

Лабораторна робота №4

Тема: Об'єктно-орієнтоване програмування в PHP

Мета роботи: навчитися працювати з класами

Завдання на лабораторну роботу:

Завдання 0. Ретельно вивчити теорію за посиланнями:

Завдання 1. (Організація класів по каталогах в проєкті)

- Створіть пустий проєкт PHP.
- Створіть каталоги: "Models", "Controllers", "Views".
- У кожному каталозі створіть по одному класу, наприклад, "UserModel", "UserController", "UIView".
- В кожному класі реалізуйте просту функціональність, наприклад, виведення повідомлення чи повернення значень.

Завдання 2. (Автопідключення класів за допомогою spl_autoload_register. PHPDoc)

- Додайте PHPDoc коментарі до всіх класів, вказавши їх призначення та властивості.
- Створіть файл **autoload.php**, який буде містити функцію для автопідключення класів.
- Використайте **spl_autoload_register** для автоматичного підключення класів на основі їхніх імен та розташування.

Завдання 3. (Неймспейси)

- Додайте неймспейси до класів у попередньому завданні. Наприклад, "namespace Models;" для "UserModel".
- Змініть файл **autoload.php** так, щоб він також враховував неймспейси при підключенні класів

| | | | | | | | | |
|-----------|------|-----------------|--------|------|---|--------------------|------|--------|
| | | | | | ДУ «Житомирська політехніка».24.121.03.00 – ЛрЗ | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | | | |
| Розроб. | | Білотіл Я. О. | | | Звіт з лабораторної роботи | Літ. | Арк. | Аркуші |
| Перевір. | | Дмитренко І. А. | | | | | 1 | 26 |
| Т. контр. | | | | | | ФІКТ Гр. ІПЗк-23-1 | | |
| Н. контр. | | | | | | | | |
| Затверд. | | | | | | | | |

Завдання 4. (Автопідключення класів з неймспейсами)

- Використовуйте аналогічний підхід до підключення класів, але тепер з урахуванням неймспейсів.
- Переконайтеся, що класи виводять повідомлення чи результати виклику.

Код програми:

index.php

```
<?php

require_once 'autoload.php';

use Controllers\UserController;
use Views\UserView;

// Створюємо екземпляри класів
$controller = new UserController();
$view = new UserView();

// Отримуємо ім'я користувача через контролер
$username = $controller->showUserName();

// Відображаємо ім'я користувача через view
$view->display($username);

// Отримуємо повідомлення з даними користувача через контролер
$userMessage = $controller->showUserMessage();

// Відображаємо повідомлення з даними користувача через view
$view->display($userMessage);

?>
```

autoload.php

```
<?php

/**
 * Автозавантаження класів на основі їх простору імен і розташування файлів.
 *
 * @param string $className Повне ім'я класу з простором імен.
 */
```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 2 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```
function myAutoload($className)
{
    // Заміна зворотних слешів у просторі імен на слеші для файлової системи.
    $file = __DIR__ . '/' . str_replace('\\', '/', $className) . '.php';

    // Перевірка існування файлу перед підключенням.
    if (file_exists($file)) {
        require $file;
    } else {
        echo "Файл $file не знайдено.";
    }
}

spl_autoload_register('myAutoload');
?>
```

UserModel.php

```
<?php

namespace Models;

/**
 * Class UserModel
 *
 * Модель для роботи з даними користувача.
 */
class UserModel
{
    /**
     * Отримує ім'я користувача.
     *
     * @return string Ім'я користувача.
     */
    public function getUsername()
    {
        return "John Doe";
    }

    /**
     * Отримує повідомлення з даними користувача.
     *
     * @return string Повідомлення з ім'ям користувача.
     */
    public function getUserMessage()
    {
        return "UserModel: User name is Yaroslav.";
    }
}
```

| | | | | | | |
|------|------|----------------|--------|------|---|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка». 24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 3 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```
}  
?>
```

UserController.php

```
<?php  
  
namespace Controllers;  
  
use Models\UserModel;  
  
/**  
 * Class UserController  
 *  
 * Контролер для управління даними користувача.  
 */  
class UserController  
{  
    /**  
     * @var UserModel  
     */  
    private $model;  
  
    /**  
     * UserController конструктор.  
     */  
    public function __construct()  
    {  
        $this->model = new UserModel();  
    }  
  
    /**  
     * Отримує ім'я користувача з моделі.  
     *  
     * @return string Ім'я користувача.  
     */  
    public function showUserName()  
    {  
        return $this->model->getUserName();  
    }  
  
    /**  
     * Отримує повідомлення з даними користувача з моделі.  
     *  
     * @return string Повідомлення з ім'ям користувача.  
     */  
    public function showUserMessage()  
    {  
        return $this->model->getUserMessage();  
    }  
}
```

| | | | | | | |
|------|------|----------------|--------|------|---|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка». 24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 4 |

```
}
}
?>
```

