

## ЛАБОРАТОРНА РОБОТА № 4

**Тема:** Об'єктно-орієнтоване програмування в PHP

**Мета роботи:** навчитися працювати з класами

### Завдання на лабораторну роботу:

**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", "UserView".
- В кожному класі реалізуйте просту функціональність, наприклад, виведення повідомлення чи повернення значень.

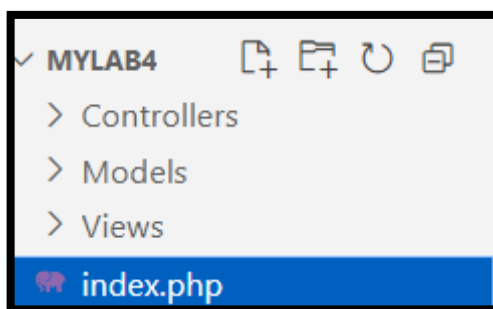


Рис. 1. Створені каталоги

Створено 3 файли: UserController, UserModel, UserViews.

Лістинг UserController:

```
<?php
require_once "Models/UserModel.php";
require_once "Views/UserView.php";

use Models\UserModel;
use Views\UserView;
```

					Житомирська політехніка.00.000.00.000 – Лр4				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Сапацінський Б.І.			Звіт з лабораторної роботи	Літ.	Арк.	Аркушів	
Перевір.		Ковтун В.В.					1	13	
Керівник						ФІКТ Гр. ІПЗк-24-1			
Н. контр.									
Зав. каф.									

```

class UserController {
    private UserModel $model;
    private UserView $view;

    public function __construct() {
        $this->model = new UserModel();
        $this->view = new UserView();
    }

    public function showUser() {
        $name = $this->model->getName();
        $this->view->render($name);
    }
}

```

Лістинг UserModel:

```

<?php
class UserModel {
    private string $name;

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

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

```

Лістинг UserViews:

```

class UserView {
    public function render(string $name) {
        echo "Hello $name <br>";
    }
}

```

1 Завдання  
Hello Новий користувач

Рис. 2. Результат

		Сапацінський Б.І.			Житомирська політехніка.00.000.00.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

**Завдання 2.** (Автопідключення класів за допомогою spl\_autoload\_register.  
PHPDoc)

- Додайте PHPDoc коментарі до всіх класів, вказавши їх призначення та властивості.

До класів додано коментарі

- Створіть файл autoload.php, який буде містити функцію для автопідключення класів.

Лістинг autoload.php:

```
spl_autoload_register(function ($className) {  
    $paths = ['Models/', 'Views/', 'Controllers/'];  
  
    foreach ($paths as $path) {  
        $file = __DIR__ . '/' . $path . $className . '.php';  
        if (file_exists($file)) {  
            require_once $file;  
            return;  
        }  
    }  
});
```

- Використайте spl\_autoload\_register для автоматичного підключення класів на основі їхніх імен та розташування.

Лістинг index.php:

```
require_once "autoload.php";
```

```
$controller = new UserController();
```

```
echo "1 Завдання<br>";
```

```
$controller->showUser();
```

**Завдання 3.** (Неймспейси)

- Додайте неймспейси до класів у попередньому завданні. Наприклад, "namespace Models;" для "UserModel".

Додато неймспейси до класів.

```
namespace Controllers;
```

```
namespace Models;
```

		Сапацинський Б.І.			Житомирська політехніка.00.000.00.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

```
namespace Views;
```

- Змініть файл autoload.php так, щоб він також враховував неймспейси при підключенні класів

Внесено зміни, лістинг:

```
spl_autoload_register(function ($className) {  
    $file = __DIR__ . '/' . $className . '.php';  
  
    if (file_exists($file)) {  
        require_once $file;  
    }  
});
```

#### Завдання 4. (Автопідключення класів з неймспейсами)

- Використовуйте аналогічний підхід до підключення класів, але тепер з урахуванням неймспейсів.

