

Список використаних джерел

1. Архитектуры и топологии многопроцессорных вычислительных систем /А.В. Богданов, В.В. Корхов, В.В. Мареев, Е.Н. Станкова/ — М.: ИНТУИТ.РУ «Интернет-Университет Информационных Технологий», 2004. — 176 с.
2. M. Ghanbari, Standard Codecs: Image Compression to Advanced Video Coding, The Institution of Engineering and Technology, 2003

УДК 004.4

ДОСЛІДЖЕННЯ КІЛЬКІСНИХ ТА ЯКІСНИХ ПОКАЗНИКІВ АСПЕКТНО-ОРІЄНТОВАНОГО ПІДХОДУ ДО РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Гончар Л.І.¹⁾, Кіндзера Ю.Р.²⁾

Тернопільський національний економічний університет

¹⁾ к.е.н., доцент; ²⁾ магістрант

Аспектно-орієнтоване програмування (АОП) доповнює об'єктно-орієнтоване програмування, збагачуючи його іншим типом модульності, який дозволяє локалізувати код реалізації - наскрізної логіки в одному або декількох модулях, які називаються аспектами. За рахунок відділення аспектно-орієнтованого коду робота із наскрізною функціональністю спрощується. Аспекти в системі можуть легко змінюватися при необхідності розв'язання нової задачі, більше того, повторне використання аспектів зменшує затрати часу, а відповідно і всіх ресурсів, для реалізації змін та нових функцій програмних додатків.

Предметом дослідження є механізми, засоби, технології, результати впровадження та застосування аспектно-орієнтованого підходу в об'єктно-орієнтованих програмах. В якості прикладу, на якому буде досліджено переваги та недоліки використання АОП, в роботі використана прикладна програма, побудована з використанням клієнт-серверних технологій, що є актуальним в час розвитку мережі Інтернет.

У рамках наукової роботи було проведено порівняльне дослідження статистичних характеристик програми з певним набором функцій, яка була написана з використанням ООП-підходу, а згодом переписана з використанням АОП -конструкцій. Важливо відмітити, що набір функцій програми не змінювався, дві версії виконують один і той самий набір функцій. Відмінністю другої версії є виключно винесення наскрізної логіки в аспекти.

Програма представляє собою інформаційну систему для збору інформації про організації та підприємства, їх географічні координати, опції пошуку по різним параметрам та послідовним відображенням координат на карті.

Таблиця 1

Порівняння кількісних характеристик коду програми, реалізованої за допомогою ООП-підходу та програми з застосуванням АОП-підходу

	Кількість рядків коду	Кількість класів	Кількість змінних	Дублювання коду (%)	Складність*
ООП	5265	104	1508	11.9	1931
ООП + АОП	2516	110	1245	0.5	774

*Складність – одиниця вимірювання на основі стандартного набору метрик, які використовує програмне забезпечення для статистичного аналізу коду SonarQube.

Судячи з порівняльної оцінки (Таб. 1) коду двох програм з однаковою функціональністю, але різними підходами, можна сказати, що використання АОП суттєво покращує характеристики коду. При винесенні наскрізної логіки в окремі модулі спостерігається зменшення об'єму коду на 52.2%, зменшення кількості змінних на 17.4%, зменшення складності коду на 59.5%, зменшення відсотку дублювання коду на 11.4%.

Із отриманих даних можна зробити висновок, що характеристики коду покращились на 35.1%. Такий високий відсоток доводить, що використання АОП вирішує основний недолік об'єктно-орієнтованого програмування, а саме, неможливість структурування наскрізної логіки.

Якщо ж не брати до уваги кількісні характеристики, то за результатами порівняння можна помітити покращення архітектури програми. Так як важлива частина функцій, які раніше були розкидані по всіх модулях, тепер знаходяться в окремих модулях, стало легше вносити зміни до такого виду функціональності. За сукупністю змін та покращень програму стало легше змінювати та супроводжувати, що в результаті приводить до менших затрат часу на розробку, а відповідно, і здешевленню розробки.

Окрім кількісного аналізу коду також було проведено порівняння часу виконання окремих функцій програми. Заміри часу виконання проводились з моменту початку запиту до отримання відповіді з результуючою інформацією (таблиця 2,3).

