

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ В ЗАДАЧІ ОЦІНЮВАННЯ СКЛАДНОСТІ ООП ЧЕРЕЗ ЗВ'ЯЗКИ МІЖ КЛАСАМИ ЗАСТОСУНКІВ, ЩО СТВОРЮЮТЬСЯ ЗА ДОПОМОГОЮ SPRING FRAMEWORK

Давидов Д.В.¹; Латанська Л.О., к.ф.-м.н., доц.²

^{1,2} Національний університет кораблебудування імені адмірала Макарова

^{1,2} Миколаїв, Україна

¹ den0802200303@gmail.com; ² llatanskaya@gmail.com

Анотація. У роботі запропоновано побудову еліпсу передбачення, оснований на перетворенні у вигляді десяткового логарифму, для оцінювання складності ООП через зв'язки між класами застосунків, що створюються за допомогою Spring Framework.

Ключові слова: еліпс передбачення; складність ООП; зв'язки між класами; застосунки; Spring Framework.

Вступна частина. На сьогоднішній день питання оцінювання складності ООП на основі зв'язків між класами є актуальним для ІТ-компаній, так як це допомагає значно знизити витрати на поточну розробку і підтримку програмних продуктів протягом їх життєвого циклу. При оцінюванні складності ООП через зв'язки між класами в [1] запропоновано використовувати разом наступні об'єктно-орієнтовані метрики: зв'язність між об'єктами (CBO, Coupling Between Objects) та відгуки класу (RFC, the Response for Class). Вказані метрики відносяться до набору метрик Чидамбера-Кемерера та, зазвичай, аналізують окремо, без урахування кореляції між ними. При цьому допустимі значення визначають лише на рівні окремого класу, що може призвести до помилкових результатів досліджень. З [2] відомо, що значення CBO менше 14 та RFC менше 50 для класу являються допустимими. Згідно з [3], перевищення цих порогів для метрик CBO та RFC може вказувати на проблеми в дизайні класів, що призводить до ускладнень у супроводженні, розширенні, налагодженні та тестуванні системи. Цей підхід забезпечує базу для прийняття рішень щодо можливого рефакторингу коду з метою зменшення зв'язності та спрощення відповідальності класів. Саме тому виникає потреба у створенні математичної моделі, яка дозволить оцінювати складність ООП на основі зв'язків між класами за значеннями метрик CBO та RFC з урахуванням кореляції між ними на рівні всієї системи, зокрема і для програмних застосунків з використанням Spring Framework.

Мета роботи полягає в побудові еліпсу передбачення для оцінювання складності ООП через зв'язки між класами застосунків, що створюються за допомогою Spring Framework.

Основна частина. Для проведення дослідження на ресурсі GitHub (<https://github.com>) знайдено 50 застосунків, що створені за допомогою Spring Framework. Для збору значень з метрик CBO та RFC застосовано ПЗ для знаходження числових характеристик класу SK Metrics [4].

Оцінювати складність ООП на основі зв'язків між класами планується за допомогою еліпса передбачення. Як відомо з [5], в статистиці рівняння еліпса передбачення використовується для виявлення викидів у двовимірних нормально розподілених даних. При перевірці нормальності розподілу даних метрик CBO та RFC буде використано критерій Мардіа, який ґрунтується на обчисленні асиметрії та ексцесу в багатовимірному просторі.

У разі виявлення негаусівського розподілу, як зазначено в [6], для оцінювання складності ООП на основі зв'язків між класами пропонується використати рівняння еліпса передбачення, що залежить від нормалізованих даних метрик СВО та RFC. В якості нормалізуючого перетворення буде використовуватися десятиковий логарифм.

Вихід точки нормалізованих даних за межі еліпса передбачення буде вказувати на наявність викидів з певною ймовірністю, для якої еліпс був побудований. Гарною практикою є використання довірчої ймовірності 0,995, яку зазвичай вибирають при визначенні викидів у багатовимірних даних. Також вихід точки даних з нормалізованими значеннями метрик СВО та RFC на рівні застосунку за межі еліпса буде свідчити про підвищену складність ООП. Якщо ж точка знаходиться всередині еліпса, то складність є невисокою.

Висновки. При оцінюванні складності ООП на основі зв'язків між класами застосунків, розроблених з використанням Spring Framework, пропонується застосовувати еліпс передбачення, який залежить, у випадку гаусівського розподілу, від даних метрик СВО та RFC, а при негаусівському розподілі – від нормалізованих даних метрик СВО та RFC. У подальших етапах планується побудова та програмна реалізація математичної моделі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Chidamber, S., & Kemerer, C. (1994). A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, 20(6), 476-493.
- [2] Benkaddour, R. (2024). *Analyzing CK Metrics Results and Quality Assessment*. <https://medium.com/@benkaddourmed54/analyzing-ck-metrics-results-and-quality-assessment-a70ba56534f0>
- [3] Sahraoui, H.A., Godin, R., & Miceli, T. (2000). Can metrics help to bridge the gap between the improvement of OO design quality and its automation? *Software Maintenance : 2000 International Conference (ICSM)* (pp. 154-162). San Jose, CA, USA. DOI: <https://doi.org/10.1109/ICSM.2000.883034>
- [4] Aniche, M. (2015). *Java code metrics calculator (CK)*. <https://github.com/mauricioaniche/ck>
- [5] Приходько, С.Б., & Смикодуб, Т.Г. (2021). Математична модель для перевірки взаємозв'язків між частинами Java-застосунків за метриками RFC та СВО. *Інформаційні технології: моделі, алгоритми, системи): II-га Всеукраїнська науково-практична інтернет-конференція (ITMAS-2021)* (С. 25-26). Миколаїв, Україна. <http://itconf.nuos.edu.ua/2021/proceedings/>
- [6] Приходько, С.Б., Кольцов, А.В., & Грабінський, М.Р. (2023). Математична модель для визначення складності об'єктно-орієнтованого проектування через зв'язки між класами застосунків з відкритим кодом на Kotlin. *Інформаційні технології: моделі, алгоритми, системи : IV-а Всеукраїнська науково-практична інтернет конференція (ITMAS-2023)* (С. 38-39). Миколаїв, Україна. <http://itconf.nuos.edu.ua/2023/proceedings/>

Davydov D.V., Latanska L.O.

Mathematical modeling in the problem of assessing the complexity of OOP through connections between classes of applications created using the Spring Framework

Abstract. The paper proposes the construction of a prediction ellipse based on the conversion in the form of a decimal logarithm to estimate the complexity of OOP through the relationships between classes of applications created using the Spring Framework.

Keywords: prediction ellipse; complexity of OOD; relationships between classes; applications; Spring Framework.