

Лабораторна робота №4

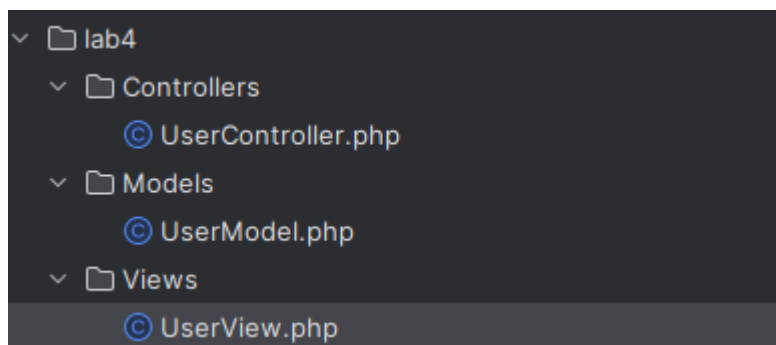
Тема: Об'єктно-орієнтоване програмування в PHP

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", "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.

Лістинг програми(spl_autoload_register):

```
<?php
```

```
/**
```

* Файл autoload.php для автоматичного підключення класів на основі їхніх імен та розташування.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Васюта В.В.			Звіт з лабораторної роботи №4	Літ.	Арк.
Перевір.		Ковтун В.В.					1
Реценз.						ФІКТ, гр. ІПЗ-22-1	
Н. Контр.							
Зав.каф.		Єфіменко А.А.					

*/

```
function autoload($className) {  
    $file = __DIR__ . '/' . str_replace('\\', '/', $className) . '.php';  
    if (file_exists($file)) {  
        require_once $file;  
    }  
}
```

spl_autoload_register('autoload');

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  
namespace Models;  
  
/**  
 * Class UserModel  
 * @package Models  
 *  
 * Модель користувача, яка забезпечує доступ до даних користувачів.  
 */  
no usages  
class UserModel {  
    /**  
     * Отримати дані користувача за його ідентифікатором.  
     *  
     * @param int $userId Ідентифікатор користувача  
     * @return string Дані користувача у вигляді рядка  
     */  
    1 usage  
    public function getUserById($userId) {  
        return "Дані користувача з ID {$userId}";  
    }  
}
```

```
<?php  
namespace Views;  
  
/**  
 * Class UserView  
 * @package Views  
 *  
 * Вид користувача, який відповідає за відображення даних користувача.  
 */  
no usages  
class UserView {  
    /**  
     * Відобразити дані користувача у вигляді HTML.  
     *  
     * @param string $userData Дані користувача у вигляді рядка  
     */  
    no usages  
    public function displayUser($userData) {  
        echo "<p>{$userData}</p>";  
    }  
}
```

```
<?php  
namespace Controllers;  
  
use Models\UserModel;  
  
/**  
 * Class UserController  
 * @package Controllers  
 *  
 * Контролер користувача, який керує взаємодією з моделлю і видом для користувача.  
 */  
no usages  
class UserController {  
    /**  
     * Отримати дані користувача за його ідентифікатором та вивести їх на екран.  
     *  
     * @param int $userId Ідентифікатор користувача  
     */  
    no usages  
    public function getUser($userId) {  
        $userModel = new UserModel();  
        $userData = $userModel->getUserById($userId);  
        echo "Дані користувача: {$userData}";  
    }  
}
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

Лістинг програми(**autoload.php**):

```
<?php
```

```
/**
```

* Файл autoload.php для автоматичного підключення класів на основі їхніх імен та розташування, включаючи неймспейси.

```
*/
```

```
function autoload($className) {
```

```
    $className = ltrim($className, '\\');
```

```
    $fileName = '';
```

```
    $namespace = '';
```

```
    if ($lastNsPos = strrpos($className, '\\')) {
```

```
        $namespace = substr($className, 0, $lastNsPos);
```

```
        $className = substr($className, $lastNsPos + 1);
```

```
        $fileName = str_replace('\\', DIRECTORY_SEPARATOR, $namespace) .
```

```
DIRECTORY_SEPARATOR;
```

```
    }
```

```
    $fileName .= str_replace('_', DIRECTORY_SEPARATOR, $className) . '.php';
```

```
    $fullFileName = __DIR__ . DIRECTORY_SEPARATOR . $fileName;
```

```
    if (file_exists($fullFileName)) {
```

```
        require $fullFileName;
```

```
    }
```

```
}
```

```
spl_autoload_register('autoload');
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

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 програми(autoload.php):

```
<?php
```

```
/**
```

* Файл autoload.php для автоматичного підключення класів на основі їхніх неймспейсів і імен.

