

Коротке рев'ю наукової публікації «Порівняння об'єктно-орієнтованої та функційної парадигм програмування у проектуванні програмного забезпечення»

Вступ

Наукова публікація за авторством А.І.Коваль, О.М.Яшина, Г.І.Радельчук, Ю.Ф.Форкун (Хмельницький університет) фокусується на порівнянні двох парадигм програмування – об'єктно-орієнтованого та функційного у контексті підходів до проектування програмного забезпечення. Автори публікації коротко характеризують обидві парадигми, окремо виділяють їх недоліки, також зупиняються на відмінностях цих парадигм. В заключенні окремо визначають тенденції в розвитку об'єктно-орієнтованого та функційного програмування. Автори дослідження ґрунтуються у своїх висновках на цілому ряді літературних джерел, однак більшість з них висвітлюють об'єктно-орієнтовну парадигму. Певно, цій парадигмі проектування ПЗ присвячено значно більше досліджень, ніж функційному програмуванню.

Методологія

Дослідження цілком ґрунтується на використанні літературних джерел, що висвітлюють тематики об'єктно-орієнтованого та функційного підходів. Окремо автори зупинилися на питанні тенденцій розвитку цих парадигм, привівши для наочності графіки динаміки використання запитів до пошукової системи Google.

Кількість використаних літературних джерел не є великою. Автори зверталися до Вікіпедії як джерела про загальні відомості щодо парадигм програмування; використали 3 дослідження для узагальненого висвітлення об'єктно-орієнтовної парадигми; звернулися ще до двох джерел з метою зрозуміти концепції функційного підходу.

Така трохи незбалансована кількість використаних джерел наочно демонструє нерівномірність сьогоdnішнього використання парадигм у проектуванні сучасного бізнесового ПЗ – досліджень на тему ООП значно більше. Це до речі наочно ілюструється і приведеними авторами графіками динаміки звернень з пошуковими запитами до пошукової системи Google, що демонструють значно більше запитів про об'єктно-орієнтовні концепції, аніж про функційні. Хоча останні роки, очевидно, що ситуація намагається вирівнятися.

Результати

Автори статті отримали низку цікавих результатів щодо відмінностей та недоліків об'єктно-орієнтованого та функційного підходів у програмуванні. Також, значущим результатом можна визначити наявність у статті істотних тенденцій використання обох парадигм.

Автори публікації вгледіли, що недоліків у ООП менше, аніж у функційного програмування і взагалі останнє є більш складним для використання, що призводить до того, що розробників, котрі використовують ФОП стає дедалі менше. Зазначу зі свого боку, що, фактично «безконтрольний» підхід з боку безлічі спрощених курсів програмування, котрі

намагаються за гроші залучити якомога більше людей з різних сфер життя суспільства у сферу програмування створює ілюзію простоти ООП у використанні розробником, бо саме концепції ООП намагаються спростити і покласти «до рота» новоспеченим студентам, щоб вони хоч якось зрозуміли абстракції цієї парадигми. Ілюзія зростання кількості розробників на ООП створюється саме за рахунок наявності таких спрощених курсів, бо функційне програмування фактично не викладається для таких студентів і не порівнюються абстракції ООП та ФОП, декларуючи, що студент вже просто буде готовий до розробки в рамках ООП. Хоча саме ООП, - повністю згоден з авторами дослідження – є складним концептом, що вони також виділили у якості недоліка ООП як парадигми програмування.

Обидві концепції – що ООП, що ФОП – сформовані з однією метою: швидка розробка з мінімальною кількістю помилок. Це фактично єдине, що є об'єднуючим фактором для обох парадигм. В іншому автори дослідження вбачають лише відмінності. Це стосується як методів обробки та зберігання даних, фокусу концепцій (основа на об'єктах і основа на функціях/процедурах), моделі парадигм (декомпозиції чи узагальнення деталей), питання абстракцій (сутність чи функція реального світу), так і в стилях програмування.

Важливим результатом дослідження зазначу виокремлення тенденцій щодо розвитку використання обох парадигм у проектуванні ПЗ. Автори зазначають, що є невірним твердження витіснення функційною парадигмою об'єктно-орієнтовного підходу в програмуванні. Такий висновок ґрунтується на поясненні, що лише завдяки ООП досягається: «Багаторазове використання коду, абстрагування даних, ефективне вирішення проблем та гнучкість поліморфізму...». Однак фактор системної безпеки визначається вже функційним програмуванням. Функційна парадигма є надважливою при розробці та тестуванні моделей машинного навчання.

Ключові інсайти

Певно, серед ключових інсайтів слід виділити різницю в моделюванні парадигм. Якщо ФОП розробляє модель зверху донизу, то ООП знизу догори. В розробці програмного забезпечення слід розуміти, що розробку краще виконувати від деталі до узагальнюючої концепції, тоді як функційне програмування навпаки визначає хід від узагальнюючої процедури до більш деталізованих функцій.

Серед найважливіших інсайтів розуміємо правильний вибір парадигми для вирішення конкретного завдання. Не існує правильних чи неправильних концепцій чи парадигм. Є лиш вірно обрана парадигма для вирішення поставленого завдання. Більше того, при необхідності можливо використати і гібридний підхід – ФОП разом з ООП.

Висновок

Наукова публікація авторів Хмельницького університету є досить цікавою та пізнавальною. Для свого дослідження вони використали ряд літературних джерел з метою порівняти дві найпопулярніші на сьогодні концепції парадигм програмування, якими є об'єктно-орієнтований та функційний підходи. Вони змогли дослідити ряд недоліків цих парадигм, а також знайшли відмінності між ними, зазначивши складність використання обох парадигм в розробці ПЗ. Також, автори статті намагалися пролити світло на майбутнє використання

концепцій, розмістивши графіки запитів до пошукової системи Google, котрі демонструють значне зближення кількості запитів як по абстракціям ООП та і по абстракціям ФОП.