

ТЕХНОЛОГІЇ РОЗРОБКИ ТА СУПРОВОДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

УДК 004. 413

Костів М.М., Крамар Ю.М.

Національний авіаційний університет

ІНСТРУМЕНТ ДЛЯ СТВОРЕННЯ СТИЛЮ ЕФЕКТИВНОГО ПРОГРАМУВАННЯ

У статті розглянута задача створення інструменту необхідного для аналізу тексту програми при розробці та застосуванні стилю ефективного програмування, описаний алгоритм функціонування інструменту та його архітектура, представлені результати експериментів, що проведені з використанням різних проектів з відкритим кодом, дослідження виконувалось для мови програмування PHP.

В статье рассмотрена задача создания инструмента необходимого для анализа текста программы при разработке и применении стиля эффективного программирования, описан алгоритм функционирования инструмента и его архитектура, представлены результаты экспериментов, проведенных с использованием различных проектов с открытым кодом, исследование проводилось для языка программирования PHP.

Considered is the tool for code analyses for application of developed style for effective programming, the architecture of the tool and the algorithm of functionality is described, the results of experiments, which are conducted with using of different projects with open source code, research was conducted for programming language PHP.

Ключові слова: стиль програмування, ефективно програмування, реверсивна інженерія, операторний базис, конструкція мови програмування.

Вступ

У 1981 році компанія ІВМ випустила в продаж перший персональний комп'ютер, де був використаний процесор Intel 8088, що працював на тактовій частоті 4,77 МГц.

Зараз у персональних комп'ютерах використовують чотирьохядерні процесори Intel ® Core i7 з тактовою частотою 3,46 ГГц. Отже швидкість обробки інформації зросла більш ніж в 700 разів. Обчислення, що займали в 1981 році більше 10 хвилин, тепер можна виконати менш ніж за секунду [1].

Сучасне програмне забезпечення має більше функціональних можливостей і складніше за прості застосування з текстовим інтерфейсом. Від навчальних застосувань до баз даних і операційних систем — скрізь потрібно більше обчислювальних ресурсів, ніж раніше, і сьогоднішньому програмному забезпеченню їх також не вистачає [2].

Якщо застосування використовує клієнт-серверну модель обробки даних і взаємодіє з користувачем, то критеріями оптимізації будуть: використання пам'яті (особливо для серверної частини застосування) і швидкість мережевої взаємодії з клієнтською частиною [1].

Першу частину проблеми можна вирішити написанням більш ефективних програмних

рішень. Друга частина проблеми вирішується лише на фізичному рівні — встановленням серверів з більш швидким доступом в мережу Інтернет та малою затримкою.

Найпростіший спосіб змусити Web-застосування працювати швидше — підвищити обчислювальну потужність сервера шляхом установки більш продуктивного процесора і збільшення об'єму пам'яті. У цьому випадку проблема продуктивності буде вирішення сама собою [2].

Іншим способом є використання ефективного стилю програмування для підвищення продуктивності роботи застосування з використанням вже існуючих ресурсів без доповнення або заміни елементів апаратної частини [3]. Цієї ефективності можна досягти двома способами. По-перше, вибором ефективного алгоритму, по-друге, використанням ефективного стилю програмування.

При роботі з Web-застосуванням користувачі перш за все сприймають те, як швидко Web-застосування обробляють запити і відображають результати на екрані, тобто наскільки швидко відбувається діалог.

Використання стилю ефективного програмування надає можливість збільшити швидкість такого діалогу.

Постановка завдання

Стилі програмування використовуються на етапах проектування, написання, кодування і тестування програм. Контроль стилю мови програмування - це процеси перевірки використання стилю мови програмування при написанні текстів програм. Необхідність виконання процесів виникає, коли потрібно визначити відповідність стилю програми заданому стилю мови програмування, яким розробляється ПО, а також при навчанні написання програм [5].

