

сортирований, та відносно великих розмірів, доцільно буде застосувати метод екстраполяції. Даний метод характеризується тим, що на відміну від бінарного та йому подібних алгоритмів, він не лише визначає зону нового пошуку, а і оцінює величину кроку. І при великих об'ємах має значно більшу швидкість сходження. Звісно слід розуміти, що в ході реалізації проекту дане рішення може змінюватися, в залежності від потреб додатку.

ЛІТЕРАТУРА

1. What is SaaS? The modern way to run software [Електронний ресурс]// www.infoworld.com – Режим доступу до ресурсу: <https://www.infoworld.com/article/3226386/saas/what-is-saas-the-modern-way-to-run-software.html>
2. Searching Algorithms [Електронний ресурс]// www.geeksforgeeks.org – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/searching-algorithms>
3. Infrastructure as a Service (IaaS) [Електронний ресурс]// www.service-architecture.com – Режим доступу до ресурсу: [service-architecture.com/articles/cloud-computing/infrastructure_as_a_service_iaas.html](http://www.service-architecture.com/articles/cloud-computing/infrastructure_as_a_service_iaas.html)

*Кирилюк Є.І.,
студент, Криворізький національний університет
Швець Д.В.,
асистент, Криворізький національний університет
Карабут Н.О.
ст. викладач, Криворізький національний університет*

ЗАСТОСУВАННЯ ФУНКЦІОНАЛЬНОЇ МОВИ ПРОГРАМУВАННЯ RACKET

Розглянуто варіанти застосування функціональної мови програмування Racket. Зазначено можливість її використання в розробці нових мов програмування, навчальному процесі, наукових дослідженнях. Зосереджено увагу на особливостях мови Racket, її перевагах та недоліках.

Racket [1] є мультипарадигмальною мовою програмування, що належить до сімейства Lisp / Scheme. Одне з призначень Racket

– це створення, розробка і реалізація мов програмування. Racket використовується в різних контекстах: як скриптова мова, як мова загального призначення, в навчальних цілях та в наукових дослідженнях. Платформа надає користувачеві реалізацію мови Racket, включаючи розвинену середу виконання, низку бібліотек, JIT-компілятор, а також середовище розробки DrRacket, написане на Racket [2].

Racket призначений для швидкої розробки нових мов програмування. Програмування є формою вирішення проблем. Правильний підхід використовує мову предметної області, щоб сформулювати проблему і сформулювати процеси рішення. Для підтримки цього режиму програмування Racket допомагає програмістам створювати і швидко розгортати нові мови. Зокрема, механізми створення та розгортання мов повинні міститися в самій мові. Цей принцип різко контрастує з численними зовнішніми інструментами та препроцесорами командного рядка, які використовуються для створення предметно-орієнтованих мов.

Racket забезпечує будівельні блоки для надійних механізмів захисту. Якщо програмування стосується вирішення проблем на коректній мові, системи обов'язково будуть складатися з взаємопов'язаних компонентів на декількох різних мовах. Завдяки зв'язкам, значення переходять з одного мовного контексту в інший. Оскільки мови несуть відповідальність за надання та збереження інваріантів, розробники мов повинні захищати інваріанти мов. З цієї причини Racket поставляється з належними будівельними блоками для створення або побудови механізмів захисту на будь-якому рівні.

Racket перетворює екстралінгвістичні механізми в лінгвістичні конструкції. Коли програмістам потрібно вдаватися до позалінгвістичних механізмів для вирішення проблеми, це означає, що обрана мова зазнала невдачі. Навіть якщо не завжди очевидно, як виправити такі помилки, дослідники мови програмування повинні прийняти загальну ідею і спробувати знайти відповідні мовні механізми. Racket є мовою програмування, яка інтегрує кілька механізмів управління ресурсами, які часто зустрічаються в базовій операційній системі. Аналогічно, ця філософія забороняє ідею «проектів», як в інших середовищах розробки, оскільки це також екстерналізує управління ресурсами, зв'язування та інші аспекти створення програм.

Принципи проектування Racket створили мову програмування, яка:

1. дозволяє швидко створювати нові мови для конкретних проблемних областей і, отже, дає змогу забезпечити мовно-орієнтоване програмування;
2. підтримує повний спектр мов програмування загального призначення з різними стандартними ступенями безпеки;
3. інтерналізує механізми з системного контексту в лінгвістичні конструкції для детального, програмованого керування [3].

Ключовою перевагою Racket є його система розширення синтаксису. Це робить досвідчених програмістів надзвичайно продуктивними, але при цьому виникає надзвичайно крута крива навчання. В той же час алгоритм розробки синтаксису є важким для засвоєння, його інструментарій досить великий і складний. Ситуація потребує спрощення синтаксичної системи та плавного розвитку інструментарію.

Окрім мови, сучасним програмістам потрібна екосистема. Дійсно, багато програмістів ототожнюють мови з їх екосистемами. Однак, DrRacket вже може автоматично підтримувати нові мови, наприклад, перевірку синтаксису в режимі онлайн і прості дії з рефакторингу. Для інших інструментів, таких як синтаксично-орієнтований степінг, цей процес потребує значного обсягу робіт і відбувається без будь-якої автоматизації.

ВИСНОВКИ

Функціональна мова програмування Racket, як діалект Lisp, має багато практичних переваг. Одним з головних напрямків її застосування можна визначити розробку нових мов програмування. Також вона може стати в нагоді у наукових дослідженнях та навчанні студентів програмуванню.

ЛІТЕРАТУРА

1. Офіційний сайт Racket [Електронний ресурс] – Режим доступу до ресурсу: <http://racket-lang.org>
2. Racket Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.racket-lang.org>
3. WHY RACKET? WHY LISP? [Електронний ресурс] – Режим доступу до ресурсу: <https://practicaltypography.com/why-racket-why-lisp.html>