

### **Практична робота №3**

**Тема:** Оголошення та виклик функцій у PHP 8.1: від класичних до стрілкових.

**Мета:** Відрізнювати синтаксис функцій у PHP, вивчити типізацію параметрів, значення за замовчуванням та анонімні функції.

### **КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ**

У PHP 8.1 функції є основними інструментами структурування коду. PHP дозволяє суверо вказувати типи вхідних даних та результату.

#### **1. Класичне оголошення функції (Function Declaration)**

У PHP функції зазвичай оголошуються за допомогою ключового слова `function`.

**PHP**

```
function getAreaDeclaration(int $width, int $height): int
{
    return $width * $height;
}
echo getAreaDeclaration(5, 10); // 50
```

**Hoisting:** У PHP класичні функції також можна викликати до їх опису (якщо вони не оголошенні всередині умовних блоків).

**Типізація:** У прикладі вказано, що параметри – це цілі числа (`int`), і результат теж буде `int`.

#### **2. Анонімні функції (Function Expression)**

У PHP анонімні функції (закриття) – це об'єкти класу `Closure`. Вони часто записуються у змінні.

**PHP**

```
$getAreaExpression = function(int $width, int $height): int {
    return $width * $height;
}
echo $getAreaExpression(5, 10); // 50
```

**Важливо:** Наприкінці виразу з анонімною функцією обов'язково ставиться крапка з комою `,`.

**Доступ до змінних:** Щоб використати змінну з глобальної області всередині анонімної функції, треба використовувати конструкцію `use ($variable)`.

#### **3. Стрілкові функції (Arrow Functions)**

З'явилися в PHP 7.4 і отримали розвиток у PHP 8.0+. Вони дуже короткі та автоматично мають доступ до змінних з батьківської області видимості.

### PHP

```
$getAreaArrow = fn(int $width, int $height): int => $width  
* $height;  
  
echo $getAreaArrow(5, 10); // 50
```

**Особливість:** Можуть містити лише **один** вираз (вираз повернення). Ключове слово `return` писати не потрібно.

### Параметри та аргументи

#### Параметри за замовчуванням

PHP дозволяє встановлювати значення за замовчуванням. Якщо аргумент не передано, буде використано стандартне значення.

### PHP

```
function myFunction(int $x, int $y = 10): int {  
    return $x + $y;  
}  
echo myFunction(5); // 15 (у стало 10)
```

#### Параметр залишки (*Variadic Functions*)

Використовується оператор `...` для збору довільної кількості аргументів у масив.

### PHP

```
function sum(int ...$numbers): int {  
    return array_sum($numbers);  
}  
echo sum(4, 9, 16, 25); // 54
```

## ЗАВДАННЯ ДЛЯ САМОСТІЙНОГО ВИКОНАННЯ

### Завдання 1. Синтаксичний практикум

Напишіть функцію обчислення площині круга  $S = \pi r^2$  трьома способами. Для числа  $\pi$  використовуйте вбудовану константу `M_PI`.

1. **Function Declaration** з типізацією `float`.
2. **Anonymous Function (Expression)**.
3. **Arrow Function**.

### Завдання 2. Параметри за замовчуванням

Створіть функцію вітання `sayHello(string $name, string $greeting = "Привіт")`.

Викличте функцію, передавши два імені.

Викличте її, передавши лише одне ім'я (використовуючи значення за замовчуванням).

### **Завдання 3. Область видимості (Scope)**

У PHP глобальні змінні не доступні всередині функцій автоматично.

Створіть глобальну змінну \$university = "НУБіП".

Спробуйте вивести її всередині функції.

Використовуйте ключове слово `global` всередині функції, щоб отримати до неї доступ, та продемонструйте різницю з локальними змінними.

### **Завдання 4. Функція-колбек (Callback)**

Напишіть функцію `processNumbers(int $a, int $b, callable $callback)`.

Створіть дві окремі стрілкові функції: `$add` та `$multiply`.

Передайте їх у `processNumbers` та виведіть результат.

*Підказка:* У PHP тип `callable` дозволяє передавати назви функцій як рядки або змінні, що містять анонімні функції.