```
*/
```

```
function autoload($className) {
```

```
    $className = ltrim($className, '\\');
```

```
    $fileName = '';
```

```
    $namespace = '';
```

```
    if ($lastNsPos = strrpos($className, '\\')) {
```

```
        $namespace = substr($className, 0, $lastNsPos);
```

```
        $className = substr($className, $lastNsPos + 1);
```

```
        $fileName = str_replace('\\', DIRECTORY_SEPARATOR, $namespace) .
```

```
DIRECTORY_SEPARATOR;
```

```
    }
```

```
    $fileName .= str_replace('_', DIRECTORY_SEPARATOR, $className) . '.php';
```

```
    $fullFileName = __DIR__ . DIRECTORY_SEPARATOR . $fileName;
```

```
    if (file_exists($fullFileName)) {
```

```
        require $fullFileName;
```

```
    }
```

```
}
```

```
sp_ autoload_register('autoload');
```

ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4

Арк.

4

Змн. Арк. № докум. Підпис Дата

Лістинг програми(index.php):

```
<?php
require_once('autoload.php');

use Controllers\UserController;
use Views\UserView;

$userController = new UserController();
$userView = new UserView();

$userData = $userController->getUser(123);
$userView->displayUser($userData);
```

Результат виконання:

Дані користувача: Дані користувача з ID 123

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

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 програми(Circle.php):

```
<?php
namespace Models;
```

```
class Circle {
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
		private \$x;				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```
private $y;
```

```
private $radius;
```

```
public function __construct($x, $y, $radius) {
```

```
    $this->x = $x;
```

```
    $this->y = $y;
```

```
    $this->radius = $radius;
```

```
}
```

```
public function __toString() {
```

```
    return "Circle with center at ({ $this->x }, { $this->y }) and radius { $this->radius }";
```

```
}
```

```
public function getX() {
```

```
    return $this->x;
```

```
}
```

```
public function getY() {
```

```
    return $this->y;
```

```
}
```

```
public function getRadius() {
```

```
    return $this->radius;
```

```
}
```

```
public function setX($x) {
```

```
    $this->x = $x;
```

```
}
```

```
public function setY($y) {
```

```
    $this->y = $y;
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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

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());
}
}

```

Лістинг програми(index.php)

```
<?php
```

```
use Models\Circle;
```

```
require_once 'D:\WampServer\www\lab4\autoload.php';
```

```
$circle1 = new Circle(0, 0, 5);
```

```
echo $circle1 . "<br>";
```

```
echo "X: " . $circle1->getX() . ", Y: " . $circle1->getY() . ", Radius: " . $circle1->getRadius() . "<br>"; // Виведе: X: 0, Y: 0, Radius: 5
```

```
$circle1->setX(2);
```

```
$circle1->setY(3);
```

```
$circle1->setRadius(8);
```

```
echo "Updated: X: " . $circle1->getX() . ", Y: " . $circle1->getY() . ", Radius: " .
```

```
$circle1->getRadius() . "<br>"; // Виведе: Updated: X: 2, Y: 3, Radius: 8
```

```
$circle2 = new Circle(4, 4, 5);
```

```
if ($circle1->intersects($circle2)) {
```

```
    echo "Кола перетинаються";
```

```
} else {
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

echo "Кола не перетинаються";
}

```

Результат виконання програми:

```

Circle with center at (0, 0) and radius 5
X: 0, Y: 0, Radius: 5
Updated: X: 2, Y: 3, Radius: 8
Кола перетинаються

```

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

Лістинг програми(index.php)

```
<?php
```

```
use Models\Circle;
```

```
require_once 'D:\WampServer\www\lab4\autoload.php';
```

```
class FileManager {
```

```
    public static $dir = "text/";
```

```
    public static function readFromFile($filename) {
```

```
        $filePath = self::$dir . $filename;
```

```
        if (file_exists($filePath)) {
```

```
            return file_get_contents($filePath);
```

```
        } else {
```

```
            return "File $filename not found!";
```

```
        }
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		


```

public static function writeToFile($filename, $contentToAppend) {
    $filePath = self::$dir . $filename;
    $content = file_exists($filePath) ? file_get_contents($filePath) : "";
    $content .= $contentToAppend;
    file_put_contents($filePath, $content);
}

```

```

public static function clearFile($filename) {
    $filePath = self::$dir . $filename;
    file_put_contents($filePath, "");
}
}

```

```

$circle1 = new Circle(0, 0, 5);
echo $circle1 . "<br>";

