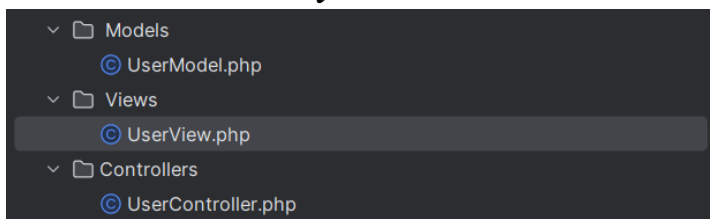


**Лабораторна робота №4**  
**Тема: Об'єктно-орієнтоване програмування в PHP**  
**Мета роботи: навчитися працювати з класами**  
**Завдання на лабораторну роботу:**

**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.

**Результат:**



**Лістинг:**

**class UserModel:**

```
<?php
class UserModel {
    public function getMessage() {
        return "User Model Message";
    }
}
```

**class UserController:**

```
<?php
class UserController {
    public function getMessage() {
        return "User Controller Message";
    }
}
```

**class UserView:**

```
<?php
class UserView {
    public function getMessage() {
        return "User View Message";
    }
}
```

					ДУ «Житомирська політехніка».24.121.4.000 – Лр4							
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи №4			Лім.	Арк.	Аркушіє		
Розроб.		Кравченко К. М.								1	16	
Перевір.		Ковтун В.В.						ФІКТ Гр. ІПЗ-22-2[2]				
Керівник												
Н. контр.												
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.

**Лістинг:**  
***class UserModel:***

```
<?php

/**
 * Клас UserModel представляє модель користувача
 * @package Models
 */
class UserModel {
    /**
     * Отримати повідомлення від моделі користувача
     * @return string
     */
    public function getMessage() {
        return "User Model Message";
    }
}
```

***class UserController:***

```
<?php

/**
 * Клас UserController представляє контролер користувача
 * @package Controllers
 */
class UserController {
    /**
     * Отримати повідомлення від контролера користувача
     * @return string
     */
    public function getMessage() {
        return "User Controller Message";
    }
}
```

***class UserView:***

```
<?php

/**
 * Клас UserView представляє представлення користувача.
 * @package Views
 */
class UserView {
    /**
     * Отримати повідомлення від представлення користувача
     * @return string
     */
    public function getMessage() {
        return "User View Message";
    }
}
```

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```
}  
}
```

### *Autoload.php:*

```
<?php  
function myAutoloader($class) {  
    $class = str_replace("\\", DIRECTORY_SEPARATOR, $class);  
    include __DIR__ . DIRECTORY_SEPARATOR . $class . '.php';  
}  
  
spl_autoload_register('myAutoloader');
```

### *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

### *Лістинг: class UserModel:*

```
<?php  
  
namespace Models;  
/**  
 * Клас UserModel представляє модель користувача  
 * @package Models  
 */  
class UserModel {  
    /**  
     * Отримати повідомлення від моделі користувача  
     * @return string  
     */  
    public function getMessage() {  
        return "User Model Message";  
    }  
}
```

### *class UserController:*

```
<?php  
  
namespace Controllers;  
/**  
 * Клас UserController представляє контролер користувача  
 * @package Controllers  
 */  
class UserController {  
    /**  
     * Отримати повідомлення від контролера користувача  
     * @return string  
     */  
    public function getMessage() {  
        return "User Controller Message";  
    }  
}
```

### *class UserView:*

```
<?php  
  
namespace Views;  
/**  
 * Клас UserView представляє представлення користувача.
```

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

* @package Views
*/
class UserView {
    /**
     * Отримати повідомлення від представлення користувача
     * @return string
     */
    public function getMessage() {
        return "User View Message";
    }
}

```

### ***Autoload.php:***

```

<?php
function myAutoloader($class) {
    $classPath = str_replace("\\", DIRECTORY_SEPARATOR, $class);
    $filePath = __DIR__ . DIRECTORY_SEPARATOR . $classPath . '.php';
    if (file_exists($filePath)) {
        include $filePath;
    }
}

spl_autoload_register('myAutoloader');

