends LightNode
{
    private $text;

    public function __construct($text)
    {
        $this->text = $text;
    }

    public function getOuterHTML()
    {
        return $this->text;
    }

    public function getInnerHTML()
    {
        return $this->text;
    }
}
```

Кохан Т.О.

Фант М. О.

№ докум.

Підпис

Дата

ДУ «Житомирська політехніка».22.121.16.000 – Лр3

Арк.

12

```

}

class LightElementNode extends LightNode
{
    private $tag;
    private $displayType;
    private $closingType;
    private $cssClasses = array();
    private $childNodes = array();

    public function __construct($tag, $displayType, $closingType, $cssClasses,
$childNodes)
    {
        $this->tag = $tag;
        $this->displayType = $displayType;
        $this->closingType = $closingType;
        $this->cssClasses = $cssClasses;
        $this->childNodes = $childNodes;
    }

    public function addChild($childNode)
    {
        $this->childNodes[] = $childNode;
    }

    public function getOuterHTML()
    {
        $html = "<{$this->tag}";

        if (!empty($this->cssClasses)) {
            $html .= ' class="' . implode(' ', $this->cssClasses) . '"';
        }

        if ($this->closingType === 'single') {
            $html .= '/>';
        } else {
            $html .= '>';
            foreach ($this->childNodes as $child) {
                $html .= $child->getOuterHTML();
            }
            $html .= "</{$this->tag}>";
        }

        return $html;
    }

    public function getInnerHTML()
    {
        $html = '';
        foreach ($this->childNodes as $child) {
            $html .= $child->getOuterHTML();
        }
        return $html;
    }
}

```

**index.php:**

```

<?php
require_once 'LightHTML.php';

$table = new LightElementNode('table', 'block', 'closing', array('my-table'),
array(
    new LightElementNode('tr', 'block', 'closing', array('row-class'), array(

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Фант М. О.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        new LightElementNode('td', 'block', 'closing', array('data-class'),
array(
            new LightTextNode('Row 1, Cell 1')
        )),
        new LightElementNode('td', 'block', 'closing', array('data-class'),
array(
            new LightTextNode('Row 1, Cell 2')
        ))
    )),
    new LightElementNode('tr', 'block', 'closing', array('row-class'), array(
        new LightElementNode('td', 'block', 'closing', array('data-class'),
array(
            new LightTextNode('Row 2, Cell 1')
        )),
        new LightElementNode('td', 'block', 'closing', array('data-class'),
array(
            new LightTextNode('Row 2, Cell 2')
        ))
    ))
));

echo "<!DOCTYPE html>\n";
echo "<html>\n";
echo "<head>\n";
echo '<link rel="stylesheet" type="text/css" href="style.css">' . "\n";
echo "</head>\n";
echo "<body>\n";
echo "<h2>Outer HTML:</h2>\n";
echo $table->getOuterHTML() . "\n\n";
echo "<h2>Inner HTML:</h2>\n";
echo $table->getInnerHTML() . "\n\n";
echo "</body>\n";
echo "</html>\n";

```

### style.css:

```

.my-table {
    border-collapse: collapse;
}

.my-table tr:nth-child(even) {
    background-color: lightgray;
}

.my-table tr:hover {
    background-color: lightgray;
}

.my-table td {
    border: 1px solid lightgray;
    padding: 8px;
}

```

### Результат виконання програми:

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – ЛрЗ	Арк.
		Фант М. О.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

## Outer HTML:

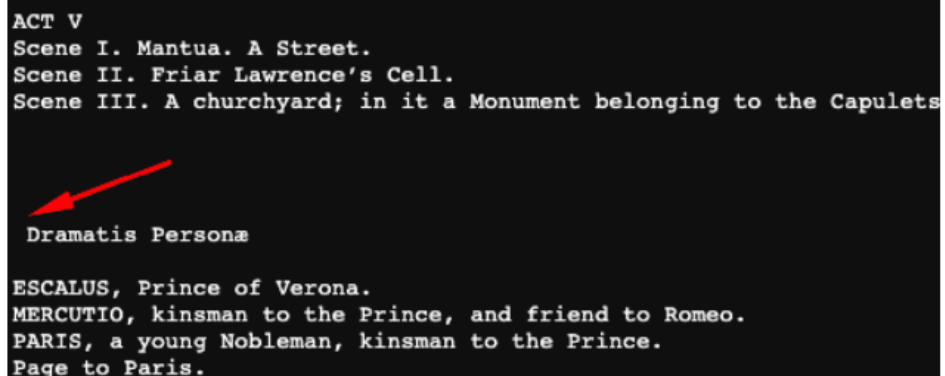
Row 1, Cell 1	Row 1, Cell 2
Row 2, Cell 1	Row 2, Cell 2

## Inner HTML:

Row 1, Cell 1Row 1, Cell 2Row 2, Cell 1Row 2, Cell 2

### Завдання 6: Легковаговик.

1. За допомогою свого LightHTML з завдання 1 перетворіть текст [книги](#) в HTML верстку за такими правилами:
  - a. Перший рядок має бути елементом `<h1>`
  - b. Якщо в рядку менше 20 символів - це має бути елемент `<h2>`
  - c. Якщо рядок починається з пробільного символу - це має бути `<blockquote>`.