Зміни вносити не потрібно оскільки файл autoload.php вже підключений.

- Переконайтеся, що класи виводять повідомлення чи результати виклику.

Код працює справно.

#### Завдання 5 (Створення класу. Методи GET і SET)

1) Створіть клас Circle з полями: координати центру і радіус кола

```
<?php  
namespace Models;  
class Circle  
{  
    private float $x;  
    private float $y;  
    private float $radius;
```

2) Створіть конструктор, що приймає значення для 3-х полів

```
    public function __construct(float $x, float $y, float $radius)  
    {  
        $this->x = $x;  
        $this->y = $y;  
        $this->radius = $radius;  
    }
```

3) Створіть метод \_\_toString(), що повертає рядок в форматі: «Коло з центром в (x, y) і радіусом radius»

```
    public function __toString(): string  
    {
```

		Сапацінський Б.І.			Житомирська політехніка.00.000.00.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

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

```

4) Створіть методи GET і SET для всіх 3-х полів.

```

public function getX(): float
{
    return $this->x;
}
public function setX(float $x): void
{
    $this->x = $x;
}
public function getY(): float
{
    return $this->y;
}
public function setY(float $y): void
{
    $this->y = $y;
}
public function getRadius(): float
{
    return $this->radius;
}
public function setRadius(float $radius): void
{
    $this->radius = $radius;
}

```

5) Створіть об'єкт та перевірте всі його методи.

```

echo "2(5) Завдання<br>";
use Models\Circle;
$circle = new Circle(0, 0, 5);

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

$circle->setX(2);
$circle->setY(5);
$circle->setRadius(8);

echo "<br>Змінене коло: " . $circle . "<br>";

```

		Сапацінський Б.І.			Житомирська політехніка.00.000.00.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

Коло з центром в (0, 0) і радіусом 5.  
 Координата X: 0  
 Координата Y: 0  
 Радіус: 5  
 Змінене коло: Коло з центром в (2, 5) і радіусом 8.

Рис. 3. Результат

### Завдання 6 (Модифікатори доступу)

1) В класі з попереднього завдання зробіть всі поля private.

```
class Circle
{
    private float $x;
    private float $y;
    private float $radius;
```

2) Створіть метод, що приймає об'єкт кола, і повертає true, якщо дані кола перетинаються, і false, якщо вони не перетинаються.

```
public function checkIntersection(Circle $otherCircle): bool
{
    $distance = sqrt(pow($this->x - $otherCircle->getX(), 2) + pow($this->y -
    $otherCircle->getY(), 2));

    return $distance <= ($this->radius + $otherCircle->getRadius());
}
```

### 3(6) Завдання

1 коло: Коло з центром в (1, 6) і радіусом 5.  
 2 коло: Коло з центром в (3, 4) і радіусом 5.  
 Кола перетинаються!

Рис. 4. Результат

### Завдання 7 (Статичні властивості і методи)

1) Створіть директорію text, а в ній 3 текстових файла

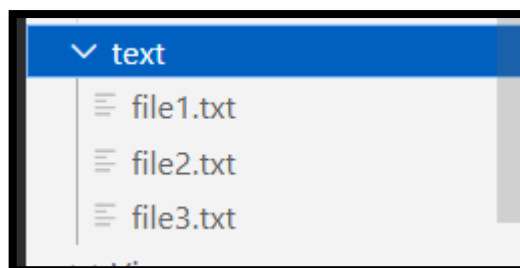


Рис. 5. Директорія text

2) Створіть клас зі статичним полем `dir="text"`

```
<?php
namespace Models;

class FileManager
{
    private static $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.
- Директорія береться зі статичного поля

Читання з файлу:

```
public static function readFile($fileName)
{
    $filePath = self::$dir . "/" . $fileName;

    if (file_exists($filePath)) {
        return file_get_contents($filePath);
    } else {
        return "Файл не знайдено!";
    }
}
```