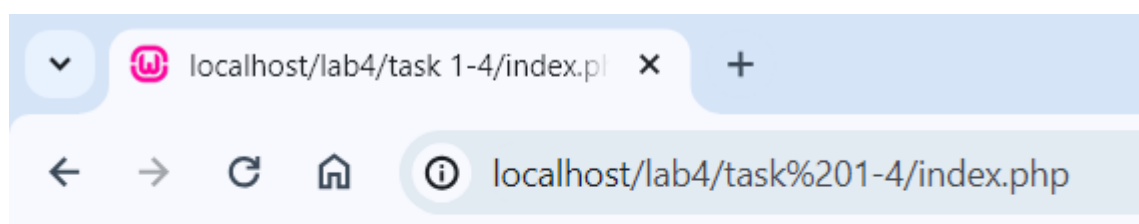
UserView.php

```
<?php

namespace Views;

/**
 * Class UserView
 *
 * Відповідає за відображення даних користувача.
 */
class UserView
{
    /**
     * Відображає дані користувача.
     *
     * @param string $data Дані для відображення.
     */
    public function display($data)
    {
        echo "UserView: $data <br>";
    }
}

?>
```



UserView: John Doe
UserView: UserModel: User name is Yaroslav.

Рис. 1 – Завдання 1-4

Завдання 5 (Створення класу. Методи GET і SET)

- 1) Створіть клас **Circle** з полями: координати центру і радіус кола
- 2) Створіть конструктор, що приймає значення для 3-х полів

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 5 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

- 3) Створіть метод **__toString()**, що повертає рядок в форматі: «Коло з центром в (x, y) і радіусом radius»
- 4) Створіть методи **GET** і **SET** для всіх 3-х полів
- 5) Створіть об'єкт та перевірте всі його методи

Код програми:

Circle.php

```
<?php

/**
 * Class Circle
 *
 * Представляє коло з заданим центром і радіусом.
 */
class Circle
{
    /**
     * @var float $x Координата X центру кола.
     * @var float $y Координата Y центру кола.
     * @var float $radius Радіус кола.
     */
    private $x;
    private $y;
    private $radius;

    /**
     * Circle конструктор.
     *
     * @param float $x Координата X центру кола.
     * @param float $y Координата Y центру кола.
     * @param float $radius Радіус кола.
     */
    public function __construct($x, $y, $radius)
    {
        $this->x = $x;
        $this->y = $y;
        $this->radius = $radius;
    }

    /**
     * Метод __toString()
     *
     * Повертає рядок в форматі: «Коло з центром в (x, y) і радіусом radius».
     *
     * @return string
     */
    public function __toString()
```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 6 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

    {
        return "Коло з центром в ({ $this->x}, { $this->y}) і радіусом { $this->radius}";
    }

    // Методи GET
    public function getX()
    {
        return $this->x;
    }

    public function getY()
    {
        return $this->y;
    }

    public function getRadius()
    {
        return $this->radius;
    }

    // Методи SET
    public function setX($x)
    {
        $this->x = $x;
    }

    public function setY($y)
    {
        $this->y = $y;
    }

    public function setRadius($radius)
    {
        $this->radius = $radius;
    }
}

?>

```

index.php

```

<?php

require_once 'autoload.php';

```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 7 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```
// Створення об'єкта класу Circle
$circle = new Circle(5.5, 10.2, 15.3);

// Перевірка методу __toString()
echo $circle . "<br>";

// Перевірка методів GET
echo "Координата X: " . $circle->getX() . "<br>";
echo "Координата Y: " . $circle->getY() . "<br>";
echo "Радіус: " . $circle->getRadius() . "<br>";

// Перевірка методів SET
$circle->setX(7.7);
$circle->setY(12.4);
$circle->setRadius(20.5);

// Перевірка методів GET після зміни значень
echo "Змінена координата X: " . $circle->getX() . "<br>";
echo "Змінена координата Y: " . $circle->getY() . "<br>";
echo "Змінений радіус: " . $circle->getRadius() . "<br>";

// Перевірка методу __toString() після зміни значень
echo $circle . "<br>";

?>
```