```
ACT V
Scene I. Mantua. A Street.
Scene II. Friar Lawrence's Cell.
Scene III. A churchyard; in it a Monument belonging to the Capulets

Dramatis Personæ

ESCALUS, Prince of Verona.
MERCUTIO, kinsman to the Prince, and friend to Romeo.
PARIS, a young Nobleman, kinsman to the Prince.
Page to Paris.
```

- d. В будь-якому іншому випадку - елемент `<p>`
2. Покажіть скільки займає все дерево Вашої верстки, коли воно повністю утримується в пам'яті процесу.
  3. Використайте Легковаговик на Ваших класах HTML елементів, щоб зменшити споживання пам'яті.
  4. Покажіть правильність роботи свого коду запустивши його в головному методі програми.

### Лістинг програми:

#### LightHTML.php:

```
<?php

abstract class LightNode
{
    abstract public function getOuterHTML();

    abstract public function getInnerHTML();
}
```

```

class LightTextNode extends LightNode
{
    private $text;

    public function __construct($text)
    {
        $this->text = $text;
    }

    public function getOuterHTML()
    {
        return htmlspecialchars($this->text);
    }

    public function getInnerHTML()
    {
        return $this->text;
    }
}

class LightElementNode extends LightNode
{
    private $tag;
    private $displayType;
    private $closingType;
    private $cssClasses = array();
    private $childNodes = array();

    public function __construct($tag, $displayType, $closingType, $cssClasses,
$childNodes)
    {
        $this->tag = $tag;
        $this->displayType = $displayType;
        $this->closingType = $closingType;
        $this->cssClasses = $cssClasses;
        $this->childNodes = $childNodes;
    }

    public function addChild($childNode)
    {
        $this->childNodes[] = $childNode;
    }

    public function getOuterHTML()
    {
        $html = "<{$this->tag}";

        if (!empty($this->cssClasses)) {
            $html .= ' class="' . implode(' ', $this->cssClasses) . '"';
        }

        if ($this->closingType === 'single') {
            $html .= '/>';
        } else {
            $html .= '>';
            foreach ($this->childNodes as $child) {
                $html .= $child->getOuterHTML();
            }
            $html .= "</{$this->tag}>";
        }

        return $html;
    }
}

```

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Фант М. О.				16
Змн.	Арк.	№ докум.	Підпис	Дата		



```
public function getInnerHTML()
{
    $html = '';
    foreach ($this->childNodes as $child) {
        $html .= $child->getOuterHTML();
    }
    return $html;
}
}
```

**index.php:**

```
<?php
require_once 'LightHTML.php';

$bookText = file_get_contents("my_book.txt");
$lines = explode("\n", $bookText);

$htmlElements = [];
foreach ($lines as $index => $line) {
    if ($index === 0) {
        $element = new LightElementNode('h1', 'block', 'double', [], [new
LightTextNode($line)]);
    } elseif (strlen($line) < 20) {
        $element = new LightElementNode('h2', 'block', 'double', [], [new
LightTextNode($line)]);
    } elseif (preg_match('/^\s/', $line)) {
        $element = new LightElementNode('blockquote', 'block', 'double', [],
[new LightTextNode($line)]);
    } else {
        $element = new LightElementNode('p', 'block', 'double', [], [new
LightTextNode($line)]);
    }
    $htmlElements[] = $element;
}

$totalMemory = memory_get_usage();

foreach ($htmlElements as $element) {
    echo $element->getOuterHTML() . "\n";
}

echo "Обсяг пам'яті: " . (memory_get_usage() - $totalMemory) . " bytes\n";
```

**Результат виконання програми:**

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Фант М. О.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

# The Project Gutenberg eBook of Romeo and Juliet

This ebook is for the use of anyone anywhere in the United States and most other parts of the world at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this ebook or online at [www.gutenberg.org](http://www.gutenberg.org). If you are not located in the United States, you will have to check the laws of the country where you are located before using this eBook.

Title: Romeo and Juliet

Author: William Shakespeare

Release date: November 1, 1998 [eBook #1513]

Most recently updated: June 27, 2023

## Language: English

Credits: the PG Shakespeare Team, a team of about twenty Project Gutenberg volunteers

\*\*\* START OF THE PROJECT GUTENBERG EBOOK ROMEO AND JULIET \*\*\*

**Висновок:** під час виконання лабораторної роботи я навчилася реалізовувати структурні шаблони проектування Адаптер, Декоратор, Міст, Компонувальник, Проксі, Легковаговик.

		Кохан Т.О.			ДУ «Житомирська політехніка».22.121.16.000 – Лр3	Арк.
		Фант М. О.				
Змн.	Арк.	№ докум.	Підпис	Дата		18