Дотримання стилю ефективного програмування особливо важливе при колективній роботі над великими проектами. Однак при створенні навіть невеликих проектів дотримання певного стилю дозволяє істотно зменшити час їх розробки і зробити програми більш ефективними.

Протестувати програми і перевірити їх на дотримання стилю можна вручну або за допомогою розроблених для цієї мети інструментів.

Програмісти можуть самостійно створювати правила і застосовувати їх при написанні програм, здійснюючи при цьому індивідуальний або колективний контроль усього процесу розробки і дотримання стилю. Іноді експерти перевіряють проекти на дотримання стилю [6].

Але більш ефективним буде метод створення спеціального інструменту, тобто програми для автоматичного пошуку тексту коду, який не відповідає стилю.

Розв'язання завдання

У даній роботі описано інструмент для дослідження дотримання стилю програмування в мові PHP, який орієнтований на ефективність програмування з точки зору швидкості виконання конструкцій мови програмування. Для створення інструменту необхідно визначити ефективні конструкції і сформулювати правила стилю ефективного програмування, які будуть використані для аналізу коду проекту і створення рекомендацій (Рис. 1).

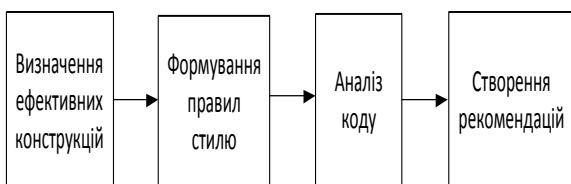


Рис. 1 Алгоритм роботи інструменту

Для формування стилю визначено ефективні і неефективні конструкції, які згруповані відповідно до задач, які вони вирішують та представлені у вигляді таблиць. (Таблиця 1)

Таблиця 1
Конструкції для роботи з масивами

Задача	Ефективна конструкція ✓	Неефективна конструкція ✗
Перевірка масиву на пустоту	<code>if (empty(\$array)) { }</code>	<code>if(count(\$array) == 0){ }</code>
Прохід по асоціативному масиву	Конструкція з використанням <code>foreach</code>	Конструкція з використанням <code>for</code>
Доступу до елемента масиву	<code>\$array['key']</code>	<code>\$array[key]</code>

Стиль ефективного програмування створено на основі правил щодо використання ефективних конструкцій замість неефективних для вирішення певних задач.

Наприклад, для першої задачі таке правило:

- для перевірки масиву на пустоту необхідно використовувати ефективну конструкцію `if(empty($array))` замість `if(count($array) == 0)`.

Також слід використовувати ефективні конструкції для роботи з рядками (Таблиця 2) і змінними (Таблиця 3).

Для дослідження краще вибрати найбільші файли проекту, оскільки імовірність того, що аналізатор знайде повільні конструкції і створить необхідні рекомендації для підвищення ефективності роботи програми, тоді безперечно зростає.

Таблиця 2
Конструкції для роботи з рядками

Задача	Ефективна конструкція ✓	Неефективна конструкція ✗
Перевірка довжини	<code>if(!isset(\$lenght){ }</code>	<code>if(strlen(\$str)<\$lenght){ }</code>
Для виводу значення із змінною	Конструкція з конкатенацією <code>echo 'hello' . \$var</code>	Конструкція без конкатенації <code>echo "hello \$var"</code>

Таблиця 3
Конструкції для роботи зі змінними

Задача	Ефективна конструкція ✓	Неефективна конструкція ✗
Перевірка значення змінної на рівність TRUE	<code>if (\$var){}</code>	<code>if (true == \$var){};</code>
Для виводу значення із змінною	<code>if(\$var===null){}</code>	<code>if(is_null(\$var)){};</code>

Інструмент створено з використанням мови програмування PHP, і він працює в усіх сучасних браузерах.

Розглянемо архітектуру засобу (Рис. 2) і відповідно діаграму класів (Рис. 3).



