Llenguatges de Programació

Treball Dirigit

Crystal

Hash: 7431

Departament de Computer Science

19 de diciembre de 2022

Índice

1. Introducció

1.1. Descripció i paradigmes

Crystal és un llenguatje de programació *open source* de propòsit general, concurrent, recollida d'escombraries i orientat a objectes, disenyat i dessarrollat principalment per Ary Borenszweig, Juan Wajnerman i Brian Cardiff. Té una sintaxi inspirada en el llenguatge Ruby, és un llenguatge compilat amb verificació estàtica de tipus, encara que especificar tipos de variables o els arguments dels mètodes és generalment innecessari ja que Crystal consta amb un algoritme avançat per fer inferència de tipus globals. [7]

Crystal és un llenguatge amb els següents **paradigmes:**

- **Programació funcional:** Permet als desenvolupadors escriure funcions que es comporten com valors i utilitzar tècnicas com la composició i la aplicació parcial per construir programes de manera declarativa i consisa.
- **Orientat a objectes:** Model de programació basat en objectes que contenen dades i procediments associats coneguts com a mètodes. Això permet als programadors crear classes i objectes amb atributs i mètodes, i utilitzar herencia i polimorfisme per reutilitzar i extendre el codi. [8].
- **Concurrent:** Permet executar diferents tasques simultàniament, les quals poden ser un conjunt de processos d'un mateix programa. Amb això es pot aconseguir millorar el rendiment i la escalabilitat de les aplicacions. [6].

Cada un d'aquests paradigmes té les seves propies ventatges i deventatges, per lo que és important elegir el que millor s'adapti a les necessitats del projecte en qüestió.

1.2. Història i objectius

El llenguatge Crystal originalment es deia Joy i es va crear al 2011, amb els objectius originals de combinar la productivitat de Ruby amb la velocitat i seguretat de tipus de un llenguatge compilat com és C. La primera versió oficial va sortir al 2014. [5]

Els principals **objectius** son [4]:

- Tenir una sintaxi similar a Ruby (la compatibilitat amb aquest no és un objectiu).
- Comprobar tipus de manera estàtica pero sense que tenir que especificar el tipus de variables o arguments dels mètodes.
- Ser capaç de utilitzar codi de C.
- Tenir una evaluació i generació de codi en temps de compilació, per evadir codi repetit.
- Compilar a un codi natiu de manera eficient com fa C.

2. Principals aplicacions

Crystal té moltes aplicacions possibles ja que com hem comentat anteriorment és un llenguatge de programació de propòsit general que es pot utilitzar per desenvolupar un ventall ampli de programes. Algunes de les aplicacions principals són:

- **Aplicacions Web:** Crystal té una comunitat de desenvolupadors que han creat una sèrie de Frameworks els cuals parlarem més endevant que fan molt fàcil el desenvolupament d'aplicacions web.
- **Aplicacions de consola:** Gràcies a la sintàxi concisa i clara que té, ens ho posa molt fàcil per poder escriure programas de linies de comandos i aplicacions de consola.
- **Jocs:** També es pot utilitzar per crear jocs gràfics i de text ja que ofereix un rendiment ràpid i un conjunt de llibreries per manegar l'entrada i sortida i gràfics de l'usuari.
- **Aplicacions d'escritori:** Amb les llibreries de GTK o Qt és poden fer de manera senzilla aplicacions d'escritori amb la seva interfície gràfica.
- **Serveis en el núvol:** Es pot utilitzar per crear serveis en el núvol que s'executen en plataformes com AWS, Azure o Google Cloud, gràcies al suport que té per la concurrència i per la seva capacitat de poder manegar varis fils d'execució simultaniament.

