

РОЗРОБКА ТЕХНОЛОГІЇ КОМПІЛЬОВАНОЇ РЕФЛЕКСІЇ В .NET

Вінницький національний технічний університет

Анотація

Запропоновано технологію компільованої рефлексії для платформи .NET, що дозволить реалізувати основний функціонал механізму рефлексії (створення об'єкту класу, виклик методів, зчитування та присвоєння властивостей) за допомогою Expression.

Ключові слова: .NET, метапрограмування, рефлексія, Expression.

Abstract

A technology of compiled reflection for the .NET platform that utilizes Expressions is proposed, which will allow implementing the main functionality of the reflection mechanism (creating a class object, calling methods, reading and assigning properties).

Keywords: .NET, metaprogramming, reflection, Expression.

Вступ

Механізм рефлексії (System.Reflection) є одним із найчастіше використовуваних інструментів метапрограмування в .NET [1]. Завдяки своїй простоті та функціональності він дозволяє легко розробляти програмні модулі, що виконують загальний функціонал або мають працювати з великою кількістю різних типів даних. Одним із типових прикладів є серіалізація та десеріалізація даних, де завдяки використанню рефлексії можна узагальнити логіку для всіх типів даних. Таким чином, кількість необхідного програмного коду для реалізації функціоналу зменшується в рази.

Головним недоліком рефлексії в .NET є швидкодія [2]. Якщо рідко використовувати рефлексію, цей недолік є непомітним, але в критичних місцях бажано уникати частого використання такого механізму. Через це доцільною є розробка підходу, що мав би схожий до рефлексії синтаксис та функціонал, але мав би значно вищу швидкодію.

Розробка моделі компільованої рефлексії

Основний функціонал рефлексії, який використовується для розробки загального функціоналу, полягає в можливості отримувати інформацію про тип даних, створювати екземпляри класу, зчитувати та присвоювати значення полям та властивостям класу. Також важливим є функціонал динамічного виклику методів, оскільки властивості в класах .NET реалізуються через `get` і `set`, що є по суті методом для зчитування (`get`) властивості та її присвоєння (`set`).

Найважливішим функціоналом, який необхідно реалізувати в обхід рефлексії, є саме дії з об'єктами класів (створення, виклик методів, зчитування та присвоєння значень полям). Саме ці операції мають найбільший вплив на швидкодію [2].

Для реалізації цього функціоналу пропонується використовувати механізм виразів (Expression) платформи .NET [3]. Вирази дозволяють отримати лямбда функції, що реалізують необхідний функціонал механізму рефлексії. Виклик скомпільованої лямбда функції є швидшим за використання рефлексії, що дозволить досягти кращої швидкодії [4].

З точки зору архітектури у такому випадку доцільно мати статичний клас, що містить у собі словники (Dictionary), в яких зберігається відповідність необхідної операції до лямбда функції. Це дозволить тримати в пам'яті вже скомпільовану функцію, яку можна швидко знайти за ключем та викликати. В якості ключа пропонується використовувати таку ієрархію: тип об'єкту, тип операції, ім'я методу або властивості.

Отже, при такому підході, необхідно зберігати в пам'яті лямбда функцію для кожного типу даних, операції та методу або властивості. Кожна з таких функцій має бути скомпільована в той чи інший час роботи додатку. Пропонується створювати таку функцію при спробі першого виклику. Наприклад,

необхідно зчитати значення властивості. У такому випадку механізм повинен спробувати знайти відповідну лямбда функцію у словнику. Якщо функцію не буде знайдено, то механізм повинен її створити, помістити до словника та виконати. При такому підході перший виклик завжди буде повільнішим, що є недоліком. Проте це компенсується тим, що будуть створені лише функції для необхідних класів та властивостей, а також це дозволить використовувати підхід з анонімними типами даних. Блок-схему загального алгоритму технології компільованої рефлексії зображено на рис. 1.

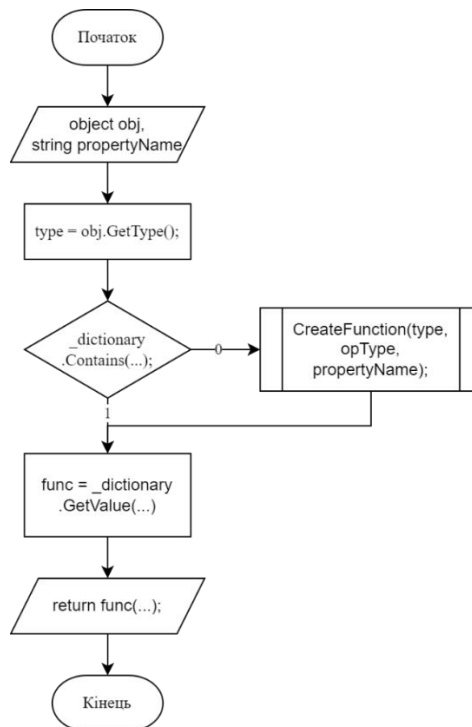


Рисунок 1 – Блок-схема загального алгоритму технології компільованої рефлексії

Висновок

Запропоновано технологію компільованої рефлексії для платформи .NET, що дозволить реалізувати основний функціонал механізму рефлексії (створення об'єкту класу, виклик методів, зчитування та присвоєння властивостей). Запропонована технологія використовує механізм виразів (Expression) для побудови та компіляції функції в процесі роботи додатку. Саме за рахунок виклику вже скомпільованих функцій можливо досягти кращої швидкодії у порівнянні зі звичайною рефлексією.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ingebrigtsen E. Metaprogramming in C#: Automate your .NET development and simplify overcomplicated code / Einar Ingebrigtsen. – Birmingham, 2023. – 352 с. – (Packt Publishing).
2. Warren M. Why is reflection slow? [Електронний ресурс] / Matt Warren. – 2016. – Режим доступу до ресурсу: <https://mattwarren.org/2016/12/14/Why-is-Reflection-slow/>.
3. Expression trees [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/dotnet/csharp/advanced-topics/expression-trees/>.
4. Hazzard K. Metaprogramming in .NET / K. Hazzard, J. Brock. – New York, 2013. – 360 с. – (Manning).

Войтко Вікторія Володимирівна – кандидат технічних наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: dekanfki@i.ua.

Позур Михайло Юрійович – аспірант кафедри програмного забезпечення, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: mixalchik545@gmail.com.

Viktoriia Voitko – Ph.D., Associate Professor of Software Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: dekanfki@i.ua.

Pozur Mykhailo – Ph.D. student, Faculty for Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: mixalchik545@gmail.com.