Рис.2. Архітектура інструменту

У структурі проекту використані наступні класи: AnalyzeBuilder, FileAnalyzer, Configurator, TimeAnalyzer, UiBuilder (Рис.3).

Інструмент не тільки аналізує код на дотримання ефективного стилю програмування, а також допомагає формувати такий стиль, вимірюючи час виконання конструкцій і визначаючи їх ефективність (клас TimeAnalyzer). Клас FileAnalyzer аналізує завантажений користувачем файл з PHP кодом, визначає у ньому рядки, які містять повільні конструкції і формує відповідні рекомендації до підвищення ефективності роботи програми.

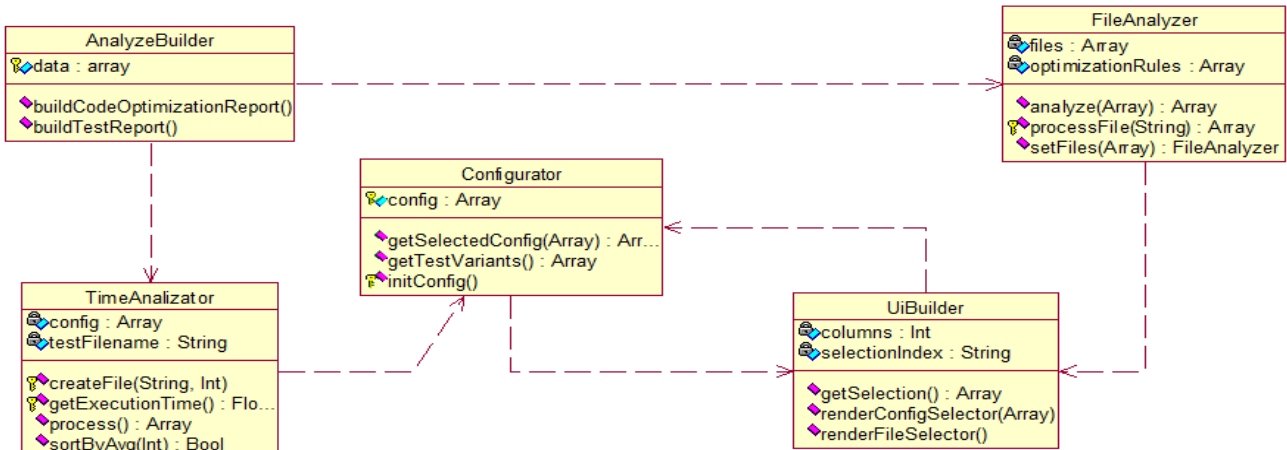


Рис.3. Діаграма класів програми

Клас AnalyzeBuilder відображає результати на екрані у вигляді таблиці після закінчення аналізу, що виконується в класі FileAnalyzer, або після вимірювання часу виконання конструкцій класом TimeAnalyzer.

Клас Configurator призначений для роботи з усіма конструкціями, які знаходяться у файлі config.php.

Для вибору файлу проекту використовується інтерфейс зображений на Рис.4.

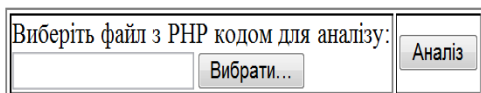


Рис.4. Інтерфейс вибору файлів для аналізу.

Після того, як був вибраний потрібний файл, аналізатор має виконати його дослідження, використовуючи порівняння з базою даних (сформованих правил стилю), і створити файл з кодом, який містить рекомендації по дотриманню стилю орієнтованого на ефективність.

Створений інструмент виконує аналіз коду у вибраному проекті і виділяє в ньому ті рядки, які містять неефективні конструкції, що порушують правила стилю.

Інструмент працює з файлами проекту, які можуть бути вибрані користувачем і тому, потрібно, наприклад, обрати файли RHPExcel.php та RHPDisp.php (Рис. 5) з проекту RHPExcel [7] для аналізу.