```

```

echo "X: " . $circle1->getX() . ", Y: " . $circle1->getY() . ", Radius: " . $circle1->getRadius() . "<br>";
$circle1->setX(2);
$circle1->setY(3);
$circle1->setRadius(8);
echo "Updated: X: " . $circle1->getX() . ", Y: " . $circle1->getY() . ", Radius: " . $circle1->getRadius() . "<br>";

```

```

$circle2 = new Circle(4, 4, 5);
if ($circle1->intersects($circle2)) {
    echo "Кола перетинаються";
} else {
    echo "Кола не перетинаються";
}

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

```
$file1 = 'file1.txt';
```

```
$file2 = 'file2.txt';
```

```
$file3 = 'file3.txt';
```

```
FileManager::writeToFile($file1, "First line of file 1.\n");
```

```
FileManager::writeToFile($file2, "First line of file 2.\n");
```

```
echo FileManager::readFromFile($file1) . "<br>";
```

```
echo FileManager::readFromFile($file2) . "<br>";
```

```
FileManager::writeToFile($file1, "Second line of file 1.\n");
```

```
echo FileManager::readFromFile($file1) . "<br>";
```

```
FileManager::clearFile($file2);
```

```
echo FileManager::readFromFile($file2) . "<br>";
```

Результат виконання програми:

Circle with center at (0, 0) and radius 5

X: 0, Y: 0, Radius: 5

Updated: X: 2, Y: 3, Radius: 8

Кола перетинаютьсяwasdawasdFirst line of file 1.

wasdaFirst line of file 2.

wasdawasdFirst line of file 1. Second line of file 1.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

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**

Лістинг програми(Human):

```
<?php
```

```
namespace Models;
```

```
class Human
```

```
{
```

```
    private $height;
```

```
    private $weight;
```

```
    private $age;
```

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

```
{
```

```
        $this->height = $height;
```

```
        $this->weight = $weight;
```

ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4

Арк.

11

```

        $this->age = $age;
    }

    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;
    }

```

	}				ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

```
public function setAge($age)
```

```
{
```

```
    $this->age = $age;
```

```
}
```

```
public function __toString()
```

```
{
```

```
    return "Людина з ростом {$this->height} см, вагою {$this->weight} кг і віком  
    {$this->age} років.";
```

```
}
```

```
}
```

Лістинг програми(Student):

```
<?php
```

```
namespace Models;
```

```
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;
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Пр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

```
}
```

```
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 advanceCourse()
```

```
{
```

```
    $this->course += 1;
```

```
}
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public function __toString()
{
    return parent::__toString() . ", студент ВНЗ {$this->university}, курс {$this->course}.";
}
}

```

Лістинг програми(Programmer):

```
<?php
```

```
namespace Models;
```

```
class Programmer extends Human
```

```

{
    private $languages;
    private $experience;

```

```
public function __construct($height, $weight, $age, $languages, $experience)
```

```

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

```

```
public function getLanguages()
```

```

{
    return $this->languages;
}

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

```
}
```

```
public function setLanguages($languages)
```

```
{
```

```
    $this->languages = $languages;
```

```
}
```

```
public function getExperience()
```

```
{
```

```
    return $this->experience;
```

```
}
```

```
public function setExperience($experience)
```

```
{
```

```
    $this->experience = $experience;
```

```
}
```

```
public function addLanguage($language)
```

```
{
```

```
    $this->languages[] = $language;
```

```
}
```

```
public function __toString()
```

```
{
```

```
    $languages = implode(' ', $this->languages);
```

```
    return parent::__toString() . ", програміст зі знанням мов: {$languages}, досвід  
роботи: {$this->experience} років.";
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Пр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16


```
}  
  
}
```

Лістинг програми(index.php)

```
<?php  
  
use Models\Human;  
  
use Models\Programmer;  
  
use Models\Student;  
  
  
require_once 'D:\WampServer\www\lab4\autoload.php';  
  
  
$human = new Human(180, 75, 30);  
echo "Людина 1: " . $human . "<br>";  
  
  
$human->setHeight(185);  
$human->setWeight(80);  
$human->setAge(35);  
echo "Оновлені дані: " . $human . "<br>";  
  
  
$programmer = new Programmer(170, 70, 25, ['PHP', 'JavaScript'], 5);  
echo "Програміст: " . $programmer . "<br>";  
  
  
$programmer->addLanguage('Python');  
echo "Оновлені дані програміста: " . $programmer . "<br>";
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

```
$student = new Student(165, 60, 20, 'Київський Політехнічний Інститут', 2);

echo "Студент: " . $student . "<br>";