autoload.php

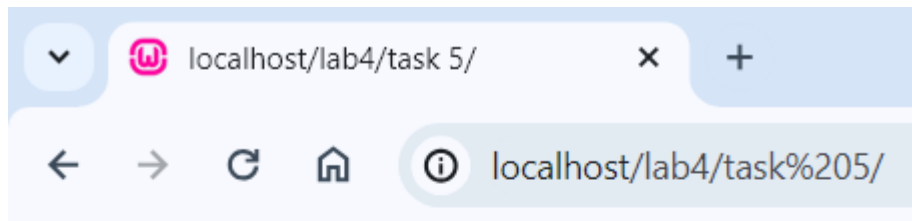
```
<?php

/**
 * Автозавантаження класів на основі їх простору імен і розташування файлів.
 *
 * @param string $className Повне ім'я класу.
 */
function myAutoload($className)
{
    // Заміна зворотних слешів у просторі імен на слеші для файлової системи.
    $file = __DIR__ . '/' . str_replace('\\', '/', $className) . '.php';

    // Перевірка існування файлу перед підключенням.
    if (file_exists($file)) {
        require $file;
    } else {
        echo "Файл $file не знайдено.<br>";
    }
}
```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 8 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |


```
spl_autoload_register('myAutoload');
?>
```



Коло з центром в (5.5, 10.2) і радіусом 15.3
 Координата X: 5.5
 Координата Y: 10.2
 Радіус: 15.3
 Змінена координата X: 7.7
 Змінена координата Y: 12.4
 Змінений радіус: 20.5
 Коло з центром в (7.7, 12.4) і радіусом 20.5

Рис. 2 – Завдання 5

Завдання 6 (Модифікатори доступу)

- 1) В класі з попереднього завдання зробіть всі поля `private`.
- 2) Створіть метод, що приймає об'єкт кола, і повертає **true**, якщо дані кола перетинаються, і **false**, якщо вони не перетинаються.

Код програми:

Circle.php

```
<?php

/**
 * Class Circle
 *
 * Представляє коло з заданим центром і радіусом.
 */
class Circle
{
    /**
     * @var float $x Координата X центру кола.
     * @var float $y Координата Y центру кола.
     * @var float $radius Радіус кола.
```

```

    */
private $x;
private $y;
private $radius;

/**
 * Circle конструктор.
 *
 * @param float $x Координата X центру кола.
 * @param float $y Координата Y центру кола.
 * @param float $radius Радіус кола.
 */
public function __construct($x, $y, $radius)
{
    $this->x = $x;
    $this->y = $y;
    $this->radius = $radius;
}

/**
 * Метод __toString()
 *
 * Повертає рядок в форматі: «Коло з центром в (x, y) і радіусом radius».
 *
 * @return string
 */
public function __toString()
{
    return "Коло з центром в ({ $this->x}, { $this->y}) і радіусом { $this->radius}";
}

// Методи GET
public function getX()
{
    return $this->x;
}

public function getY()
{
    return $this->y;
}

public function getRadius()
{
    return $this->radius;
}

// Методи SET

```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 10 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

public function setX($x)
{
    $this->x = $x;
}

public function setY($y)
{
    $this->y = $y;
}

public function setRadius($radius)
{
    $this->radius = $radius;
}

/**
 * Метод intersects
 *
 * Перевіряє, чи перетинається це коло з іншим колом.
 *
 * @param Circle $circle Інший об'єкт кола для перевірки.
 * @return bool Повертає true, якщо кола перетинаються, і false, якщо ні.
 */
public function intersects(Circle $circle)
{
    $distance = sqrt(pow($this->x - $circle->getX(), 2) + pow($this->y -
$circle->getY(), 2));
    return $distance <= ($this->radius + $circle->getRadius());
}
}

?>

```

Index.php

```

<?php

require_once 'autoload.php';

// Створення об'єктів класу Circle
$circle1 = new Circle(100, 150, 50);
$circle2 = new Circle(200, 150, 75);

// Перевірка методу __toString()
echo $circle1 . "<br>";
echo $circle2 . "<br>";

