

УДК 004.384

Луценко Н.В.¹, Паромова Т.О.¹, Івашенко Н.А.², Бредун А.Ю.³

¹ старш. викл. НУ "Запорізька політехніка"

² студ. гр. КНТ-518 НУ "Запорізька політехніка"

³ студ. гр. КНТ-528 НУ "Запорізька політехніка"

ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА МОВ ПРОГРАМУВАННЯ C++ ТА CLOJURE

Сьогодні незважаючи на появу нових мов програмування, все ще активно використовується така мова як C++. Ця мова прийшла з далекого 1983 року на зміну Common Lisp, Fortran, Algol 68 та інших. Мова C++ – це розширення мови C. При цьому в C++ не намагаються вирішити проблеми шляхом видалення елементів C, а додається можливість роботи з класами та об'єктами. Оригінальна мова C++ була розроблена Б. Страуструпом, для якого базових можливостей звичайного C виявилось замало, коли він почав працювати над задачами теорії черг.

Мова C++ відноситься до об'єктно-орієнтованих мов зі строгою типізацією, можливістю процедурного програмування та імперативною парадигмою. Важливо зауважити, що вона зберігає сумісність з C, наскільки це можливо. Мова C++ з самого початку орієнтована на компіляцію в машинний код заданої платформи (хоча, теоретично, ніщо не заважає створювати для C++ транслятори в проміжний код). Це вже визначає різницю в сферах застосування мов. Мова C++ зберігає можливість роботи з низькорівневими показниками та ручним керуванням пам'яттю. З одного боку це можна віднести до переваг, проте на великих проектах це скоріш зайве та заважає сконцентруватися на реалізації потрібного завдання. А от що дійсно завжди супроводжувало C++, так це швидкість компілятора.

У якості протиприкладу, можна розглядати мову Clojure, яка базується на хост-машині Java. У 2007 році Річ Хіккі розробив мову Clojure, як сучасний діалект Lisp, який орієнтується на функціональну складову, та особливо "чисті" функції. Мова Clojure - мова загального призначення, застосовувати його можна для рішення будь-яких задач. Мета, яка ставилася при проектуванні Clojure – зробити написання програми простими. І, як наслідок, прискорити їх створення, тестування. А головне, зменшити час на їх розуміння, редагування і супровід. На мові Clojure можна писати і повноцінні

веб-сайти та виконати досить складні обчислення. При цьому Clojure підтримує інтерактивну розробку за допомогою REPL. Також Clojure полегшує роботу з конкурентним та паралельним програмуванням.

Головною особливістю мови Clojure є імутабельність (незмінність) структур даних. Немає способу змінити елемент вектору. Можливо тільки створити новий вектор, у якого буде змінений один елемент. Дуже важливий момент в тому, що Clojure зберігає алгоритмічну складність (за часом і за пам'яттю) для всіх стандартних операцій над колекціями. Мова Clojure володіє гомоіконністю, тобто написаний код зчитується в структури даних, які виглядають так само як і початковий код. Звідси випливає те, що Clojure має в своєму арсеналі такий потужний інструмент, як макроси. Вони дозволяють виконувати написаний код під час компіляції та підставляти його замість самого макросу вже під час виконання. Якщо використовувати макроси, коли це дійсно потрібно, то можна зменшити обсяг написаного коду. Також варто зауважити, що Clojure підтримує глобально інтерактивну розробку за допомогою середовища програмування REPL (read-eval-print-loop). Ця можливість дозволяє експериментувати з програмою на ходу, тобто без необхідності компілювати її. Clojure нечиста мова, і функції можуть мати побічні ефекти. Наприклад, `println` - це виклик функції, дія. Важливим є те, що сама суть подібних функцій полягає у взаємодії із зовнішнім світом. Вивести значення в файл, відправити HTTP запит, виконати SQL - всі ці дії не мають сенсу у відриві від створюваного ними побічного ефекту. Тому дуже корисно такі функції (чисті і брудні) розділяти.