Таблиця 2

Час запиту функцій програми, реалізованої з використанням ООП-підходу

Опція/Кількість потоків	2	4	6	8	10	12	14	16	18	20
Пошук	25	26	29	31	35	41	47	56	62	69
Отримання координат всіх організацій	217	252	363	446	573	806	995	1272	1480	1724
Збереження організації	37	37	41	46	50	58	64	78	85	94
Отримання даних про всі організації	326	357	481	574	711	967	1176	1489	1719	1989
Кількість помилок	0	0	0	0	0	0	0	0	0	0

Таблиця 3

Час запиту функцій програми, реалізованої з використанням ООП та АОП підходів

Опція/Кількість потоків	2	4	6	8	10	12	14	16	18	20
Пошук	20	24	26	29	30	41	43	55	61	67
Отримання координат всіх організацій	200	240	350	420	560	750	911	1155	1201	1687
Збереження організації	33	34	39	42	48	52	58	75	79	86
Отримання даних про всі організації	315	349	460	564	701	943	1098	1489	1697	1915
Кількість помилок	0	0	0	0	0	0	0	0	0	0

Після застосування до об'єктно-орієнтованої програми аспектно-орієнтованого підходу, зменшився час обробки запитів користувача. Скорочення часу виконання зумовлено зменшенням

кількості змінних, а відповідно, і зменшенням об'єму пам'яті, яка використовується для роботи програми. На скорочення часу обробки запитів вплинуло і винесення функцій по обробці винятків в окремі модулі. Зменшився загальний відсоток дублювання коду, а відповідно, і затрати системи на його виконання.

Висновок

Варто відмітити, що суттєве скорочення коду відбулось на сервісних рівнях та рівні контролера. Судячи з цього можна сказати, що впровадження АОП не впливає на структури даних, якими оперує програма, воно тільки скорочує процедуру їх обробки та бізнес-логіку програми. Той факт, що впровадження АОП не вплинуло на структури даних, свідчить і про те, що використання АОП тільки доповнює ООП - програму, та не унеможливило використання принципів ООП.

Список використаних джерел:

1. Spring Framework Documentation, [Електронний ресурс]. – Режим доступу: <http://docs.spring.io/>.
2. AOP@Work: Мифы и реальности АОП, [Електронний ресурс]. – Режим доступу: <https://www.ibm.com/>.
3. Аспектно-Ориентированное Программирование в Spring, [Електронний ресурс].
4. Знакомство с АОП, [Електронний ресурс]. – Режим доступу: <http://habrahabr.ru/post/114649/>.
5. Г. Буч. Объектно-ориентированный анализ и проектирование / Г. Буч. – СПб.: Издательство Бинум, Невский диалект, 1998. – 560 с.

УДК 004.4

ІНЖЕНЕРІЯ МЕТОДІВ РОЗРОБКИ ВЕБ-ДОДАТКІВ ЗАСОБАМИ JAVA STANDART EDITION

Гончар Л.І.¹⁾, Кохан І.І.²⁾

Тернопільський національний економічний університет

¹⁾ к.е.н., доцент; ²⁾ магістрант

Недоліком використання Java Enterprise Edition є залежність від великих фреймворків, в яких закладено набагато більше логіки, ніж потрібно для конкретного програмного продукту. Взаємодія цих фреймворків із сторонніми бібліотеками та з модулями додатку значно ускладнює програмний продукт.

Для того щоб в програмному коді було легше знайти бізнес-логіку, розробка повинна відштовхуватись від області застосування додатку.

Використання веб-фреймворків не є обов'язковим для Java. Натомість слід розглядати можливість створення простих додатків методами Java Standard Edition [2].

Проблема розгортки проекту на сервері може бути вирішена вбудованим сервером, який буде запускати проект із Java-коду. Прикладом такого сервера є Jetty. Jetty може бути запущений локально (для розробки або у дрібномасштабному виробництві) або як частина іншого HTTP сервера для повномасштабної роботи. Використання Maven, Gradle, та інших інструментів для збірки проекту не завжди виправдане. Невеликі проекти доцільніше збирати напряму за допомогою Ant, а їх залежності дуже легко регулюються Ivy. Це може зменшити кількість конфігураційних файлів та умов для розгортки додатку до мінімуму.

Таким чином, не використовуючи функціонал JEE, лише за допомогою Java Standard Edition, можна створити повноцінний веб-проект, розробка і подальша підтримка якого буде набагато ефективніша по часу і вартості продукту.

У науковій роботі було проведено порівняльне дослідження статистичних характеристик програми з ідентичним до предмету дослідження функціоналом. Важливо відмітити, що набір функцій програми не змінювався, дві версії виконують один і той самий набір функцій. Відмінністю другої версії є відмова від використання Spring MVC, JSP, ESB та SOAP веб-сервісів.