3. Comparació amb altres LP's

Crustal comparteix moltes similituds amb altres llenguatges de programació com podrian ser Ruby, Python i JavaScript entre d'altres. Un dels principals reptes de Crystal es que al ser relativament nou, pot dificultar trobar solucions a problemes específics. Tot i això Crystal té una sèria de ventatges repecte els altres llenguatges de programació que són:

- **Velocitat:** Compila el codi de màquina natiu el que permet que s'executi molt ràpid.
- **Legibilitat:** Té una sintaxi molt semblant a la de Ruby, lo que ho fa fàcil per escriure i llegir.
- **Tipat estàtic:** Això significa que a diferència de llenguatges com Python, les variables s'han de declarar amb un tipus concret i no es poden cambiar dinàmicament en temps d'execució. Això fa que sigui més fàcil evitar errors de codi pero fa que tot sigui més ferragós .
- **Compilat:** Crystal és compila a codi de màquina natiu, lo que significa com hem dit anteriorment que s'executa molt ràpid una vegada compilat. A més d'això, es poden detectar errors en el temps de compilat que ens poden evitar problemes més tard en el temps d'execució.

La major desventatge de Crystal es que, al ser un llenguatge relativament nou, encara no té la mateixa cantitat de llibreries i recursos disponibles que altres llenguatges més populars com Java, Pythom o C++ tenen. A més al ser un llenguatge compilat, el temps de compilació pot treure més temps que en altres llenguatges interpretats.

En resum, en comparació amb altres llenguatges de programació, Crystal destaca per el seu focus a la eficiència i la legibilitat del codi. Tot i això, que sigui tant nou pot arribar a no ser la millor opció per tots els casos d'ús degut a la seva comunitat i ecosistema que és més petit.

4. Frameworks

Crystal consta d'una gran comunitat que va creixent de manera ràpida. Això ha fet possible el desenvolupament de diferents frameworks: [2]

- **Kemal:** Un framework per desenvolupar webs de manera molt fàcil i ràpida. [3]
- **Amber:** Un framework per desenvolupar webs escrita en Crystal e insipirada en Kemal, Phoenix i altres frameworks. Utilitza programació orientada a objectes.
- **Lucky:** Un framework per fer molt bones aplicacions webs de manera ràpida, divertida i fàcil.
- **Grip:** Un microframework per construir aplicacions web RESTful. Esta dissenyat perque sigui modular i fàcil d'utilitzar.
- **Granite:** Un framework que ofereix suport per bases de dades i plantilles de renderitzat.

Cada un d'aquests frameworks té les seves propies característiques i ventatges, és important investigar cada un i escollir el que millor s'adapti a les necessitats del projecte.

5. Exemples de Codi

5.1. Hello World!

La primera linia de qualsevol llenguatge de programació ha de ser la mítica del Hello World! i així es com es fa en Crystal. [1]

```
# Aixo es un comentari en Crystal
puts "Hello World!"
```

La funció `put` posa un salt de linia automàticament.

5.2. Assignació i tipus de variables

El operador d'assignació és el (=). L'objectiu d'una assignació pot ser tant per una variable local, una variable d'instància, una variable de classe, una constamt o un mètode d'assignació, veguem els exemples en codi:

```
# Assigns to a local variable
local = 1

# Assigns to an instance variable
@instance = 2

# Assigns to a class variable
@@class = 3

# Assigns to a constant
CONST = 4

# Assigns to a setter method
foo.method = 5
foo[0] = 6
```

Aquests són els diferents tipus que incorpora Crystal amb exemples respectivament:

- **Nil:** nil
- **Bool:** true, false
- **Integers:** 18, -12, 19_i64, 14_u32, 64_u8
- **Floats:** 1.0, 1.0_f32, 1e10, -0.5
- **Char:** 'a', '\n'
- **String:** "foo\tbar", " %q(foo foo)"

- **Symbol:** :symbol, :"foo bar"
- **Array:** [1, 2, 3], [1, 2, 3] of Int32, %w(one two three)
- **Array-like:** Set1, 2, 3
- **Hash:** {"foo" => 2}, {} of String => Int32
- **Hash-like:** MyType{"foo" => "bar"}
- **Range:** 1..9, 1...10, 0..var
- **Regex:** /(foo)?bar/, /foo foo/imx, %r(foo/)
- **Tuple:** {1, "hello", 'x'}
- **NamedTuple:** {name: 'Crystal', year: 2011}, {"this is a key": 1}
- **Command:** `echo foo`, %x(echo foo)
- **Proc:** $\rightarrow (x : Int32, y : Int32)\{x + y\}$

5.3. Funcions i objectes

A Crystal, tot és un objecte. La definició d'objecte es resumeix en que té un tipus i pot respondre a certs mètodes.

