УДК 004.43

Є.О. КАСПРУК студент, О.Г. РИБАЛЬЧЕНКО ст. викл., С.В. БІЛАШЕНКО, асистент,
Криворізький національний університет

ЛЯМБДА-ЧИСЛЕННЯ ТА ВИКОРИСТАННЯ АНОНІМНИХ ФУНКЦІЙ В ПРОГРАМУВАННІ

Концепцію анонімних функцій, де всі функції не мають імені, вперше представив Алонзо Черч у своїй роботі про лямбда числення. Після нього ідею лямбда числення продовжили досліджувати і розширювати інші вчені.

Однією з головних фігур в розвитку теорії типів і лямбда-числення був Хаскель Каррі. Він розширив ідеї Черча, впровадивши поняття "каррівської абстракції", яка дозволила створювати анонімні функції шляхом узагальнення існуючих функцій.

Петер Ландін також був активним дослідником у галузі теорії типів і лямбда-числення. Його роботи спрямовані на розробку та формалізацію математичних концепцій, що стосуються лямбда-числення, а також вивчення його застосувань у різних галузях, зокрема, у функціональному програмуванні та математичній логіці.

Лямбда-числення може бути розглянуте як ідеальна та мінімалістична мова програмування, що подібна до машини Тюринга - іншої мінімалістичної абстракції, здатної визначати будь-який алгоритм. Однак, вони різняться за парадигмою програмування: лямбда-числення відповідає функціональній парадигмі, тоді як машина Тюринга — імперативній. У машини Тюринга є певний "стан", що описується переліком символів, що можуть змінюватись з кожною наступною інструкцією. Навпаки, лямбда-числення уникає концепції станів: воно працює з функціями, які отримують значення параметрів та повертають результати обчислень, але не змінюють вхідні дані напряму.

Анонімні функції є важливою концепцією в багатьох мовах програмування. У деяких мовах програмування для оголошення анонімних функцій використовується ключове слово lambda, і такі функції часто отримують назву лямбда абстракцій або лямбда виразів. Вони дозволяють створювати функції без необхідності надавати їм імені, що забезпечує більшу гнучкість і компактність коду. Анонімні функції підтримуються в багатьох мовах програмування, включаючи C++, Python, JavaScript, Ruby, C#, Java і багато інших. У кожній мові вони можуть мати свої особливості та синтаксис.

Анонімні функції знаходять своє застосування в різних ситуаціях. Один із випадків їх використання - передавати функції як аргументи в інші функції. Це особливо корисно, коли вам потрібно виконати певну дію всередині функції, а сама функція буде викликана зовнішньо. Іншим поширеним застосуванням є випадок, коли функція використовується лише один раз або обмежену кількість разів у програмі. В цій ситуації використання анонімних функцій може бути простішим з точки зору синтаксису, ніж створення окремої іменованої функції. Анонімні функції, як правило, широко застосовуються у функціональних мовах програмування, де вони служать для функціональних типів тим же самим цілям, що і літерали для інших типів даних.

Анонімні функції є формою вкладених функцій в тому плані, що вони дозволяють доступ до змінних в зоні видимості функції, в яку вони вкладені. Це означає, що анонімні функції повинні визначатися з використанням замикань. Такий підхід дозволяє створювати більш складні логічні структури та уникати змінних, що глобальні для всієї програми.

На відміну від іменованих вкладених функцій, лямбда-функції не можуть бути рекурсивними без використання оператору фіксованої точки. Анонімні функції можуть допомогти у втіленні принципу інкапсуляції, оскільки вони дозволяють обмежувати доступ до функцій лише в межах конкретного контексту. У функціональному програмуванні вони відіграють важливу роль, оскільки дозволяють виконувати операції на списку значень без необхідності визначення окремих функцій.

Отже, лямбда-функції застосовуються скрізь, де потрібні об'єкти-функції. Вони можуть мати довільну кількість аргументів, але обчислюють та повертають лише одне значення. Синтаксично лямбда-функції обмежені та дозволяють представити лише один вираз. Анонімні функції мають безліч варіантів застосування в конкретних сферах програмування, поряд з іншими типами виразів, що використовуються у функціях.