```

### ***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)***

- Створіть клас Circle з полями: координати центру і радіус кола
- Створіть конструктор, що приймає значення для 3-х полів
- Створіть метод \_\_toString(), що повертає рядок в форматі: «Коло з центром в (x, y) і радіусом radius»
- Створіть методи GET і SET для всіх 3-х полів
- Створіть об'єкт та перевірте всі його методи

### ***Лістинг:***

```

<?php
class Circle {
    public $x;
    public $y;
    public $radius;

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

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

```

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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;
}
}

$circle = new Circle(3, 4, 5);

echo $circle->__toString() . "\n";

echo "X: " . $circle->getX() . "\n";
echo "Y: " . $circle->getY() . "\n";
echo "Радіус: " . $circle->getRadius() . "\n";

$circle->setX(10);
$circle->setY(15);
$circle->setRadius(8);

echo "Нові значення:\n";
echo "X: " . $circle->getX() . "\n";
echo "Y: " . $circle->getY() . "\n";
echo "Радіус: " . $circle->getRadius() . "\n";

echo $circle->getX();
echo $circle;

```

**Результат:**

**Перевірка:**

Коло з центром в (3, 4) і радіусом 5 X: 3 Y: 4 Радіус: 5 Нові значення: X: 10 Y: 15 Радіус: 8 10 Коло з центром в (10, 15) і радіусом 8

### Завдання 6 (Модифікатори доступу)

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

**Лістинг:**

```

<?php

class Circle {
    private $x;
    private $y;

```

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

private $radius;

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

public function __toString() {
    return "Коло з центром в ($this->x, $this->y) і 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 intersectsWith(Circle $otherCircle) {
    $distance = sqrt(($this->x - $otherCircle->getX()) ** 2 + ($this->y - $otherCircle->getY()) ** 2);
    $sumRadii = $this->radius + $otherCircle->getRadius();

    return $distance < $sumRadii;
}

}

$circle1 = new Circle(3, 4, 5);
$circle2 = new Circle(8, 10, 6);

echo $circle1->__toString() . "\n";
echo $circle2->__toString() . "\n";

if ($circle1->intersectsWith($circle2)) {
    echo "Кола перетинаються.\n";
} else {
    echo "Кола не перетинаються.\n";
}
}

```

### Результат:

Коло з центром в (3, 4) і радіусом 5 Коло з центром в (8, 10) і радіусом 6 Кола перетинаються.

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

### Завдання 7 (Статичні властивості і методи)

- Створіть директорию text, а в ній 3 текстових файла
- Створіть клас зі статичним полем dir="text"
- 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
- Створіть метод, що дозволяє стерти вміст файлу
- Перевірте роботу всіх методів

**Лістинг:**

**File1.txt:**

Кравченко Катерина

**File2.txt:**

Лабораторна 4

**File3.txt:**

07.03.2023 Backend

**Filemanager:**

```
<?php
namespace text;

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

    public static function readFile($fileName) {
        $filePath = self::$dir . '/' . $fileName;
        if (file_exists($filePath)) {
            return file_get_contents($filePath);
        } else {
            return "Файл '$fileName' не існує.";
        }
    }

    public static function writeFile($fileName, $content) {
        $filePath = self::$dir . '/' . $fileName;
        file_put_contents($filePath, $content, FILE_APPEND);
        return "Дані успішно записані у файл '$fileName'.";
    }

    public static function clearFile($fileName) {
        $filePath = self::$dir . '/' . $fileName;
        file_put_contents($filePath, '');
        return "Вміст файлу '$fileName' очищено.";
    }
}

file_put_contents('text/file1.txt', 'Hello, ');
file_put_contents('text/file2.txt', 'World!');
file_put_contents('text/file3.txt', 'This is a test.');
```

```
echo FileManager::readFile('file1.txt') . "\n";
echo FileManager::readFile('file2.txt') . "\n";
```