Podem definir mètodes de la següent manera:

```
# Compute the nth Fibonacci number
def fibonacci(n : Int32) : Int32
  # Base case: the 0th and 1st Fibonacci numbers are 0 and 1
  return n if n < 2
  # Recursive case: the nth Fibonacci number is the sum of the (n-1)th
      and (n-2)th numbers
  return fibonacci(n-1) + fibonacci(n-2)
end

# Compute the 10th Fibonacci number
puts fibonacci(10) # => 55
```

Els mètodes poden tenir paràmetres opcionals (com a python) que en cas de que no es passi quan es crida a la funció es dona un per defecte.

```
def say_hello(recipient = "World")
  puts "Hello #{recipient}!"
end

say_hello
say_hello "Crystal"
```

Això produiria el següent output amb el comportament explicat:

```
Hello World!
Hello Crystal!
```

Els mètodes poden acceptar un bloc de codi que s'executa amb la paraula clau de `yield`. Per definir una funció que rep un bloc de codi, simplement usem `yield` dins seu i el compilador sabrà com gestionar-ho.

Per invicar un mètode i donar-li un bloc, es pot utilitzar tant do ... end o { ... }. Totes les següents invocacions són equivalents.

```
def twice
  yield
  yield
end

twice() do
  puts "Hello!"
end

twice do
  puts "Hello!"
end

twice { puts "Hello!" }
```

5.4. Classes

Per veure com es defineixen les classes crearem la classe de `Persona`. Com hem dit anteriorment en Crystal tot és un objecte, osigui que necessitarem mètodes per accedir als atributs. Per guardar aquests informació, hem de posar el prefix de @ per accedir a les variables d'instància.

```
class Person
  def initialize(name : String)
    @name = name
    @age = 0
  end

  def name
    @name
  end
```

```
    def age
      @age
    end
  end
```

Un cop tenim definida aquesta classe podem definir persones de la següent manera:

```
  john = Person.new "John"
  peter = Person.new "Peter"

  john.name # => "John"
  john.age # => 0

  peter.name # => "Peter"
```

5.5. Exemples

El següent codi és un quicksort fet en Crystal, podem veure lo llegible que és lo qual facilita molt al programador a l'hora de debugar. També podem veure la semblança de sintaxi que té amb altres llenguatges comentats anteriorment.

```
def quick_sort(arr : Array(Int32)) : Array(Int32)
  return arr if arr.size <= 1

  pivot = arr[arr.size / 2]

  left = [] of Int32
  right = [] of Int32

  for i in 0...arr.size-1
    if i == arr.size / 2
      next
    end
    if arr[i] < pivot
      left << arr[i]
    else
      right << arr[i]
    end
  end

  left = quick_sort(left)
  right = quick_sort(right)

  return left + [pivot] + right
end
```

En el següent codi podem veure com es poden utilitzar diferents expressions de control com poden ser `until` i `case`.

```
def dicotomic_search(arr : Array(Int32), value : Int32) : Int32
  left = 0
  right = arr.size - 1

  until left > right
    pivot = (left + right) / 2

    case value
    when arr[pivot]
      return pivot
    when value < arr[pivot]
      right = pivot - 1
    else
      left = pivot + 1
    end
  end

  return -1
end
```