$student->advanceCourse();

echo "Оновлений курс студента: " . $student->getCourse() . "<br>";
```

Результат виконання програми:

Людина 1: Людина з ростом 180 см, вагою 75 кг і віком 30 років.
 Оновлені дані: Людина з ростом 185 см, вагою 80 кг і віком 35 років.
 Програміст: Людина з ростом 170 см, вагою 70 кг і віком 25 років., програміст зі знанням мов: PHP, JavaScript, досвід роботи: 5 років.
 Оновлені дані програміста: Людина з ростом 170 см, вагою 70 кг і віком 25 років., програміст зі знанням мов: PHP, JavaScript, Python, досвід роботи: 5 років.
 Студент: Людина з ростом 165 см, вагою 60 кг і віком 20 років., студент ВНЗ Київський Політехнічний Інститут, курс 2.
 Оновлений курс студента: 3

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 програми(Human):

```
<?php
```

```
namespace Models;
```

```
abstract class Human
```

```
{
```

```
    private $height;
```

```
    private $weight;
```

```
    private $age;
```

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

Арк.

ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4

18

Змн. Арк. № докум. Підпис Дата

```
{
    $this->height = $height;
    $this->weight = $weight;
    $this->age = $age;
}
```

```
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()				ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

```

{
    return $this->age;
}

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

abstract protected function birthAnnouncement();

public function giveBirth()
{
    $this->birthAnnouncement();
}

public function __toString()
{
    return "Людина з зростом {$this->height} см, вагою {$this->weight} кг і віком
    {$this->age} років";
}
}

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

Лістинг програми(Student):

```
<?php
```

```
namespace Models;
```

```
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;
```

```
    }
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

```
public function getCourse()
```

```
{
```

```
    return $this->course;
```

```
}
```

```
public function setCourse($course)
```

```
{
```

```
    $this->course = $course;
```

```
}
```

```
public function advanceCourse()
```

```
{
```

```
    $this->course += 1;
```

```
}
```

```
protected function birthAnnouncement()
```

```
{
```

```
    echo "Студент народив дитину!<br>";
```

```
}
```

```
public function __toString()
```

```
{
```

```
    return parent::__toString() . ", студент ВНЗ {$this->university}, курс {$this->course}.";
```

```
}
```

```
}
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг програми(Programmer):

```
<?php
```

```
namespace Models;
```

```
class Programmer extends Human
```

```
{
```

```
    private $languages;
```

```
    private $experience;
```

```
    public function __construct($height, $weight, $age, $languages, $experience)
```

```
    {
```

```
        parent::__construct($height, $weight, $age);
```

```
        $this->languages = $languages;
```

```
        $this->experience = $experience;
```

```
    }
```

```
    public function getLanguages()
```

```
    {
```

```
        return $this->languages;
```

```
    }
```

```
    public function setLanguages($languages)
```

```
    {
```

```
        $this->languages = $languages;
```

	}				ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

```
public function getExperience()
```

```
{
```

```
    return $this->experience;
```

```
}
```

```
public function setExperience($experience)
```

```
{
```

```
    $this->experience = $experience;
```

```
}
```

```
public function addLanguage($language)
```

```
{
```

```
    $this->languages[] = $language;
```

```
}
```

```
protected function birthAnnouncement()
```

```
{
```

```
    echo "Програміст народив дитину!<br>";
```

```
}
```

```
public function __toString()
```

```
{
```

```
    $languages = implode(', ', $this->languages);
```

```
    return parent::__toString() . ", програміст зі знанням мов: {$languages}, досвід  
роботи: {$this->experience} років.";
```

```
}
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24


```
}
```

Лістинг програми(index.php):

```
<?php
```

```
require_once 'D:\WampServer\www\lab4\autoload.php';
```

```
use Models\Human;
```

```
use Models\Student;
```

```
use Models\Programmer;
```

```
$programmer = new Programmer(180, 75, 30, ['PHP', 'JavaScript'], 8);
```

```
echo "Програміст:<br>";
```

```
echo $programmer . "<br>";
```

```
$programmer->giveBirth();
```

```
echo "<br>";
```

```
$student = new Student(170, 60, 20, 'KHU', 3);
```

```
echo "Студент:<br>";
```

```
echo $student . "<br>";
```

```
$student->giveBirth();
```

```
echo "<br>";
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

Результат виконання програми:

Програміст:

Людина з зростом 180 см, вагою 75 кг і віком 30 років, програміст зі знанням мов: PHP, JavaScript, досвід роботи: 8 років.
Програміст народив дитину!

Студент:

Людина з зростом 170 см, вагою 60 кг і віком 20 років, студент ВНЗ КНУ, курс 3.
Студент народив дитину!

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 програми(HouseCleaning)