// Перевірка перетину кіл

```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 11 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

$intersects = $circle1->intersects($circle2) ? "Кола перетинаються." : "Кола не
перетинаються.";
echo $intersects . "<br>";

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Кола</title>
    <style>
        canvas {
            border: 1px solid black;
        }
    </style>
</head>
<body>

<canvas id="myCanvas" width="500" height="400"></canvas>

<script>
    function drawCircle(ctx, x, y, radius, color) {
        ctx.beginPath();
        ctx.arc(x, y, radius, 0, 2 * Math.PI);
        ctx.fillStyle = color;
        ctx.fill();
        ctx.stroke();
    }

    function drawAxes(ctx) {
        ctx.strokeStyle = 'black';
        ctx.lineWidth = 2;

        ctx.beginPath();
        ctx.moveTo(0, 200);
        ctx.lineTo(500, 200);
        ctx.stroke();

        ctx.beginPath();
        ctx.moveTo(250, 0);
        ctx.lineTo(250, 400);
        ctx.stroke();

        // Позначки на осі X
        for (let x = -200; x <= 200; x += 50) {
            ctx.beginPath();
            ctx.moveTo(250 + x, 200 - 5);

```

| | | | | | | |
|------|------|----------------|--------|------|---|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка». 24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 12 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        ctx.lineTo(250 + x, 200 + 5);
        ctx.stroke();
        ctx.fillText(x, 250 + x - 5, 200 + 15);
    }

    // Позначки на осі Y
    for (let y = -200; y <= 200; y += 50) {
        ctx.beginPath();
        ctx.moveTo(250 - 5, 200 - y);
        ctx.lineTo(250 + 5, 200 - y);
        ctx.stroke();
        ctx.fillText(y, 250 - 25, 200 - y + 5);
    }
}

const canvas = document.getElementById('myCanvas');
const ctx = canvas.getContext('2d');

// Малюємо систему координат
drawAxes(ctx);

// Координати та радіуси кіл
const circle1 = { x: <?php echo $circle1->getX() + 250; ?>, y: <?php echo 200
- $circle1->getY(); ?>, radius: <?php echo $circle1->getRadius(); ?> };
const circle2 = { x: <?php echo $circle2->getX() + 250; ?>, y: <?php echo 200
- $circle2->getY(); ?>, radius: <?php echo $circle2->getRadius(); ?> };

// Малюємо кола
drawCircle(ctx, circle1.x, circle1.y, circle1.radius, 'rgba(255, 0, 0, 0.5)');
drawCircle(ctx, circle2.x, circle2.y, circle2.radius, 'rgba(0, 0, 255, 0.5)');
</script>

</body>
</html>

```

Autoload.php

```

<?php

/**
 * Автозавантаження класів на основі їх простору імен і розташування файлів.
 *
 * @param string $className Повне ім'я класу.
 */
function myAutoload($className)
{
    // Заміна зворотних слешів у просторі імен на слеші для файлової системи.
    $file = __DIR__ . '/' . str_replace('\\', '/', $className) . '.php';
}

```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 13 |

```
// Перевірка існування файлу перед підключенням.
if (file_exists($file)) {
    require $file;
} else {
    echo "Файл $file не знайдено.<br>";
}

spl_autoload_register('myAutoload');
?>
```

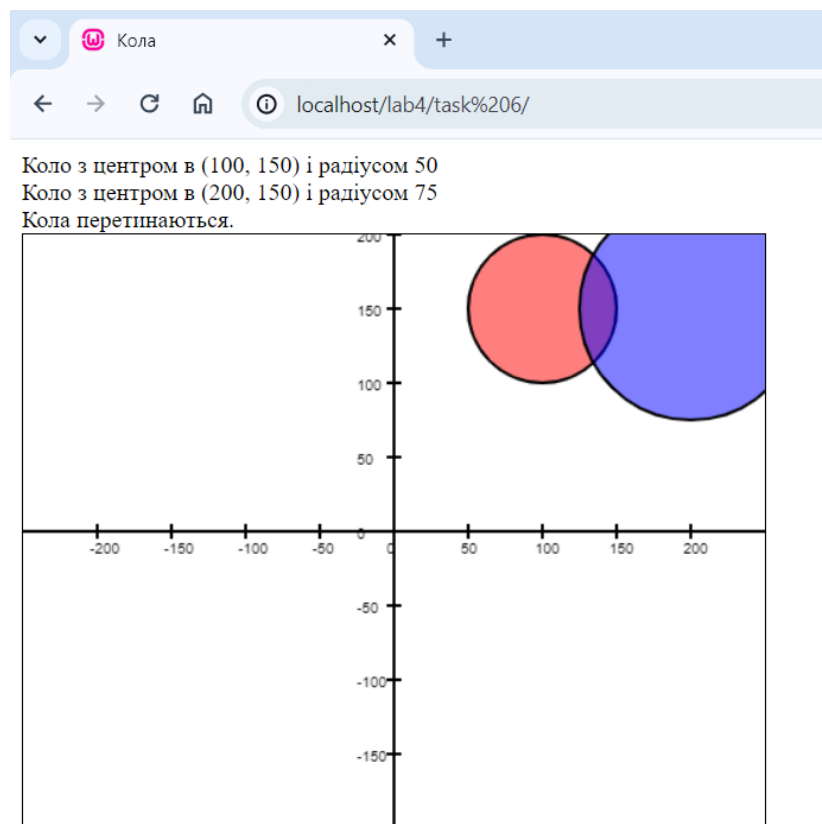


Рис. 3 – Завдання 6

Завдання 7 (Статичні властивості і методи)

- 1) Створіть директорію **text**, а в ній 3 текстових файла
- 2) Створіть клас зі статичним полем **dir="text"**
- 3) Створіть 2 статичних методи в класі: на читання та запис в файл:
 - Ім'я файлу передається як параметр метода.
 - В метод «на запис в файл» передається ще й рядок, який потрібно дописати в файл.

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 14 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

- Директорія береться зі статичного поля

4) Створіть метод, що дозволяє стерти вміст файлу

Перевірте роботу всіх методів

Код програми:

Index.php