5.6. Característiques particulars

- **Comentaris:** Els comentaris comencen amb el caràcter #.
- **Constants matemàtiques:** Algunes constants matemàtiques estan disponibles en el mòdul de `Math` com per exemple: `Math::E` o `Math::PI`.
- **C bindings:** Crystal permet bindejar llibreries existents en C sense tenir que escriure ninguna sola linia de codi en C.
- **Llibreries:** Per poder importar llibreries o altres fitxers, en Crystal és tan fàcil com posar `require "..."`.
- **Primitives de baix nivell:** Crystal proporciona algunes primitives de baix nivell que normalment s'utilitzen quan fas coses amb llibreries de C i per codi de baix nivell. Alguns exemples són: `pointerof`, `sizeof`, `offsetof`...

6. Conclusió i visió personal

Per a concloure, durant el desenvolupament del treball hem pogut observar les característiques principals del llenguatge Crystal i alguns dels pilars bàsics per a programar amb aquest. Cal recalcar que aquest llenguatge de programació, a causa de les seves moltes similituds amb altres llenguatges mes coneguts que han estat comentats en el treball, resulta molt intuïtiu i sorprenentment fàcil de llegir i comprendre. També caldria ressaltar la bona estructuració que segueix aquest llenguatge i l'ampli ventall de possibilitats que ofereix a l'hora d'escriure codi, especialment la possibilitat de desenvolupar programes de manera eficient.

Cal dir que Crystal compta amb diverses extensions per a poder enfocar el nostre en la programació orientada a objectes, eliminar la necessitat de reimplementar funcions bàsiques. Més a mes aquestes llibreries compten amb una distribució via Git, la qual cosa implica, que no necessiten d'un repositori central.

Dit tot això, personalment reconec que Crystal m'ha sorprès positivament i opino que té alguns aspectes molt prometedors. És un llenguatge que no coneixia i del qual no sàvia res i durant el procés d'aquest treball puc dir que a causa de la gran facilitat que proporciona he pogut aprendre algunes coses i fins a escriure petits fragments de codi, opino que la gran llegibilitat que proporciona pot resultar molt atractiva per a altres programadors. Crystal és un llenguatge, que avui en dia, fins i tot aquesta en creixement, però no hi ha dubte que a mesura que passi el temps i vagi adquirint una certa reputació és probable que sigui cada vegada mes usat.

Referencias

[1] Crystal. Crystal documentation. https://crystal-lang.org/reference/1.6/syntax_and_semantics/low_level_primitives.html. [Online; accessed December 14, 2022].

[2] isaced. Top crystal web frameworks. https://github.com/isaced/crystal-web-framework-stars. [Online; accessed December 13, 2022].

[3] Kemal. Kemal. https://kemalcr.com/. [Online; accessed December 13, 2022].

[4] Crystal Language. Crystal language. https://github.com/crystal-lang/crystal. [Online; accessed December 6, 2022].

[5] Meraj Molla. A first look at crystal programming language and its ecosystem. https://itnext.io/a-first-look-at-crystal-programming-language-and-its-ecosystem-efab042fbc07. [Online; accessed December 6, 2022].

[6] Wikipedia. Concurrent computing. https://en.wikipedia.org/wiki/Concurrent_computing. [Online; accessed December 6, 2022].

[7] Wikipedia. Crystal (programming language). https://en.wikipedia.org/wiki/Crystal_(programming_language). [Online; accessed December 6, 2022].

[8] Wikipedia. Object-oriented programming. https://en.wikipedia.org/wiki/Object-oriented_programming. [Online; accessed December 6, 2022].

7. Descripció i avaluació de les fonts

En aquest treball només he utilitzat fonts d'internet ja que al ser un llenguatge tant nou no disposo de material físic per consultar.

Tota la informació usada en el desenvolupament d'aquest treball ha estat obtinguda directament de la web oficial de Crystal, entre altres. La qual ofereix una àmplia explicació del funcionament del propi llenguatge i els pilars en els quals es basa Crystal. A més la web ofereix una petita introducció gratuïta per a iniciar-se en la programació amb Crystal.