Особливості мови Clojure: реалізація багатопоточності заснована на незмінності значень і змінних посиланнях на них, які дозволяють управляти паралельним доступом з різних потоків; функції є об'єктами першого класу; підтримка множин, векторів і словників як інтегрованих структур даних; акцент робиться не на цикли, а на рекурсію і функції вищого порядку.

В Clojure добре зроблений інтерфейс взаємодії з колекціями даних і взагалі зі структурами. Всі колекції обробляються однаково за рахунок абстракції «послідовності». Для доступу до елементів послідовностей є багато функцій, які дозволяють отримати бажаний результат незалежно від того, чи працюють вони зі списком, вектором або безліччю.

Однією важливою відмінністю між мовами C++ та Clojure є те, що остання має в своєму розпорядженні потужний інструмент, який автоматично збирає "сміття". При інтенсивній роботі з незмінними колекціями іноді виникає занадто багато проміжних об'єктів, що досить неефективно. В Clojure є можливість тимчасово використовувати змінні колекції даних використовуючи перехідні (transient) структури даних. Ця функціональність була спеціально введена в мову для оптимізації продуктивності. Це стало можливим завдяки Java, на якій базується Clojure. Важливим є те, що Clojure

повністю сумісний з кодом Java, адже у підсумку компілюється у байт-код. Це надає можливість запускати написані на Clojure програми усюди, де підтримується JVM. Однією перевагою виступає також незмінність даних та “чисті” функції, на яких базуються функціонально-орієнтовані мови програмування. Незмінність в Clojure гарантує пряму поведінку даних, адже по суті, всі дані є константами, адже створюються лише один раз і не піддаються зміні. При спробі змінити певну структуру даних, створюється її копія завдяки персистентності. Ця особливість гарантує те, що при створенні, наприклад, нової колекції, вона не копіюється повністю, а копіюється тільки частина, яка була змінена.

Цікавим фактом є те, що замість імперативних циклів у мові Clojure явно використовується декларативне програмування. Це означає вирішення багатьох проблем рекурсивними методами. Особливо гарним стилем є так звана “хвостова” рекурсія, яка в свою чергу не забиває стек виконання.

Мова Clojure – це інструмент робочий, надійний, досить універсальний, простий. Звичайно, для його освоєння доведеться витратити деякий час. Багато що може здатися незвичним та дивним. Але вся краса цієї мови в її органічності, тонкому стикуванні окремих елементів в єдине ціле. Ознайомитися з Clojure однозначно варто. Навіть якщо цей інструмент не пригодиться з тій або іншої причини, то ідеї, які в нього закладені, виявляться дуже корисними.

Таблиця 1 – Порівняння основних концептів та парадигм C++ та Clojure

	C++	Clojure
Семантика	Об’єктно-орієнтована	Функціональна
Типізація	Статична, слабка	Динамічна, сильна
Парадигма	Імперативна	Декларативна
Підтримка гомоіконності	Ні	Так
Очистка сміття	Відсутня	Збирач сміття
Синтаксис	Префіксний. Інфіксний. Постфіксний	Префіксний

Всі мови програмування і технології по-своєму цікаві. І, в принципі, все їх можна задіяти в тій чи іншій сфері. Не можна однозначно сказати, яка мова програмування краще: C++, Java, PHP, Python, Clojure та інші. Вибір технології багато в чому залежить від планів з розвитку продукту та бюджету.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Campbell Parallel Programming with Microsoft® Visual C++® / Campbell. – Москва: Гостехиздат, 2011. – 784 с.
2. Альфред В. Ахо Компиляторы. Принципы, технологии и инструментарий/Альфред В. Ахо и др. – Москва:Высшая школа, 2015.– 882 с.
3. Гонсалвес, Энтони Изучаем Java EE 7 / Энтони Гонсалвес. – М.: Питер, 2016. –640 с.
4. Шилдт, Герберт, Java 8. Руководство для начинающих / Герберт Шилдт. – М.: Вильямс, 2015. – 720 с
5. Daniel Higginbotham, Clojure for the Brave and True: Learn the Ultimate Language and Become a Better Programmer – USA, 2015. – 328 с.