

OOP hook cares

Piotr
Niewiadomski
PIWIK PRO Developer

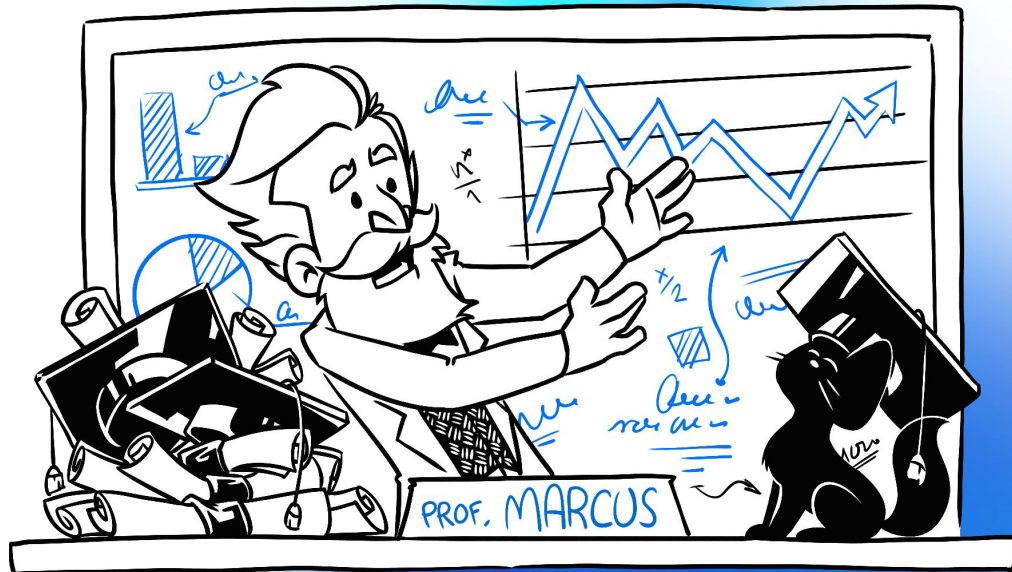


Agenda

- ① Podstawy
- ② OOP



Podstawy



“Haki umożliwiają interakcję/modyfikację innego fragmentu kodu w określonych, wcześniej zdefiniowanych miejscach.”

Event-Driven Architecture

Observer / Publisher-Subscriber Pattern

Haki = Akcje + Filtry



Akcje - pozwalają na wykonanie określonych działań w konkretnym momencie cyklu przetwarzania żądania



Filtry - pozwalają na modyfikację danych w konkretnym momencie cyklu przetwarzania żądania

Przykładowe Akcje

- muplugins_loaded
- plugins_loaded
- setup_theme
- init
- parse_request
- parse_query
- shutdown
- wp_loaded
- wp_enqueue_scripts
- wp_head
- wp_print_styles
- wp_print_scripts
- the_post
- wp_footer
- ...

Przykładowe Filtry

- body_class
- the_content
- the_excerpt
- the_tags
- the_category
- the_title
- the_permalink
- stylesheet
- template
- wp_redirect
- posts_results
- query
- request
- option_{option name}
- ...

Zahaczenie akcji

```
function add_action(  
    string $hook_name,  
    callable $callback,  
    int $priority = 10,  
    int $accepted_args = 1  
) : true
```


Zahaczanie filtrów

```
function add_filter(  
    string $hook_name,  
    callable $callback,  
    int $priority = 10,  
    int $accepted_args = 1  
) : true
```

Akcje / Filtry

```
function add_action(  
    string $hook_name,  
    callable $callback,  
    int $priority = 10,  
    int $accepted_args = 1  
) : true
```

```
function add_filter(  
    string $hook_name,  
    callable $callback,  
    int $priority = 10,  
    int $accepted_args = 1  
) : true
```

Akcje = Filtry

```
function add_action(  
    string $hook_name,  
    callable $callback,  
    int $priority = 10,  
    int $accepted_args = 1  
) : true
```

```
function add_filter(  
    string $hook_name,  
    callable $callback,  
    int $priority = 10,  
    int $accepted_args = 1  
) : true
```

```
function add_action( $hook_name, $callback, $priority = 10, $accepted_args = 1 ) {  
    return add_filter( $hook_name, $callback, $priority, $accepted_args );  
}
```

