

## ЛАБОРАТОРНА РОБОТА №4

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

**Мета:** навчитися працювати з класами

*Хід роботи:*

### 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.

*Лістинг:*

```
<?php
namespace models;

class UserModel{

    function WriteHello(){
        echo 'Hello';
    }
}
```

```
<?php
namespace controllers;
class UserController{
    function WriteHello(){
        echo 'Hello';
    }
}
```

```
<?php
namespace views;
class UserView{
    function WriteHello(){
        echo 'Hello';
    }
}
```

					ДУ «Житомирська політехніка».24.121.12.000 – Лр4			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Руденко О.І.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Ковтун В.В.						1
Керівник								9
Н. контр.							ФІКТ Гр. ІПЗ-22-2[2]	
Зав. каф.								

## 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

## 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.

### Лістинг:

```
<?php
use views\UserView as UserView;
use models\Circle as Circle;
use controllers\TextController as TextController;
use models\Human as Human;
use models\Student as Student;
use models\Programmer as Programmer;

function autoload($class){
    $path = str_replace('\\', '/', $class) . '.php';
    if(is_file($path)){
        include_once($path);
    }
}

spl_autoload_register("autoload");

$obj = new UserView();
$obj->WriteHello();
echo '<br/>';
```

		Руденко О.І.			ДУ «Житомирська політехніка».24.121.12.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

$objj2 = new Circle(1,2,3);
echo $obj2->__toString();
echo '<br/>';

$objj3 = new TextController();
$objj3->writeDataToFile('file1.txt','some text');
print_r($objj3->readDataFromFile('file1.txt'));

echo '<br/>';
echo '<br/>';
$student = new Student(156, 40, 11, "Uuu1", 1);
$programmer = new Programmer(186, 60, 45, ["Python", "Dart"], 2);

echo "Університет студента : " . $student->getUniversityName() . ", Курс : " .
$student->getCourse() . "\n";
echo "Мови програміста : " . implode(", ", $programmer->getProgrammingLanguages()) .
", Досвід : " . $programmer->getExperience() . " years\n";
echo '<br/>';
$student->setUniversityName("Uuu2");
$student->moveOnToNextCourse();
$programmer->addProgrammingLanguage("C#");
echo '<br/>';
echo "Університет студента : " . $student->getUniversityName() . ", Курс : " .
$student->getCourse() . "\n";
echo '<br/>';
echo "Мови програміста : " . implode(", ", $programmer->getProgrammingLanguages()) .
", Досвід : " . $programmer->getExperience() . " years\n";
echo '<br/>';

echo $student->giveBirth();
echo $programmer->giveBirth();
echo '<br/>';
$student->cleanRoom();
$student->cleanKitchen();
echo '<br/>';
$programmer->cleanRoom();
$programmer->cleanKitchen();

```

### 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
- 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

		Руденко О.І.			ДУ «Житомирська політехніка».24.121.12.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

### 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.

#### Лістинг:

```
<?php
namespace models;
class Circle
{
    private $x;
    private $y;
    private $radius;
    public function __construct($x, $y, $radius) {
        $this->x = $x;
        $this->y = $y;
        $this->radius = $radius;
    }

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

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

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

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

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

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

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

    public function intersect(Circle $otherCircle) {
        $distance = sqrt(($this->x - $otherCircle->getX())**2 + ($this->y - $otherCircle->getY())**2);
        $sumOfRadii = $this->radius + $otherCircle->getRadius();

        return $distance < $sumOfRadii;
    }
}
```

		Руденко О.І.			ДУ «Житомирська політехніка».24.121.12.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

### Завдання 7 (Статичні властивості і методи)

- 1) Створіть директорию **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.
  - Директорія береться зі статичного поля
- 4) Створіть метод, що дозволяє стерти вміст файлу

Перевірте роботу всіх методів

#### Лістинг:

```
<?php
namespace controllers;

class TextController{

    private static $dir = "text";

    public static function readDataFromFile($fileName) {
        $filePath = self::$dir . "/" . $fileName;
        return explode(PHP_EOL, file_get_contents($filePath));
    }

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

}
```

### 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.

		Руденко О.І.			ДУ «Житомирська політехніка».24.121.12.000 – Лр4	Арк.
		Ковтун В.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

- 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»

#### Лістинг:

```
<?php
namespace models;

abstract class Human implements HouseCleaning {
    private $height;
    private $weight;
    private $age;

    protected abstract function birthMessage();

    public function giveBirth() {
        $message = $this->birthMessage();
        echo $message . "\n";
    }

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

    public function cleanRoom() {
        echo get_class($this) . " прибирає кімнату\n";
    }

    public function cleanKitchen() {
        echo get_class($this) . " прибирає кухню\n";
    }

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

    public function getWeight() {
```

		Руденко О.І.			ДУ «Житомирська політехніка».24.121.12.000 – Лр4	Арк.
		Ковтун В.В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return $this->weight;
    }

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

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

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

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

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;
    }

    public function getProgrammingLanguages() {
        return $this->programmingLanguages;
    }

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

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

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

    public function addProgrammingLanguage($newLanguage) {

```

		Руденко О.І.			ДУ «Житомирська політехніка».24.121.12.000 – Лр4	Арк.
		Ковтун В.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $this->programmingLanguages[] = $newLanguage;
    }

    protected function birthMessage() {
        return "Програміст народився!!! Знає такі мови : " . implode(", ", $this->programmingLanguages) . "!";
    }

    public function cleanRoom() {
        echo "Програміст прибирає кімнату\n";
    }

    public function cleanKitchen() {
        echo "Програміст прибирає кухню\n";
    }
}

```

#### 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

***Лістинг:***

```

<?php

namespace models;

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

```

***Результат виконання:***

		Руденко О.І.			ДУ «Житомирська політехніка».24.121.12.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		8



Hello

Коло з центром в (1, 2) та радіусом 3

Array ( [0] => some text [1] => some text [2] => some text [3] => some text [4] => some text [5] => some text [6] => some text [7] => some text [8] => some text [9] => some text [10] => some text [11] => some text [12] => some text [13] => some text [14] => some text [15] => some text [16] => some text [17] => some text [18] => some text [19] => some text [20] => )

Університет студента : Uuu1, Курс : 1 Мови програміста : Python, Dart, Досвід : 2 years

Університет студента : Uuu2, Курс : 2

Мови програміста : Python, Dart, C#, Досвід : 2 years

Студент народився: Uuu2, курс 2! Програміст народився!!! Знає такі мови : Python, Dart, C#!

Студент прибирає кімнату Студент прибирає кухню

Програміст прибирає кімнату Програміст прибирає кухню

***Висновки:*** в ході виконання лабораторної роботи №4 навчився працювати з класами

***Посилання на репозиторій:*** [hoowalex/Back-End-lab4 \(github.com\)](https://github.com/hoowalex/Back-End-lab4)

		Руденко О.І.			ДУ «Житомирська політехніка».24.121.12.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		9