```
<?php

require_once 'autoload.php';

// Записуємо рядок в file1.txt
FileManager::writeFile('file1.txt', "Це перший рядок у file1.txt.\n");

// Читаємо вміст file1.txt
$content = FileManager::readFile('file1.txt');
echo "Вміст file1.txt:<br>";
echo nl2br($content) . "<br>";

// Допишуємо ще один рядок в file1.txt
FileManager::writeFile('file1.txt', "Це другий рядок у file1.txt.\n");

// Читаємо знову вміст file1.txt
$content = FileManager::readFile('file1.txt');
echo "Оновлений вміст file1.txt:<br>";
echo nl2br($content) . "<br>";

// Стираємо вміст file1.txt
FileManager::clearFile('file1.txt');

// Читаємо вміст file1.txt після стирання
$content = FileManager::readFile('file1.txt');
echo "Вміст file1.txt після стирання:<br>";
echo nl2br($content) . "<br>";

?>
```

FileManager.php

```
<?php

/**
 * Class FileManager
 *
 * Управління файлами в заданій директорії.
 */
class FileManager
```

| | | | | | | |
|------|------|----------------|--------|------|---|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка». 24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 15 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

{
    /**
     * @var string $dir Директорія для роботи з файлами.
     */
    public static $dir = 'text';

    /**
     * Статичний метод для читання файлу.
     *
     * @param string $filename Ім'я файлу.
     * @return string Вміст файлу.
     */
    public static function readFile($filename)
    {
        $filepath = self::$dir . '/' . $filename;
        if (file_exists($filepath)) {
            return file_get_contents($filepath);
        } else {
            return "Файл не знайдено.";
        }
    }

    /**
     * Статичний метод для запису у файл.
     *
     * @param string $filename Ім'я файлу.
     * @param string $content Рядок, який потрібно дописати в файл.
     */
    public static function writeFile($filename, $content)
    {
        $filepath = self::$dir . '/' . $filename;
        file_put_contents($filepath, $content, FILE_APPEND);
    }

    /**
     * Статичний метод для стирання вмісту файлу.
     *
     * @param string $filename Ім'я файлу.
     */
    public static function clearFile($filename)
    {
        $filepath = self::$dir . '/' . $filename;
        if (file_exists($filepath)) {
            file_put_contents($filepath, '');
        } else {
            echo "Файл не знайдено.<br>";
        }
    }
}

```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 16 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |


```
?>
```

