

МОДЕЛІ КЕРУВАННЯ ДИНАМІЧНОЮ ПАМ'ЯТТЮ В ОБ'ЄКТНО-ОРІЄНТОВАНИХ МОВАХ ПРОГРАМУВАННЯ

*канд. техн. наук, доц. І.О. Музика, Криворізький національний
університет, м. Кривий Ріг*

Стрімкий розвиток технологій розробки програмного забезпечення сприяв становленню об'єктно-орієнтованої парадигми як найбільш зручного, ефективного та надійного способу створення різноманітних інформаційних систем. Проте сьогодні залишається актуальним питання обґрунтованого вибору засобів та інструментів для реалізації систем реального часу, високонавантажених вебсервісів, машинного навчання тощо. Одним із вагомих факторів, які впливають зокрема на швидкість роботи програми та її ефективність у середовищах IoT, є модель керування динамічною пам'яттю [1].

Проведений аналіз сучасних об'єктно-орієнтованих мов програмування дозволяє виділити такі основні підходи як ручне керування пам'яттю (C++), автоматичний збиральник Garbage Collector (Java, Kotlin, Python, C#, JavaScript) та підрахунок посилань на об'єкти Reference Counting (Swift, PHP, Delphi). Остання модель може бути кращою для програм, де продуктивність є критично важливою, а динамічне виділення великих блоків пам'яті використовується відносно нечасто. Підрахунок посилань більш економічний, ніж Garbage Collection, оскільки не використовує додаткових ресурсів для відстеження та видалення об'єктів. Однак слід зауважити, що така модель інколи призводить до надмірної фрагментації оперативної пам'яті [2], тому вимагає від програміста додаткових зусиль під час вибору структур даних та правильної архітектури застосунку.

За допомогою комп'ютерного моделювання було встановлено, що програма для розпізнавання зображень із роздільною здатністю до 10 МР, розроблена мовою C++, витрачає майже у 3,5 менше часу на обробку та споживає у 1,9 рази менше оперативної пам'яті, ніж подібна утиліта, створена на базі технології .NET.

Таким чином, зручність розробки програмного забезпечення з використанням C#, Java часто може вимагати додаткової оптимізації алгоритмів, зокрема під час створення рішень, що функціонують в умовах високих навантажень.

Список літератури: 1. Jones R., Hosking A., Moss E. The garbage collection handbook: the art of automatic memory management. 2nd ed. Chapman & Hall, 2023. 573 p. 2. Zhang J., Yeung S. H., Shu Y., He B., Wang W. Efficient memory management for GPU-based deep learning systems. arXiv preprint arXiv:1903.06631, 2019.