```
<?php
```

```
namespace Interfaces;
```

```
interface HouseCleaning
```

```
{
```

```
    public function cleanRoom();
```

```
    public function cleanKitchen();
```

```
}
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

Лістинг програми(Human)

```
<?php
```

```
namespace Models;
```

```
use Interfaces\HouseCleaning;
```

```
abstract class Human implements HouseCleaning
```

```
{
```

```
    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;
```

```
    }
```

```
    public function setHeight($height)
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

```

{
    $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 birthAnnouncement();

```

	public function giveBirth()				Арк.
		ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4			28
Змн.	Арк.	№ докум.	Підпис	Дата	

```

{
    $this->birthAnnouncement();
}
}

```

Лістинг програми(Student)

<?php

namespace Models;

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;
    }

```

| | | | | | | |
|------|------|----------|--------|------|---------------------------------------------|------|
| | } | | | | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4 | Арк. |
| | | | | | | 29 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

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

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

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

```
public function nextCourse()
{
    $this->course += 1;
}
```

```
protected function birthAnnouncement()
{
    echo "Студент народив дитину!<br>";
}
```

| | | | | | |
|------|-----------------------------|---------------------------------------------|--------|------|------|
| | public function cleanRoom() | | | | Арк. |
| | | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4 | | | 30 |
| Змн. | Арк. | № докум. | Підпис | Дата | |

```

{
    echo "Студент прибирає кімнату.<br>";
}

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

public function __toString()
{
    return "Student: " . $this->getHeight() . " см, " . $this->getWeight() . " кг, " . $this->getAge() . " років, Університет: " . $this->university . ", Курс: " . $this->course;
}
}

```

Лістинг програми(Programmer)

```
<?php
```

```
namespace Models;
```

```
class Programmer extends Human
```

```

{
    private $languages;
    private $experience;
}

```

| | | | | | | |
|------|------|----------|--------|------|---------------------------------------------|------|
| | | | | | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 31 |

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

| | | | | | | |
|------|------|----------|--------|------|---------------------------------------------|------|
| | | | | | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 32 |


```
public function addLanguage($language)
```

```
{
```

```
    $this->languages[] = $language;
```

```
}
```

```
protected function birthAnnouncement()
```

```
{
```

```
    echo "Програміст народив дитину!<br>";
```

```
}
```

```
public function cleanRoom()
```

```
{
```

```
    echo "Програміст прибирає кімнату.<br>";
```

```
}
```

```
public function cleanKitchen()
```

```
{
```

```
    echo "Програміст прибирає кухню.<br>";
```

```
}
```

```
public function __toString()
```

```
{
```

```
    return "Programmer: " . $this->getHeight() . " см, " . $this->getWeight() . " кг, " .  
    $this->getAge() . " років, Мови програмування: " . implode(', ', $this->languages) . ",  
    Досвід: " . $this->experience . " років";
```

```
}
```

```
}
```

| | | | | | | |
|------|------|----------|--------|------|---------------------------------------------|------|
| | | | | | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 33 |

Лістинг програми(index.php)

```
<?php
```

```
require_once 'D:\WampServer\www\lab4\autoload.php';
```

```
use Models\Human;
```

```
use Models\Student;
```

```
use Models\Programmer;
```

```
$student = new Student(180, 75, 25, 'Київський національний університет', 2);
```

```
echo $student;
```

```
echo "<br>";
```

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

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

```
echo "<br>";
```

```
$programmer = new Programmer(175, 70, 30, ['PHP', 'JavaScript'], 5);
```

```
echo $programmer;
```

```
echo "<br>";
```

```
$programmer->cleanRoom();
```

```
$programmer->cleanKitchen();
```

```
echo "<br>";
```

| | | | | | | |
|------|------|----------|--------|------|---------------------------------------------|------|
| | | | | | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 34 |

Результат виконання роботи:

Student: 180 см, 75 кг, 25 років, Університет: Київський національний університет, Курс: 2
Студент прибирає кімнату.
Студент прибирає кухню.

Programmer: 175 см, 70 кг, 30 років, Мови програмування: PHP, JavaScript, Досвід: 5 років
Програміст прибирає кімнату.
Програміст прибирає кухню.

Посилання на github: <https://github.com/loorkky/Backend.git>

| | | | | | | |
|------|------|----------|--------|------|---------------------------------------------|------|
| | | | | | ЖИТОМИРСЬКА ПОЛІТЕХНІКА.24.121.05.000 –Лр.4 | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 35 |