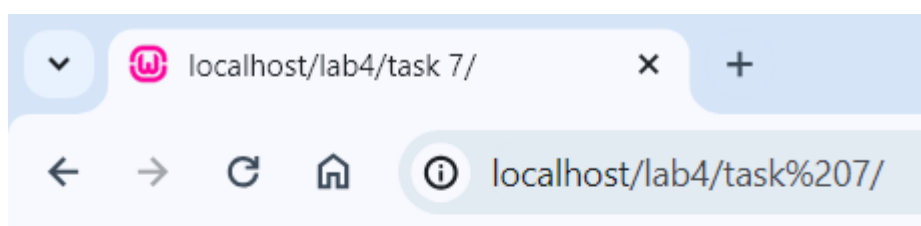
autoload.php

```
<?php

/**
 * Автозавантаження класів на основі їх простору імен і розташування файлів.
 *
 * @param string $className Повне ім'я класу.
 */
function myAutoload($className)
{
    // Заміна зворотних слешів у просторі імен на слеші для файлової системи.
    $file = __DIR__ . '/' . str_replace('\\', '/', $className) . '.php';

    // Перевірка існування файлу перед підключенням.
    if (file_exists($file)) {
        require $file;
    } else {
        echo "Файл $file не знайдено.<br>";
    }
}

spl_autoload_register('myAutoload');
?>
```



Вміст file1.txt:
Це перший рядок у file1.txt.

Оновлений вміст file1.txt:
Це перший рядок у file1.txt.
Це другий рядок у file1.txt.

Вміст file1.txt після стирання:

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 17 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

Завдання 8 (Наслідування)

- 1) Створіть клас **Human** з властивостями, що характеризують людину (зріст, маса, вік...). Створіть методи **GET** і **SET** для кожної властивості
- 2) Створіть клас **Student**, який успадковуватиметься від класу **Human**:
 1. Додайте властивості, специфічні тільки для студента (назва ВНЗ, курс...)
 2. Додайте в клас методи **GET** і **SET** для всіх нових властивостей.
 3. Реалізуйте метод, який буде переводити студента на новий курс (тобто просто збільшувати значення поля «курс» на 1)
- 3) Створіть клас **Programmer**, який успадковуватиметься від класу **Human**:
 - Додайте властивості, специфічні тільки для програміста (масив з мовами програмування, які він знає, досвід роботи...).
 - Додайте в клас методи **GET** і **SET** для всіх нових властивостей.
 - Реалізуйте метод, який буде додавати в масив з мовами ще одну мову.

Перевірте роботу всіх класів і всіх методів. Не забудьте змінити зріст і масу у студентів і програмістів, скориставшись методами з батьківського класу **Human**

Завдання 9 (Абстрактні класи)

- 1) Зробіть клас **Human** абстрактним.
- 2) Напишіть метод «**Народження дитини**» в класі **Human**, що викликає метод «**Повідомлення при народженні дитини**» (не забудьте поставити модифікатор **protected**), який буде абстрактним
- 3) Реалізуйте «**Повідомлення при народженні дитини**» у класів **Student** та **Programmer**

Перевірте роботу методів «народження»

Код програми:

Index.php

```
<?php

// Клас Human
class Human {
    private $height;
    private $weight;
    private $age;
```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 18 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

public function __construct($height, $weight, $age) {
    $this->height = $height;
    $this->weight = $weight;
    $this->age = $age;
}

// Методи GET
public function getHeight() {
    return $this->height;
}

public function getWeight() {
    return $this->weight;
}

public function getAge() {
    return $this->age;
}

// Методи SET
public function setHeight($height) {
    $this->height = $height;
}

public function setWeight($weight) {
    $this->weight = $weight;
}

public function setAge($age) {
    $this->age = $age;
}
}

// Клас Student, який успадковується від класу Human
class Student extends Human {
    private $university;
    private $course;

    public function __construct($height, $weight, $age, $university, $course) {
        parent::__construct($height, $weight, $age);
        $this->university = $university;
        $this->course = $course;
    }

    // Методи GET і SET для властивостей студента
    public function getUniversity() {
        return $this->university;
    }
}

```

| | | | | | | |
|------|------|----------------|--------|------|---|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка». 24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 19 |

```

    public function setUniversity($university) {
        $this->university = $university;
    }

    public function getCourse() {
        return $this->course;
    }

    public function setCourse($course) {
        $this->course = $course;
    }

    // Метод для перевodu студента на новий курс
    public function moveToNextCourse() {
        $this->course++;
    }
}

// Клас Programmer, який успадковується від класу Human
class Programmer extends Human {
    private $programmingLanguages = [];
    private $experience;

    public function __construct($height, $weight, $age, $programmingLanguages,
    $experience) {
        parent::__construct($height, $weight, $age);
        $this->programmingLanguages = $programmingLanguages;
        $this->experience = $experience;
    }