На Рис. 5 зображений результат аналізу коду, де у першій колонці відображується код файлу, а у другій наведені правила стилю для підвищення ефективності відповідних рядків коду.

Код файлу PHP	Рекомендації
<pre> public function addSheet(PHPExcel_Worksheet \$pSheet, \$iSheetIndex = NULL) { if (\$this->sheetNameExists(\$pSheet->getTitle())) { throw new PHPExcel_Exception("Workbook already contains a worksheet named '{\$pSheet->getTitle()}'. Rename this worksheet first."); } if(\$iSheetIndex === NULL) { if (\$this->activeSheetIndex < 0) { \$this->activeSheetIndex = 0; } \$this->workSheetCollection[] = \$pSheet; } else { // insert the sheet at the requested index array_splice(\$this->workSheetCollection, \$iSheetIndex, 0, array(\$pSheet)); } } /** * This workbook have additionnal object for the ribbon ? * @return true false */ public function hasRibbonBinObjects(){ return !is_null(\$this->_ribbonBinObjects); } </pre>	<p>Замість змінної у подвійних лапках використовуйте конкатенацію</p> <p>Для перевірки на рівність значення змінної NULL використовуйте конструкцію <code>if(\$var===null){}</code> замість <code>if(is_null(\$var)){};</code></p> <p>Для перевірки існування елемента асоціативного масиву за індексом використовуйте функцію <code>isset</code> з вказуванням індексу елемента <code>isset(\$array['key'])</code> замість функції <code>array_key_exists('key', \$array);</code></p>
<pre> if(is_array(\$this->_ribbonBinObjects) && array_key_exists('data', \$this->_ribbonBinObjects) </pre>	

Рис. 5. Аналіз коду файлу PHPDisp.php

У вихідному файлі рядки з повільними конструкціями виділяються кольором для того, щоб привернути до себе увагу користувача.

Інструмент також перевірить файл на відповідність створеному стилю програмування.

Висновки

Використання стилю ефективного програмування збільшує продуктивність роботи застосування з використанням вже існуючих ресурсів без доповнення або заміни елементів апаратної частини.

Функціональність інструменту можна розширити шляхом створення нових правил стилю для PHP та інших мов програмування.

Список використаних джерел

1. Гербер Р. Оптимізація ПО. Сборник рецептов / Р. Гербер, А. Бик, К. Смит, К. Тиан. – С-Пб.: Питер, 2010. – 352 с.

2. Магда Ю.С. Ассемблер. Разработка и оптимизация Windows-приложений / Ю.С. Магда – С-Пб.: БХВ-Петербург, 2003. – 544 с.

3. Сидоров М.О. Створення стилю ефективного програмування / М.О. Сидоров, М.М. Костів // Інженерія програмного забезпечення. – № 3 (15). – 2013. – С. 7 – 12.

4. Дж. Макконнелл. Основы современных алгоритмов / Макконнелл Дж. – М.: Техносфера, 2004. – 368 с.

5. Крамар Ю.М. Средства для автоматизированного синтеза стилей программирования / Ю.М. Крамар // Вісник НАУ. – 2002. – №2 (13). – С. 52 – 60.

6. Сидоров Н.А. Стилистика программирования / Н.А. Сидоров // Проблеми інформатизації та управління: збірник наукових праць. – 2003. – Вип. 8. – С. 204 – 207.

7. PHPExcel [Електронний ресурс] – Режим доступа: <https://phpexcel.codeplex.com/>

Відомості про авторів:



Костів Мілана Миколаївна – студентка 6 курсу Інституту комп'ютерних інформаційних технологій Національного авіаційного університету. Наукові інтереси: інженерія програмного забезпечення.

E-mail: milana.kostiv@livenau.net



Крамар Юлія Михайлівна – к.т.н., доцент кафедри інженерії програмного забезпечення Національного авіаційного університету. Наукові інтереси: інженерія програмного забезпечення, стилістика програмування.

E-mail: yulia.kramar@livenau.net