**4.2.4 Особливості розробки алгоритмів методів програмних класів або процедур/функцій**

Класичний клас

Це найбільш частий і зрозумілий метод. Використовується метод \_\_construct () для створення екземпляра класу. Але він же найбільш проблемний через те що багато програмісти можуть його неправильно готувати.

Приклад:

class MyPlugin1

{

public function \_\_construct()

{

add\_action('plugins\_loaded', array($this, 'loaded'));

}

public function loaded()

{

// do stuff

}

}

Cтатичний клас

Цей підхід використовую найчастіше. За його гнучкість і можливість перемикання в інші стилі і підходи. Часто такі механіки можна зустріти в WooCommerce.

Приклад:

class My\_Plugin2

{

public static function init ()

{

add\_action ('plugins\_loaded', array (\_\_ CLASS\_\_, 'loaded'));

}

public static function loaded ()

{

// do stuff

}

}

My\_Plugin2 :: init ();

Сінглтон

Часто застосовуються там де потрібно гарантовано мати один екземпляр об'єкта в системі без дублювання.

Плюс більш передбачувана і зрозуміла робота зі станом об'єкта ніж у Статичної підходу.

Бував два типу старту: через init () або instance ()

Приклад:

class MyPlugin3

{

private static $ ins;

public static function init ()

{

add\_action ('plugins\_loaded', array (\_\_ CLASS\_\_, 'instance'));

}

public static function instance ()

{

is\_null (self :: $ ins) && self :: $ ins = new self;

return self :: $ ins;

}

private function \_\_construct ()

{

add\_action ('plugins\_loaded', array ($ this, 'loaded'), 20);

}

public function loaded ()

{

// do stuff

}

}