    // Методи GET і SET для властивостей програміста
    public function getProgrammingLanguages() {
        return $this->programmingLanguages;
    }

    public function setProgrammingLanguages($programmingLanguages) {
        $this->programmingLanguages = $programmingLanguages;
    }

    public function getExperience() {
        return $this->experience;
    }

    public function setExperience($experience) {
        $this->experience = $experience;
    }

    // Метод для додавання нової мови програмування

```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 20 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        public function addProgrammingLanguage($language) {
            $this->programmingLanguages[] = $language;
        }
    }

    // Перевірка роботи всіх класів і методів

    // Створення об'єктів класу Student
    $student = new Student(175, 70, 20, "Harvard", 2);
    echo "Студент: Зріст - " . $student->getHeight() . ", Маца - " . $student->
    >getWeight() . ", Вік - " . $student->getAge() . ", Університет - " . $student->
    >getUniversity() . ", Курс - " . $student->getCourse() . "<br>";

    // Переведення студента на новий курс і перевірка
    $student->moveToNextCourse();
    echo "Студент після переведення на новий курс: Курс - " . $student->getCourse() .
    "<br>";

    // Створення об'єктів класу Programmer
    $programmer = new Programmer(180, 80, 25, ["PHP", "JavaScript"], "3 роки");
    echo "Програміст: Зріст - " . $programmer->getHeight() . ", Маца - " .
    $programmer->getWeight() . ", Вік - " . $programmer->getAge() . ", Мови
    програмування - " . implode(", ", $programmer->getProgrammingLanguages()) . ",
    Досвід роботи - " . $programmer->getExperience() . "<br>";

    // Додавання нової мови програмування програмісту
    $programmer->addProgrammingLanguage("Python");
    echo "Програміст після додавання нової мови програмування: Мови програмування - "
    . implode(", ", $programmer->getProgrammingLanguages()) . "<br>";

    ?>

```

Index2.php

```

<?php

// Абстрактний клас Human
abstract class Human {
    protected $height;
    protected $weight;
    protected $age;

    // Абстрактний метод "Повідомлення при народженні дитини"
    abstract protected function birthMessage();

    public function __construct($height, $weight, $age) {
        $this->height = $height;
        $this->weight = $weight;
    }
}

```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 21 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        $this->age = $age;
    }

    // Методи GET
    public function getHeight() {
        return $this->height;
    }

    public function getWeight() {
        return $this->weight;
    }

    public function getAge() {
        return $this->age;
    }

    // Метод "Народження дитини"
    public function giveBirth() {
        $this->birthMessage();
    }
}

// Клас Student, який успадковується від класу Human
class Student extends Human {
    private $university;
    private $course;

    public function __construct($height, $weight, $age, $university, $course) {
        parent::__construct($height, $weight, $age);
        $this->university = $university;
        $this->course = $course;
    }

    // Реалізація абстрактного методу "Повідомлення при народженні дитини" для
Student
    protected function birthMessage() {
        echo "Студент народив дитину!";
    }
}

// Клас Programmer, який успадковується від класу Human
class Programmer extends Human {
    private $programmingLanguages = [];
    private $experience;

    public function __construct($height, $weight, $age, $programmingLanguages,
$experience) {
        parent::__construct($height, $weight, $age);
        $this->programmingLanguages = $programmingLanguages;
    }
}

```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 22 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

        $this->experience = $experience;
    }

    // Реалізація абстрактного методу "Повідомлення при народженні дитини" для
    Programmer
    protected function birthMessage() {
        echo "Програміст народив дитину!";
    }
}

// Перевірка роботи методу "Народження дитини"

// Студент народжує дитину
$student = new Student(175, 70, 20, "Harvard", 2);
$student->giveBirth(); // Виведе: "Студент народив дитину!"

echo "<br>";

// Програміст народжує дитину
$programmer = new Programmer(180, 80, 25, ["PHP", "JavaScript"], "3 роки");
$programmer->giveBirth(); // Виведе: "Програміст народив дитину!"

?>

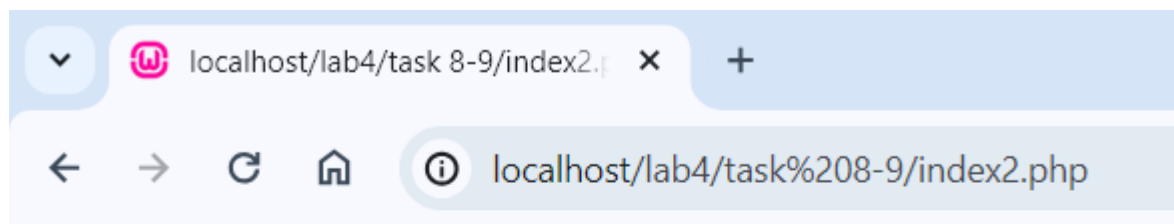
```

