Осипчук Л. Рев’ю**[​](https://textbook.edu.goit.global/lms-python-homework/uk/docs/programming-core/hw-01/" \l "%D0%BD%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D0%B0-%D0%BF%D1%83%D0%B1%D0%BBi%D0%BA%D0%B0%D1%86i%D1%8F-1-%D1%80%D0%B5%D0%B2%D1%8E" \o "Direct link to heading)** Порівняння об’єктно-орієнтованої та функційної парадигм програмування у проектуванні програмного забезпечення

У науковій публікації «Порівняння Об’єктно-Орієнтованої та Функційної парадигм програмування у проектуванні програмного забезпечення»авторів А.І. Коваль, О.М. Яшина, Г.І. Радельчук, Ю.В. Форкун з *Хмельницького Національного Університету*досліджено відмінності та переваги об’єктно-орієнтованої та функційної парадигм програмування та проаналізовано доцільність використання кожної з них для конкретних цілей**.** Основною метою даної публікацїї був порівняльний аналіз ефективності та зручності використання двох найпопулярніших парадигм програмування, які найчастіше використовуються проектувальниками, розробниками та дизайнерами при розробці програмного забезпечення.

Автори наводить ряд особливостей та переваг кожної з парадигм, таких як:

- доцільно використовувати Об'єктно-Орієнтовані мови програмування, коли розробник має фіксований набір операцій і для розширення коду додаються нові класи, що реалізують існуючі методи;

-використання функційних мов програмування доцільно за наявності у розробника фіксованого набору речей, коли для розширення коду, у першу чергу, додаються нові операції над існуючими речами, додавши нові функції, які обчислюються з існуючими типами даних.

Цікавим є те, що фундаментальною являється різниця концепцій між двома парадигмами. У функційно-орієнтованому дизайні найбільш значущою є функція програми, а в об'єктно-орієнтованій програмі в свою чергу - дані та маніпулювання ними. При цьому функційно-орієнтоване програмування використовує ітеративну процедурну декомпозицію, що є стратегією зверху вниз, а функційно-орієнтована програма розробляється, починаючи від висококонцептуальної моделі, переходячи до деталей нижчого рівня поки не буде досягнуто рівень атомарності окремої функції, тобто використовує підхід знизу догори.

В результаті аналізу автори дійшли висновків, що функціональне та об'єктно-орієнтоване програмування мають на меті створення зрозумілих та ефективних програм без помилок і, незважаючи на їх кардинальні відмінності, роблять свою роботу добре при цьому використовуючи абсолютно різні підходи до створення цих програм. Підхід Об'єктно-Орієнтованих мов програмування - не вказуючи, ЩО потрібно виконати визначає, ЯК це зробити, в той час як функціональне програмування визначає, ЩО потрібно виконати, не вказучи, ЯК.

Отже, вибір заключається в тому, яка з парадигм найкраще вирішує поставлену задачу, але у великих проектах, у яких бере участь багато людей, досить часто складно уникнути труднощів, тому може знадобитися гібридний підхід, який об’єднує переваги обох парадигм.