Гаманюк І.А., здобувач,
Петросян Р.В., ст. викладач
Державний університет «Житомирська політехніка»

ПРИНЦИПИ ТА ОСОБЛИВОСТІ DATA-ORIENTED DESIGN

Data Oriented Design (DOD) – парадигма програмування, яка ставить в центр уваги організацію даних, що дозволяє підвищити ефективність роботи з пам'яттю та продуктивність програмного забезпечення. У контексті DOD переосмислюється об'єктно-орієнтоване програмування (ООП) та використовуються основні принципи SOLID, щоб підвищити адаптивність і модульність коду. Однак традиційний ООП із суворим дотриманням цих принципів часто може створювати надмірно складні та громіздкі системи. ООП зі своїми шаблонами проектування є корисним, але його класична реалізація не завжди ефективна для сучасних вимог високопродуктивного коду, де потрібна оптимізація пам'яті та обчислень.

Функціональний підхід до проектування даних також відіграє ключову роль у формуванні структури сучасних програм. Зокрема, функціональне програмування привнесло корисні концепції, такі як імутабельність даних та оператори вищого порядку, як-от ".map", ".filter", ".reduce". Імутабельність забезпечує спрощене управління даними та безпечність, особливо в умовах багатопотоковості. Але повністю функціональний підхід є неефективним для обчислень, оскільки комп'ютери розроблені для роботи з мутаціями даних, і кожна зміна імутабельних структур потребує їх копіювання, що призводить до суттєвих витрат ресурсів. Крім того, імутабельність може негативно впливати на продуктивність через потребу у постійному копіюванні об'єктів для забезпечення незмінності.

Водночас імутабельні структури даних мають переваги при розміщенні в стеку або кеші процесора, де вони можуть бути доступними швидше. Тому доцільно знаходити баланс між імутабельністю та мутабельністю, беручи найкраще з функціонального підходу та доповнюючи його ефективними можливостями мутації ООП. Комбінація ООП та функціональних елементів дозволяє зберігати простоту коду без шкоди для продуктивності, оскільки ми можемо використовувати мутабельність там, де це є ефективним, та імутабельність для надійності.

Отже, підхід Data-Oriented Design вимагає переоцінки традиційних принципів чистого ООП та функціонального програмування. Сучасне програмування потребує підходів, що можуть ефективно поєднувати високу продуктивність і оптимізацію пам'яті, зберігаючи при цьому зручність розробки. Для цього необхідно використовувати адаптивні методи, які дозволяють забезпечити ефективність обробки даних без втрат у гнучкості та підтримуваності коду. Підхід DOD дозволяє знаходити баланс між ООП, функціональним програмуванням і оптимізацією пам'яті, що дозволяє створювати системи з високою продуктивністю, не жертвуючи при цьому якістю коду. Завдяки цьому, Data Oriented Design стає потужним інструментом для розробки сучасного програмного забезпечення, що відповідає вимогам швидкості та ефективності при збереженні чистоти архітектури.

Список використаних джерел

1. Data-Oriented Design URL: https://www.dataorienteddesign.com/dodbook/ (date of access: 10.11.2024).
2. A Comprehensive Guide To Data-Oriented Design For Improved Software Efficiency URL: https://arpanext.medium.com/a-comprehensive-guide-to-data-oriented-design-for-improved-software-efficiency-6434d520d0e4 (date of access: 10.11.2024).
3. Data Oriented Programming in Java URL: https://www.infoq.com/articles/data-oriented-programming-java/. (date of access: 10.11.2024).
4. Data-Oriented Design Now And In The Future URL: https://gamesfromwithin.com/category/data-oriented-design (date of access: 10.11.2024).
5. J. D. Bayliss, "The Data-Oriented Design Process for Game Development," in Computer, vol. 55, no. 5, pp. 31-38, May 2022, doi: 10.1109/MC.2022.3155108.