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

echo FileManager::readFile('file3.txt') . "\n";

echo FileManager::writeFile('file1.txt', 'New content.') . "\n";

echo FileManager::readFile('file1.txt') . "\n";

echo FileManager::clearFile('file2.txt') . "\n";

echo FileManager::readFile('file2.txt') . "\n";

```

**Результат:**

**File1.txt:**

Hello,

**File2.txt:**

World!

**File3.txt:**

This is a test.

### **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...)
- Додайте в клас методи GET і SET для всіх нових властивостей.
- Реалізуйте метод, який буде переводити студента на новий курс (тобто просто збільшувати значення поля «курс» на 1)
- 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

**Лістинг:**

```

<?php

class Human {
    private $height;
    private $weight;
    private $age;

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

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

```

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



```

    }

    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;
    }
}

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;
    }

    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 moveToNextCourse() {
        $this->course++;
    }
}

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;
    }
}

```

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

    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;
    }

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

$human = new Human(175, 70, 25);
echo "Human:\n";
echo "Height: " . $human->getHeight() . "\n";
echo "Weight: " . $human->getWeight() . "\n";
echo "Age: " . $human->getAge() . "\n";

$student = new Student(170, 65, 20, 'University A', 1);
echo "\nStudent:\n";
echo "Height: " . $student->getHeight() . "\n";
echo "Weight: " . $student->getWeight() . "\n";
echo "Age: " . $student->getAge() . "\n";
echo "University: " . $student->getUniversity() . "\n";
echo "Course: " . $student->getCourse() . "\n";
$student->moveToNextCourse();
echo "After moving to the next course:\n";
echo "Course: " . $student->getCourse() . "\n";

$programmer = new Programmer(180, 75, 30, ['Java', 'Python'], 5);
echo "\nProgrammer:\n";
echo "Height: " . $programmer->getHeight() . "\n";
echo "Weight: " . $programmer->getWeight() . "\n";
echo "Age: " . $programmer->getAge() . "\n";
echo "Programming Languages: " . implode(', ', $programmer->getProgramming-
Languages()) . "\n";
echo "Experience: " . $programmer->getExperience() . " years\n";
$programmer->addProgrammingLanguage('JavaScript');
echo "After adding JavaScript:\n";
echo "Programming Languages: " . implode(', ', $programmer->getProgramming-
Languages()) . "\n";

```

### Результат:

← → ↺ Не конфіденційний backendlab4/task8.9.10.php

Human: Height: 175 Weight: 70 Age: 25 Student: Height: 170 Weight: 65 Age: 20 University: University A Course: 1 After moving to the next course: Course: 2 Programmer: Height: 180 Weight: 75 Age: 30 Programming Languages: Java, Python Experience: 5 years After adding JavaScript: Programming Languages: Java, Python, JavaScript

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

### Завдання 9 (Абстрактні класи)

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

#### Лістинг:

```
<?php

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

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

    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;
    }

    abstract protected function birthNotification();
}

class Student extends Human {
    private $university;
    private $course;

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

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $this->course = $course;
    }

    protected function birthNotification() {
        echo "Народився новий студент!\n";
    }

    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 moveToNextCourse() {
        $this->course++;
    }
}

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;
    }

    protected function birthNotification() {
        echo "Народився новий програміст!\n";
    }

    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;
    }

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