Akcje **vs** Filtry



Akcje - funkcje “zahaczone” do akcji **nic nie zwracają**



Filtry - funkcje “zahaczone” do filtrów **zwracają zmodyfikowane dane**

Przykładowa Akcja

```
add_action(  
    'publish_post',  
    function( $post_id, $post ) {  
        wp_mail( 'kontakt@piotr.press', $post->get_title(), get_permalink( $post ) );  
    },  
    1,  
    2  
);
```

Przykładowa Akcja

```
add_action(  
    'publish_post', // $hook_name  
    function( $post_id, $post ) { // $callback  
        wp_mail( 'kontakt@piotr.press', $post->get_title(), get_permalink( $post ) );  
    },  
    1, // $priority  
    2 // $accepted_args  
);
```

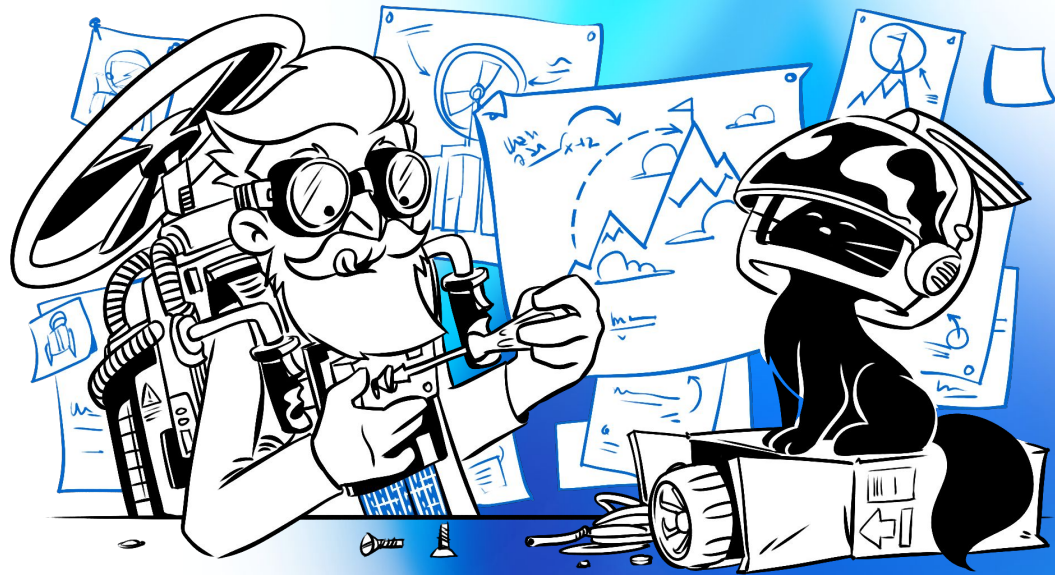
Przykładowy Filtr

```
add_filter(  
    'the_title',  
    function( $title ) {  
        return "<strong>$title</strong>";  
    }  
);
```

Przykładowy Filtr

```
add_filter(  
    'the_title', // $hook_name  
    function( $title ) { // $callback  
        return "<strong>$title</strong>";  
    }  
    // $priority = 10,  
    // $accepted_args = 1  
);
```


OOP



```
class Plugin {  
    public function send_mail( $post_id, $post ) : void {  
        wp_mail( 'kontakt@piotr.press', $post->get_title(), get_permalink( $post ) );  
    }  
  
    public function bold_title( $title ) : string {  
        return "<strong>$title</strong>";  
    }  
}  
  
$plugin = new Plugin();  
  
add_action( 'publish_post', [ $plugin, 'send_mail' ], 1, 2 );  
add_filter( 'the_title', [ $plugin, 'bold_title' ] );
```

```
class Plugin {
    public function __construct() {
        add_action( 'publish_post', [ $this, 'send_mail' ], 1, 2 );
        add_filter( 'the_title', [ $this, 'bold_title' ] );
    }

    public function send_mail( $post_id, $post ) : void {
        wp_mail( 'kontakt@piotr.press', $post->get_title(), get_permalink( $post ) );
    }

    public function bold_title( $title ) : string {
        return "<strong>$title</strong>";
    }
}
```

Wady

✘ Każdą metodę musimy zahaczyć ręcznie.

