



FASE 1

Proyecto Algoritmos y Estructura de Datos

Diana Corado

#191025

Orlando Cabrera

#19943

Alfredo Quezada

#191002

Lisp (List-Processing) es un lenguaje que almacena y muestra la información estructurada en listas, se apoya en utilización de funciones matemáticas. A este tipo de lenguaje se le denomina aplicativos o funcionales (porque se basa en las aplicaciones de funciones a los datos). Diferencia dos tipos de elementos básicos, el átomo y datos elementales (números, símbolos, caracteres y cadenas de caracteres).

Ejemplos de átomos: Hola, Casa, Mientras, Uno34, F59as5

Funciones en Lisp

Funciones normales, son las que se suelen incluir en las implementaciones de LISP.

Funciones de lista, para manipulación de listas:

- car L, devuelve el primer elemento de L.
- cdr L, devuelve la cola (lista - primero).
- cons x y, devuelve una lista formada por x e y.
- List x y z, devuelve la lista (x y z).
- quote x, no se evalúa x.

Predicados

- atom x, devuelve True si x es un átomo.
- numberp x, devuelve True si x es un número.
- greaterp x y, devuelve True si $x > y$.
- lessp x y, devuelve True si $x < y$.
- null x, devuelve True si x es nulo.
- and x y, devuelve x and y.
- or x y, devuelve x or y.
- not x, devuelve not x.
- eq x y, devuelve True si $x = y$.

Ejemplos de No átomos: 5456dgfv, %bc, A5

Funciones aritmética

- +, -, *, y /.
- rem x y, devuelve el módulo x/y (remainder).

Funciones de entrada y salida

- Load nombrearchivo, lee el archivo a memoria.
- print x, imprime el elemento x.
- open nombrearchivo, abre un archivo y devuelve una puntero al mismo.
- read, lee del terminal un átomo.
- help, proporciona ayuda.
- Trace, traza la función.
- bye, termina LISP.

Abstracción y encapsulamiento

Lisp, en origen, no contiene características de encapsulamiento. CLOS fue una ampliación de LISP con orientación de objetos. Características:

- Herencia múltiple
- Funciones genérica
- Metaclases (clase cuyas instancias son clases) y metaobjetos
- Creación e inicialización de objetos que permite control del proceso por parte del usuario

(EcuRed, sf)

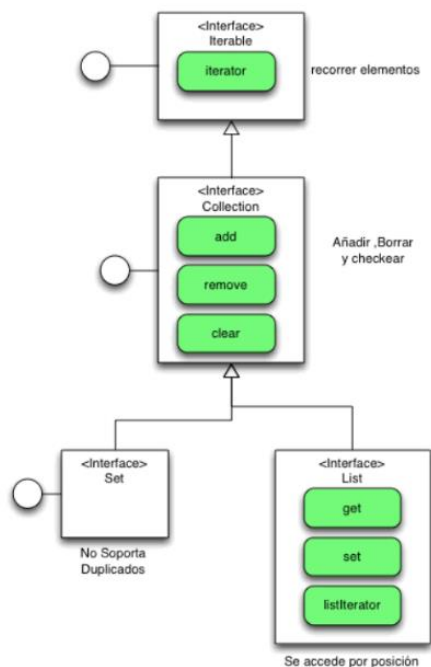
JAVA COLLECTIONS FRAMEWORKS

Una colección es un objeto que representa un grupo de objetos. Collections framework es una arquitectura unificada para la representación y manipulación de colecciones, permitiendo que las colecciones sean manipulables independientemente de la implementación.

Colección de interfaces: java.util *

Para implementar una colección se suele poner el nombre de la clase, seguido de la palabra “Implements” junto con el nombre de dicha colección (“class implements interface”).

(Oracle, sf)



Como se puede observar en la imagen del lado derecho, en el framework de colecciones la interfaz más general es Iterable el cual permite que se pueda recorrer una colección devolviendo un iterador

Seguidamente a eso aparece la interfaz Collection la cual define un conjunto de elementos recorribles.

Y finalmente se observa las interfaces List (añade métodos que nos permiten acceder por posición a cada elemento de la lista) y Set (define que no existan elementos repetidos) las cuales son hijos de Collection.

Las clases que implementan las interfaces Set y List en el Collections Framework con las siguientes:



Cecilio Caules, 2015)

IMPLEMENTACIONES EN EL INTERPRETADOR

- **HashTable**

Es una estructura de datos que utiliza una función hash para identificar los datos mediante una llave o clave. Esta función transforma una llave a un valor índice de un arreglo de elementos. El primer elemento será la clave y el segundo será el valor para almacenar. No permite null como clave. (Meza, 2013).

- ✓ Nos permite tener acceso a un dato por medio de una clave. Como si fuera un diccionario.

- **List<E>**

Son elementos en una secuencia particular que mantienen un orden y permite duplicados. Tipos de construcciones:

ArrayList<E>: La ventaja de esta lista es que el acceso a un elemento en particular es ínfimo. Su desventaja es que, para eliminar un elemento, se ha de mover toda la lista para eliminarlo. (Acedo, 2011).

- **Vector<E>**

Es igual que ArrayList, pero sincronizado. Es decir, si se usa varios hilos, no se tendrá de qué preocuparse hasta cierto punto. (Acedo, 2011).

- ✓ Es una manera útil para almacenar y manipular cantidades grandes de datos, así como en un arreglo.

- **Árbol**

Al contrario que las listas, es una estructura de datos no lineal. Los árboles pueden tener dos o más sucesores.

Consiste en un conjunto de nodos o vértices y un conjunto de aristas o arcos que satisface los siguientes requisitos:

Existe una jerarquía de nodos, de forma que a cada nodo hijo le llega una arista de otro nodo padre. De esta forma se establece la relación padre-hijo. El nodo donde comienza la jerarquía se llama nodo raíz, este nodo no es hijo de ningún nodo padre. (Roldán, 2016).

- ✓ Nos permite representar con sintaxis abstracta. Un árbol binario cuyas hojas se asocian a los operandos y el resto de los nodos que contiene los caracteres asociados a los operadores.

PROGRAMAS LISP

- Serie Fibonacci:

```
(DEFUN fibo (N)
  (COND
    ((< N 2) 1)
    (T (+
        (fibo (- N 1))
        (fibo (- N 2))
      )
    )
  )
)
```

- Conversor de grados:

```
(DEFUN convert ()
  (format t "Ingrese los grados" )
  (LET (fahr )
    (SETQ fahr (read fahr))
    (APPEND '(celsius is) * (- fahr 32) (/ 5 9)) )
  )
)
```

Referencias

- EcuRed (sf), LISP, Extraído de: <https://www.ecured.cu/LISP>
- Oracle (sf), Collections Framework Overview. Extraído de: <https://docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html>
- Caules, Cecilio Álvarez. 17 de febrero de 2015. "Java Collections List vs Set". Extraído de: <https://www.arquitecturajava.com/java-collections-list-vs-set/>
- Acedo, J. (2011). Colecciones y tipos genéricos en Java. Obtenido de: <http://programacion.jias.es/2011/10/colecciones-genericos-en-java/>
- Meza, F. (2013). Usar una Hashtable Java. Obtenido de: <http://lineadecodigo.com/java/usar-una-hashtable-java/>
- Roldán, A. (2016). Árboles en Java. Obtenido de: <http://www.ciberaula.com/articulo/arboles>