localhost/lab4/task 8-9/

localhost/lab4/task%208-9/

Студент: Зріст - 175, Маса - 70, Вік - 20, Університет - Harvard, Курс - 2
 Студент після переведення на новий курс: Курс - 3
 Програміст: Зріст - 180, Маса - 80, Вік - 25, Мови програмування - PHP, JavaScript, Досвід роботи - 3 роки
 Програміст після додавання нової мови програмування: Мови програмування - PHP, JavaScript, Python

| | | | | | | |
|------|------|----------------|--------|------|---|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка». 24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 23 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |



Студент народив дитину!
Програміст народив дитину!

Рис. 5-6 – Завдання 8-9

Завдання 10 (Інтерфейси)

- 1) Створіть інтерфейс «**Прибирання будинку**», в якому опишіть 2 методи: «**Прибирання кімнати**» і «**Прибирання кухні**»
- 2) Додайте створений інтерфейс в клас **Human**
- 3) Реалізуйте у кожному класі-спадкоємці (**Student** та **Programmer**) обидва методи
- 4) Реалізація повинна бути у вигляді одного з рядків: «**Студент прибирає кімнату**», «**Студент прибирає кухню**», «**Програміст прибирає кімнату**», «**Програміст прибирає кухню**»,
- 5) Перевірте роботу методів прибирання в обох класах

Код програми:

```
<?php

// Інтерфейс "Прибирання будинку"
interface HouseCleaning {
    public function cleanRoom();
    public function cleanKitchen();
}

// Клас Human, що реалізує інтерфейс "Прибирання будинку"
class Human implements HouseCleaning {
    public function cleanRoom() {
        echo "Людина прибирає кімнату<br>";
    }
}
```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 24 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |


```

    }

    public function cleanKitchen() {
        echo "Людина прибирає кухню<br>";
    }
}

// Клас Student, який успадковується від класу Human та реалізує інтерфейс
"Прибирання будинку"
class Student extends Human {
    // Реалізація методів інтерфейсу "Прибирання будинку"
    public function cleanRoom() {
        echo "Студент прибирає кімнату<br>";
    }

    public function cleanKitchen() {
        echo "Студент прибирає кухню<br>";
    }
}

// Клас Programmer, який успадковується від класу Human та реалізує інтерфейс
"Прибирання будинку"
class Programmer extends Human {
    // Реалізація методів інтерфейсу "Прибирання будинку"
    public function cleanRoom() {
        echo "Програміст прибирає кімнату<br>";
    }

    public function cleanKitchen() {
        echo "Програміст прибирає кухню<br>";
    }
}

// Перевірка роботи методів прибирання в обох класах

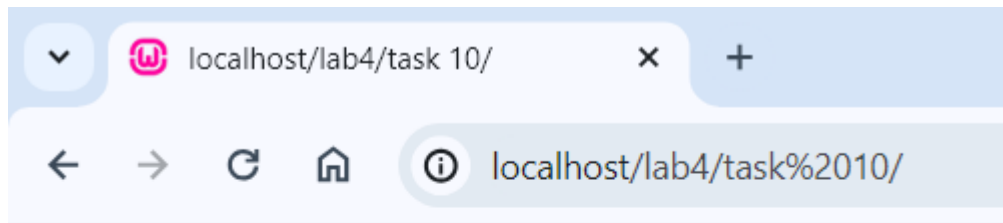
$student = new Student();
$student->cleanRoom(); // Виведе: "Студент прибирає кімнату"
$student->cleanKitchen(); // Виведе: "Студент прибирає кухню"

$programmer = new Programmer();
$programmer->cleanRoom(); // Виведе: "Програміст прибирає кімнату"
$programmer->cleanKitchen(); // Виведе: "Програміст прибирає кухню"

?>

```

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | 25 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |



Студент прибирає кімнату
 Студент прибирає кухню
 Програміст прибирає кімнату
 Програміст прибирає кухню

Рис. 7 – Завдання 10

| | | | | | | |
|------|------|----------------|--------|------|--|------|
| | | Білотіл Я.О. | | | ДУ «Житомирська політехніка».24.121.03.000 – Лр4 | Арк. |
| | | Дмитренко І.А. | | | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | 26 |