```

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

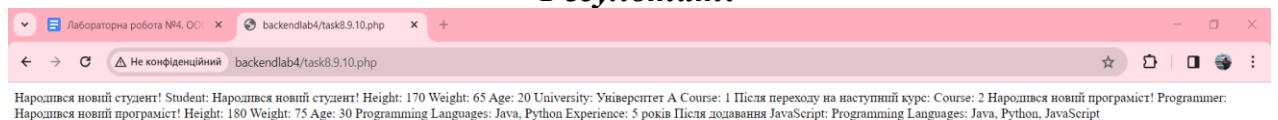
```

$student = new Student(170, 65, 20, 'Університет А', 1);
echo "\nStudent:\n";
echo "Height: " . $student->getHeight() . "\n";
echo "Weight: " . $student->getWeight() . "\n";
echo "Age: " . $student->getAge() . "\n";
echo "University: " . $student->getUniversity() . "\n";
echo "Course: " . $student->getCourse() . "\n";
$student->moveToNextCourse();
echo "Після переходу на наступний курс:\n";
echo "Course: " . $student->getCourse() . "\n";

$programmer = new Programmer(180, 75, 30, ['Java', 'Python'], 5);
echo "\nProgrammer:\n";
echo "Height: " . $programmer->getHeight() . "\n";
echo "Weight: " . $programmer->getWeight() . "\n";
echo "Age: " . $programmer->getAge() . "\n";
echo "Programming Languages: " . implode(', ', $programmer->getProgramming-
Languages()) . "\n";
echo "Experience: " . $programmer->getExperience() . " років\n";
$programmer->addProgrammingLanguage('JavaScript');
echo "Після додавання JavaScript:\n";
echo "Programming Languages: " . implode(', ', $programmer->getProgramming-
Languages()) . "\n";
?>

```

### Результат:



### Завдання 10 (Інтерфейси)

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

### Лістинг:

```

<?php

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

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

```

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

```

private $age;

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

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;
}

abstract protected function birthNotification();

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

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

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;
    }

    protected function birthNotification() {
        echo "Народився новий студент!\n";
    }

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

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

```

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }

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

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

    public function moveToNextCourse() {
        $this->course++;
    }
}

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;
    }

    protected function birthNotification() {
        echo "Народився новий програміст!\n";
    }

    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;
    }

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

$student = new Student(170, 65, 20, 'Університет А', 1);
echo "\nStudent:\n";
echo "Народився новий студент!\n";
echo "Height: " . $student->getHeight() . "\n";
echo "Weight: " . $student->getWeight() . "\n";
echo "Age: " . $student->getAge() . "\n";
echo "University: " . $student->getUniversity() . "\n";
echo "Course: " . $student->getCourse() . "\n";
$student->moveToNextCourse();
echo "Після переходу на наступний курс:\n";
echo "Course: " . $student->getCourse() . "\n";
echo "Виконання методів прибирання:\n";

```

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$student->cleanRoom();
$student->cleanKitchen();

$programmer = new Programmer(180, 75, 30, ['Java', 'Python'], 5);
echo "\nProgrammer:\n";
echo "Народився новий програміст!\n";
echo "Height: " . $programmer->getHeight() . "\n";
echo "Weight: " . $programmer->getWeight() . "\n";
echo "Age: " . $programmer->getAge() . "\n";
echo "Programming Languages: " . implode(', ', $programmer->getProgramming-
Languages()) . "\n";
echo "Experience: " . $programmer->getExperience() . " років\n";
$programmer->addProgrammingLanguage('JavaScript');
echo "Після додавання JavaScript:\n";
echo "Programming Languages: " . implode(', ', $programmer->getProgramming-
Languages()) . "\n";
echo "Виконання методів прибирання:\n";
$programmer->cleanRoom();
$programmer->cleanKitchen();
?>

```

### Результат:

Народився новий студент! Student: Народився новий студент! Height: 170 Weight: 65 Age: 20 University: Університет А Course: 1 Після переходу на наступний курс: Course: 2 Виконання методів прибирання: Чистимо кімнату Чистимо кухню Народився новий програміст! Programmer: Народився новий програміст! Height: 180 Weight: 75 Age: 30 Programming Languages: Java, Python Experience: 5 років Після додавання JavaScript: Programming Languages: Java, Python, JavaScript Виконання методів прибирання: Чистимо кімнату Чистимо кухню

**Висновок:** навчилась працювати з класами.

		Кравченко К. М.			ДУ «Житомирська політехніка».24.121.4.000 – Лр4	Арк.
		Ковтун В.В.				16
Змн.	Арк.	№ докум.	Підпис	Дата		