## Міністерство освіти і науки України

## Запорізький національний технічний університет

кафедра програмних засобів

# реферат

з дисципліни «Якість програмного забезпечення» на тему:

# «PHPunit»

Виконав:

студент групи КНТ-115 Л.І. Кулик

Прийняла:

проф., д.т.н Г.В.Табунщик

2018

## ВСТУП

PHPUnit — фреймворк для модульного тестування під час розробки ПЗ на PHP. Є представником сімейства фреймворків XUnit на основі пакету SUnit, створеного Кентом Беком. PHPUnit розроблений Себастьяном Бергманом.

PHPUnit був створений з позиції - чим раніше ви виявите помилки в коді, тим швидше ви зможете їх виправити. Як і всі фреймворки модульного тестування PHPUnit використовує твердження для перевірки, що поведінка одиниці тестованого коду поводиться як і очікувалося.

# Різновиди тестів

Перед тим, як розглядати PHPUnit необхідно розділити тести на категорії за рівнем їх специфічності., щоб краще зрозуміти можливості фреймворку для тестування. За даними Вікіпедії. В цілому існує 4 визнаних рівня тестів:

Unit-тестування (модульне): цей рівень тестує найменшу одиницю функціональності. З точки зору розробника його завданням є переконатися, що тестована функція виконує саме те, для чого вона реалізована. Таким чином, вона повинна бути мінімально залежна або абсолютно незалежна від іншої функції або класу. Вона повинна бути написана таким чином, щоб вона повністю виконувалася в пам'яті, тобто вона не повинна коннектіться до БД, не повинна звертатися до мережі або використовувати ФС і т.д. Unit-тестування має бути якомога простішим.

Інтеграційне тестування: цей рівень "з'єднує" різні одиниці коду і тестує правильно працюють їх комбінації. Він надбудовується зверху над unit-тестуванням і здатний відловити баги, які не можна виявити за допомогою unit-тестування, тому що інтеграційне тестування перевіряє, чи працює клас А з класом Б.

Системне тестування: воно створено для відтворення роботи сценаріїв в умовах, наближених до бойових. Воно, в свою чергу, надбудовується зверху над інтеграційним тестуванням. У той час як інтеграційне тестування забезпечує злагоджену роботу різних частин системи. Системне тестування відповідає за те, що система працює так, як передбачає користувач, перш ніж відправити її наступний рівень.

Приймальнe тестування: в той час як вище наведені тести призначені для розробників на стадії розробки, приймальним тестування фактично виконується користувачами ПЗ. Користувачів не цікавлять внутрішні особливості ПЗ, їх цікавить тільки як працює це ПЗ[1].

# Опис PHPUnit

Одна з цілей PHPUnit полягає в тому, що тести повинні бути складовими: ми хочемо запускати будь-яку кількість або комбінацію тестів разом, наприклад, всі тести для всього проекту, або тести всіх класів компонента, який є частиною проекту, або просто тести для одного класу.

PHPUnit підтримує різні способи організації тестів і складання їх у набір тестів.

Можливо, найпростіший спосіб скласти набір тестів - це тримати всі вихідні файли тестів в тестовому каталозі. PHPUnit може автоматично виявляти і запускати тести шляхом рекурсивного обходу тестового каталогу.

XML-файл конфігурації PHPUnit (Конфігураційний XML-файл) також може використовуватися для складання набору тестів.

PHPUnit може генерувати звіт про покриття коду на основі HTML, а також лог-файли в поданні XML з інформацією про покриття коду в різних форматах (Clover, Crap4J, PHPUnit). Інформація про покриття коду також може бути представлена у вигляді тексту (і надрукована в STDOUT) і експортована як код PHP для подальшої обробки.

PHPUnit можна розширити різними способами для полегшення процесу написання тестів і настройки зворотного зв'язку, одержуваної від виконання тестів[2].

Приклад тесту:

**class** **StackTest** **extends** TestCase

{

**public** **function** testPushAndPop()

{

$stack = [];

$this->assertSame(0, count($stack));

array\_push($stack, 'foo');

$this->assertSame('foo', $stack[count($stack)-1]);

$this->assertSame(1, count($stack));

$this->assertSame('foo', array\_pop($stack));

$this->assertSame(0, count($stack));

}

}

## ПЕРЕЛІК ПОСИЛАНЬ

1. Sofware testing: [Електронний ресурс] – режим доступу: https://en.wikipedia.org/wiki/Software\_testing#Testing\_levels
2. PHPUnit manual: [Електронний ресурс] – режим доступy: <https://phpunit.readthedocs.io>