

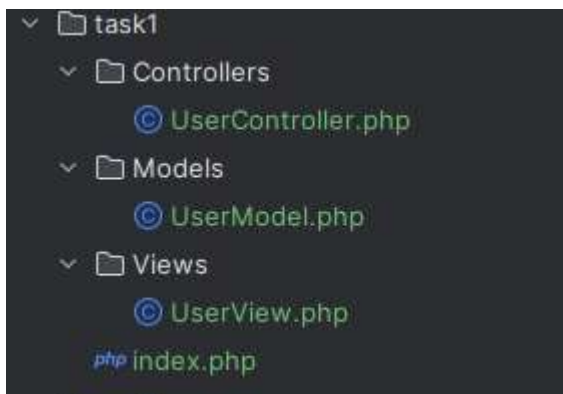
Лабораторна робота №4

Тема: ООП в PHP

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", "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.

					ДУ «Житомирська політехніка».24.121.03.000–Лр6			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Вандалович В.В..			Звіт з лабораторної роботи		Лім.	Арк.
Перевір.		Чижмотря О.В.						1
Керівник								14
Н. контр.							ФІКТ Гр. ІПЗ-23-3[1]	
Зав. каф.								

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

```
php task7\index.php  php task8\index.php  php autoload.php x  Cleaning.php
1  <?php
2  spl_autoload_register(function ($class) {
3      include_once(__DIR__ . DIRECTORY_SEPARATOR . $class . '.php');
4  });
5  |
```

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.

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

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

    public function __toString(): string
    {
        return "Коло з центром в ({ $this->x}, { $this->y}) і радіусом { $this->radius}";
    }
}
```

		Вандалович В.В.			ДУ «Житомирська політехніка».24.121.03.000 – Лр6	Арк.
		Чижмоторя О.В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

>radius}";
    }

    private $x;
    public function getX(): float
    {
        return $this->x;
    }
    public function setX(float $x)
    {
        $this->x = $x;
    }

    private $y;
    public function getY(): float
    {
        return $this->y;
    }
    public function setY(float $y)
    {
        $this->y = $y;
    }

    private $radius;
    public function getRadius(): float
    {
        return $this->radius;
    }
    public function setRadius(float $radius)
    {
        $this->radius = $radius;
    }
}

```

localhost/lab4/task5/

```

radius: 8
x: -1
y: -2

```

		Вандалович В.В.			ДУ «Житомирська політехніка».24.121.03.000 – Лр6	Арк.
		Чижевотря О.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

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.

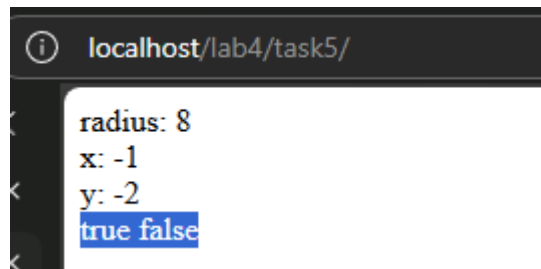
```
3. public function intersects(Circle $circle): bool
{
    $diffx = $circle->getX() - $this->x;
    $diffy = $circle->getY() - $this->y;
    $distanceMin = $circle->getRadius() + $this->radius;

    $distance = sqrt(pow($diffx, 2) + pow($diffy, 2));

    return $distance < $distanceMin;
}
```

```
$c2 = new Circle(5, 0, 10);
$c3 = new Circle(0, 25, 10);

echo $c1->intersects($c2) ? "true " : "false ";
echo $c1->intersects($c3) ? "true " : "false ";
```