Запис до файлу:

```
public static function writeFile($fileName, $text)
{
    $filePath = self::$dir . "/" . $fileName;
    file_put_contents($filePath, $text, FILE_APPEND);
}
```

4) Створіть метод, що дозволяє стерти вміст файлу.

```
public static function clearFile($fileName)
{
    $filePath = self::$dir . "/" . $fileName;
    if (file_exists($filePath)) {
        file_put_contents($filePath, "");
    } else {
        return "Файл не знайдено!";
    }
}
```

		Сапацінський Б.І.			Житомирська політехніка.00.000.00.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

Перевірте роботу всіх методів

```
use Models\FileManager;
```

```
echo "<br>4(7) Завдання<br>";
```

```
echo"Вміст file1.txt<br>";
```

```
echo FileManager::readFile("file1.txt") . "<br>";
```

```
FileManager::writeFile("file1.txt", "\nHello world!");
```

```
echo"Вміст file1.txt<br>";
```

```
echo FileManager::readFile("file1.txt") . "<br>";
```

```
echo"Вміст file2.txt<br>";
```

```
echo FileManager::readFile("file2.txt") . "<br>";
```

```
FileManager::clearFile("file2.txt");
```

```
echo"Вміст file2.txt<br>";
```

```
echo FileManager::readFile("file2.txt") . "<br>";
```

4(7) Завдання

Вміст file1.txt

aaaaaaaa aaaaaaaa aaa aaa aaaaaaaa aaa aaa aa a a a a a

Вміст file1.txt

aaaaaaaa aaaaaaaa aaa aaa aaaaaaaa aaa aaa aa a a a a a Hello world!

Вміст file2.txt

bbb bbb bbbbbb bbbbbbbb bbbbbb bbbbbbbb bbbbbbbb bbbbbb bbb

Вміст file2.txt

Рис. 6. Результат

## Завдання 8 (Наслідування)

1) Створіть клас Human з властивостями, що характеризують людину (зріст, маса, вік...). Створіть методи GET і SET для кожної властивості

```
<?php
```

```
namespace Models;
```

```
class Human
```

```
{
```

```
    private $name;
```

```
    private $height;
```

```
    private $weight;
```

```
    private $age;
```

```
    public function __construct($name,$height, $weight, $age)
```

```
{
```

		Сапацінський Б.І.			Житомирська політехніка.00.000.00.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8



```

        $this->name = $name;
        $this->height = $height;
        $this->weight = $weight;
        $this->age = $age;
    }

    public function getName()
    {
        return $this->name;
    }
    public function setName($name)
    {
        $this->name = $name;
    }

    public function getHeight()
    {
        return $this->height;
    }
    public function setHeight($height)
    {
        $this->height = $height;
    }

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

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

```

2) Створіть клас Student, який успадковуватиметься від класу Human:

1. Додайте властивості, специфічні тільки для студента (назва ВНЗ, курс...)

```

private $university;
private $course;

```

		Сапацінський Б.І.			Житомирська політехніка.00.000.00.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

```
private $specialization;
```

2. Додайте в клас методи GET і SET для всіх нових властивостей.

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

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

```
public function getSpecialization()  
{  
    return $this->specialization;  
}  
public function setSpecialization($specialization)  
{  
    $this->specialization = $specialization;  
}
```

3. Реалізуйте метод, який буде переводити студента на новий курс (тобто просто збільшувати значення поля «курс» на 1)

```
public function NextCourse()  
{  
    $this->course += 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...).

```
private $languages;  
private $experience;
```

- Додайте в клас методи GET і SET для всіх нових властивостей.

