УДК 004.43

М.Р. ПОСТОЄНКО, студент, О.Г. РИБАЛЬЧЕНКО ст. викладач, С.В. БІЛАШЕНКО, асистент,
Криворізький національний університет

ПЕРЕВАГИ ЗАСТОСУВАННЯ КОМПОНЕНТА LINQ ПЛАТФОРМИ .NET

В мові програмування C# при роботі зі структурованими даними з використанням імперативного підходу, тобто чіткої послідовності виконання команд, виникає ряд незручностей та проблем. Першою з них є складність розуміння написаного коду через розподіл логіки по всій програмі, що призводить до неочевидності рішень. Другою є проблема підтримки розробленої програми через її непрозорість та об'ємність. Третя полягає у можливості перенесення програми на іншу платформу, оскільки реалізація залежить від середовища програмування. Четверта проблема - це управління станом програми, яке потребує використання великих конструкцій та багатьох змінних, що зменшує оптимізацію програми. Усі ці фактори у сукупності стають «каменем» у підтримці програмного продукту.

Натомість використання декларативного підходу, сутність якого полягає в описі бажаного результату виконання програми та набору правил для його досягнення, дає можливість подолати основні проблеми імперативного підходу. Компонент LINQ, як інтегрована мова запитів до структурованих даних, є прикладом саме такого підходу. Він був створений у 2007 році, та був вперше інтегрований в .NET Framework версії 3.5.

Головна ідея вбудованої мови запитів - це посередництво між основною програмою та джерелом даних. Завдяки своєму інтегрованому характеру LINQ може використовувати у своїй реалізації об'єктно-орієнтований підхід. Серед важливих особливостей компонента назвемо наступні. Лямбда-вирази дозволяють просте визначення предикатів для методів. Анонімні типи дають можливість створювати об'єкти довільної структури і не оголошувати тип для результату LINQ-виразу. Дерева виразів можуть зберігати предикати у вигляді об'єктів. Методи розширення дозволяють розширяти вже існуючі типи без їх модифікації і наслідування, це розповсюджує застосування LINQ до великої кількості сторонніх типів. Ініціалізатори об’єктів та колекцій дозволяють створювати об'єкти та ініціалізувати їх поля без використання конструкторів.

Використанням LINQ ми не лише вирішуємо проблеми написання функціоналу на чистій мові, а й набуваємо нових можливостей та особливостей декларативного підходу. Це відчувається з самого початку роботи з LINQ. Першою перевагою, яку ми отримуємо, є робота з різними джерелами даних, такими як JSON, XML, бази даних, колекції тощо. Одні й ті самі базові висловлювання запиту дозволяють однаково легко отримувати і перетворювати дані з баз даних SQL, наборів даних ADO.NET, XML-документів, XML-потоків і колекцій .NET. Компонент має багатий власний інструментарій, який не вимагає додаткових реалізацій функціоналу програмістом, встановлення додаткових бібліотек або компонентів для роботи з даними. Все це сприяє прискоренню написання проекту та його впровадженню в експлуатацію. Другою, дуже важливою перевагою, є типобезпечність, яка гарантує перевірку лінгвістичних запитів під час компіляції. Це допомагає уникнути поширених помилок завдяки статичній типізації .NET та C#. Також ця перевага допомагає розробникам уникнути помилок завдяки інтелектуальним підказкам у середовищі розробки, якщо вони використовують Visual Studio. Наступною, але не менш вагомою перевагою, є універсальність, яка надає широкий спектр операцій та функцій для маніпуляції даними, таких як фільтрація, сортування, об'єднання та обчислення. Це дає змогу легко, швидко та ефективно обробляти дані. Четвертою перевагою є підтримка паралельності, суть якої полягає в розподіленні обробки запитів між доступними ядрами процесора та використання потоків для розділення завдань. Це надає можливість підвищити швидкодію та оптимізацію запитів. П'ятою перевагою є легкість читання коду завдяки декларативному підходу, який використовує LINQ. Крім того, це зменшує довжину програми та спрощує її розуміння. Шостою перевагою є адаптивність компонента, що дозволяє розробникам розширювати його функціонал до конкретних потреб або специфіки бази даних, використовуючи власні рішення або сторонні бібліотеки та компоненти.

Застосування LINQ у платформі .NET надає значні переваги розробникам, завдяки його декларативному підходу до роботи з даними та ефективному зв’язку з різними джерелами даних.