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

Перевірте роботу всіх методів

		Вандалович В.В.			ДУ «Житомирська політехніка».24.121.03.000 – Лр6	Арк.
		Чижевотря О.В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

class FileManager
{
    public static $dir = 'text';

    public static function writeToFile(string $filename, string $text)
    {
        $path = self::$dir . '/' . $filename;
        file_put_contents($path, $text . PHP_EOL, FILE_APPEND);
    }

    public static function readFile(string $filename): string
    {
        $path = self::$dir . '/' . $filename;
        return file_exists($path) ? file_get_contents($path) : '';
    }

    public static function clearFile(string $filename)
    {
        $path = self::$dir . '/' . $filename;
        file_put_contents($path, '');
    }
}

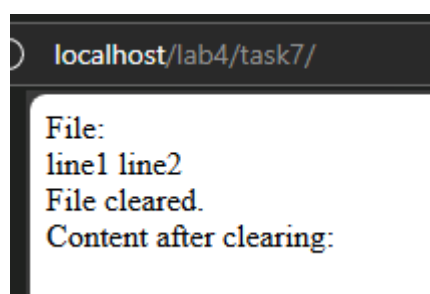
FileManager::writeToFile('file1.txt', 'line1');
FileManager::writeToFile('file1.txt', 'line2');

$content = FileManager::readFile('file1.txt');
echo "File:<br>$content";

FileManager::clearFile('file1.txt');
echo "<br>File cleared.<br>";

$content = FileManager::readFile('file1.txt');
echo "Content after clearing:<br>$content";

```



Завдання 8 (Наслідування)

- Створіть клас **Human** з властивостями, що характеризують людину (зріст, маса, вік...). Створіть методи **GET** і **SET** для кожної властивості
- Створіть клас **Student**, який успадковуватиметься від класу **Human**:
 - Додайте властивості, специфічні тільки для студента (назва ВНЗ, курс...)

		Вандалович В.В.			ДУ «Житомирська політехніка».24.121.03.000 – Лр6	Арк.
		Чижмотря О.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

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**

```
localhost/lab4/task8/

student : Politeh, course: 2
new height: 175, new weight: 65, course: 3
programmer knows languages: PHP, JavaScript, experience: 5 years
new height: 182, new weight: 78
new languages: PHP, JavaScript
```

		Вандалович В.В.			ДУ «Житомирська політехніка».24.121.03.000 – Лр6	Арк.
		Чижмоторя О.В.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

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**

```
public function create_child(){
    $this->child_message();
}
protected abstract function child_message();
```

Перевірте роботу методів «народження»

```
localhost/lab4/task8/
student : Politeh, course: 2
new height: 175, new weight: 65, course: 3
programmer knows languages: PHP, JavaScript, experience: 5 years
new height: 182, new weight: 78
new languages: PHP, JavaScript
student created child
programmer created child
```

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

```
6. interface Cleaning{
    function cleanBedroom();
    function cleanKitchen();
}
```

		Вандалович В.В.			ДУ «Житомирська політехніка».24.121.03.000 – Лр6	Арк.
		Чижемотря О.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

Student.php

```

4      class Student extends Human
46     function cleanBedroom()
47     {
48         echo "student cleaned bedroom<br>";
49     }
50
1 usage new *
51     function cleanKitchen()
52     {
53         echo "student cleaned kitchen<br>";
54     }
55 }

```

Programmer.php

```

4      class Programmer extends Human
48     function cleanBedroom()
49     {
50         echo "programmer cleaned bedroom<br>";
51     }
52
1 usage new *
53     function cleanKitchen()
54     {
55         echo "programmer cleaned kitchen<br>";
56     }
57 }

```

localhost/lab4/task8/

```

student : Politeh, course: 2
new height: 175, new weight: 65, course: 3
programmer knows languages: PHP, JavaScript, experience: 5 years
new height: 182, new weight: 78
new languages: PHP, JavaScript
student created child
programmer created child
student cleaned bedroom
student cleaned kitchen
programmer cleaned bedroom
programmer cleaned kitchen

```

Висновок: під час виконання лабораторної роботи я ознайомився з класами та роботою з ООП в мові PHP

		Вандалович В.В.			ДУ «Житомирська політехніка».24.121.03.000 – Лр6	Арк.
		Чижмотря О.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		