```
public function getLanguages()  
{  
    return $this->languages;  
}
```

		Сапацінський Б.І.			Житомирська політехніка.00.000.00.000 – Лр4	Арк.
		Ковтун В.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public function setLanguages($languages)
{
    $this->languages = $languages;
}

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

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

```

- Реалізуйте метод, який буде додавати в масив з мовами ще одну мову.

```

public function addLanguage($language)
{
    $this->languages[] = $language;
}

```

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

```

5(8) Завдання
Ім'я: Артур
Зріст: 175 см
Маса: 70 кг
Вік: 23 років
Студент: Політехніка, Курс: 2, Спеціальність: 121 ППЗ
Переведено на курс: 3
Програміст: Мови програмування: PHP, JavaScript, Досвід: 2 років
Програміст після вивчення мови: PHP, JavaScript, Python
Ім'я: Богдан
Зріст: 180 см
Маса: 80 кг
Вік: 30 років
Ім'я: Олександр
Зріст: 160 см
Маса: 75 кг
Вік: 35 років

```

Рис. 7. Результат

## Завдання 9 (Абстрактні класи)

- 1) Зробіть клас Human абстрактним.

```

<?php
namespace Models;

abstract class AbHuman implements HouseCleaning
{
    private $name;
    private $height;
}

```

		Сапачінський Б.І.			Житомирська політехніка.00.000.00.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		11

```

private $weight;
private $age;
...
}

```

2) Напишіть метод «Народження дитини» в класі Human, що викликає метод «Повідомлення при народженні дитини» (не забудьте поставити модифікатор protected), який буде абстрактним

```

abstract protected function birthNotification();
public function birthChild()
{
    echo "Народжено дитину! ";
    $this->birthNotification();
}

```

3) Реалізуйте «Повідомлення при народженні дитини» у класів Student та Programmer.

```

protected function birthNotification()
{
    echo "Студент має дитину!";
}
protected function birthNotification()
{
    echo "Програміст має дитину!";
}

```

Перевірте роботу методів «народження»

**6(9) Завдання**  
 Народжено дитину! Студент має дитину!  
 Народжено дитину! Програміст має дитину!

Рис. 8. Результат

### Завдання 10 (Інтерфейси)

1) Створіть інтерфейс «Прибирання будинку», в якому опишіть 2 методи: «Прибирання кімнати» і «Прибирання кухні»

```

<?php
namespace Models;

interface HouseCleaning
{
    public function cleanRoom();
    public function cleanKitchen();
}

```

		Сапацінський Б.І.			Житомирська політехніка.00.000.00.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

2) Додайте створений інтерфейс в клас Human

```
public function cleanRoom()
{
    echo "Прибирання кімнати";
}
public function cleanKitchen()
{
    echo "Прибирання кухні";
}
```

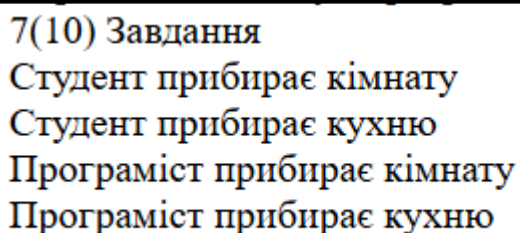
3) Реалізуйте у кожному класі-спадкоємці (Student та Programmer) обидва методи

4) Реалізація повинна бути у вигляді одного з рядків: «Студент прибирає кімнату», «Студент прибирає кухню», «Програміст прибирає кімнату», «Програміст прибирає кухню»,

```
public function cleanRoom(): string {
    return "Студент прибирає кімнату";
}
public function cleanKitchen(): string {
    return "Студент прибирає кухню";
}

public function cleanRoom(): string {
    return "Програміст прибирає кімнату";
}
public function cleanKitchen(): string {
    return "Програміст прибирає кухню";
}
```

5) Перевірте роботу методів прибирання в обох класах



```
7(10) Завдання
Студент прибирає кімнату
Студент прибирає кухню
Програміст прибирає кімнату
Програміст прибирає кухню
```

Рис. 9. Результат

## Висновок

Було ознайомлено з роботою з класами, інтерфейсами та наслідуванням, а також з реалізацією абстрактних методів та інтерфейсів у класах-нащадках.

		Сапачінський Б.І.			Житомирська політехніка.00.000.00.000 – Лр4	Арк.
		Ковтун В.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		