✘ Musimy ręcznie zdefiniować ilość argumentów metody.

**Zahaczenie
po nazwie**

```
class Plugin {
    public function __construct() { $this->add_hooks(); }

    public function action_publish_post_1( $post_id, $post ) : void {
        wp_mail( 'kontakt@piotr.press', $post->get_title(), get_permalink( $post ) );
    }

    public function filter_the_title( $title ) : string {
        return "<strong>$title</strong>";
    }
}
```

Wady

✘ Tylko jedna metoda per hak.

✘ Tylko jeden hak per metoda.

✘ Nie zahaczmy haków, które zawierają znaki specjalne np. **piotrpress/custom-action**

Zahaczenie po adnotacji


```
class Plugin {
    public function __construct() { $this->add_hooks();}

    /**
     * @action publish_post 1
     */
    public function send_mail( $post_id, $post ) : void {
        wp_mail( 'kontakt@piotr.press', $post->get_title(), get_permalink( $post ) );
    }

    /**
     * @filter the_title
     */
    public function bold_title( $title ) : string {
        return "<strong>$title</strong>";
    }
}
```

Wady

✘ Nie zadziała jeśli z naszego kodu zostaną usunięte komentarze np. **OPCache** z wyłączonymi metodami **save_comments** i **load_comments**

✘ Wolne API Refleksji i **parsowanie tekstu**.



Kudos: **Jakub Mikita**



<https://github.com/micropackage/dochooks>


Zahaczanie po atrybucie

```
class Plugin {
    public function __construct() { Hooks::add( $this ); }

    #[ Action( 'publish_post', 1 ) ]
    public function send_mail( $post_id, $post ) : void {
        wp_mail( 'kontakt@piotr.press', $post->get_title(), get_permalink( $post ) );
    }

    #[ Filter( 'the_title' ) ]
    public function bold_title( $title ) : string {
        return "<strong>$title</strong>";
    }
}
```

Wady

 Prędkość wykonania z uwagi na użycie **API Refleksji**.*

 Kudos: **Sebastian Pisula**

```
class Plugin {
    public function __construct() { Hooks::add( $this ); }

    #[ Action( 'publish_post', 1 ) ]
    public function send_mail( $post_id, $post ) : void {
        wp_mail( 'kontakt@piotr.press', $post->get_title(), get_permalink( $post ) );
    }

    #[ Filter( 'the_title' ) ]
    public function bold_title( $title ) : string {
        return "<strong>$title</strong>";
    }
}

$plugin = new Plugin();
Hooks::remove( $plugin );
```

```
class Plugin {
    #[ Action( 'publish_post', 1 ) ]
    public function send_mail( $post_id, $post ) : void {
        wp_mail( 'kontakt@piotr.press', $post->get_title(), get_permalink( $post ) );
    }

    #[ Filter( 'the_title' ) ]
    public function bold_title( $title ) : string {
        return "<strong>$title</strong>";
    }
}

$plugin = new Plugin();
Hooks::add( $plugin, 'send_mail' );
Hooks::remove( $plugin, 'bold_title' );
```

```
#[ Action( 'publish_post', 1 ) ]  
public function send_mail( $post_id, $post ) : void {  
    wp_mail( 'kontakt@piotr.press', $post->get_title(), get_permalink( $post ) );  
}
```

```
#[ Filter( 'the_title' ) ]  
public function bold_title( $title ) : string {  
    return "<strong>$title</strong>";  
}
```

```
Hooks::add( callback: 'send_mail' );  
Hooks::remove( callback: 'bold_title' );
```


Feature requests

👍 Kudos: **Mateusz Gbiorczyk**

👍 Kudos: **Krzysztof Grabania**

👍 Kudos: **Dominik Kawula**



Źródła

 <https://github.com/PiotrPress/wordpress-hooks>

- **v4.x** PHP ≥ 7.4 - wszystko
- **v5.x** PHP ≥ 8.0 - najnowsze



Dziękuję za uwagę

PIWIK **PRO**



@ kontakt@piotr.press

 <https://github.com/PiotrPress>

 <https://profiles.wordpress.org/PiotrPress>

