**Lisp**

Lisp es una familia de lenguajes de programación multiparadigma que se basan en la notación matemática del cálculo lambda. Lisp fue creado por John McCarthy en 1958 y se considera el segundo lenguaje de alto nivel más antiguo después de FORTRAN. Lisp ha sido muy influyente en el campo de la inteligencia artificial y ha introducido muchas ideas innovadoras en la ciencia de la computación, como la recursividad, las estructuras de datos de árbol, el manejo de memoria automático, los tipos dinámicos y la metaprogramación.

Lisp tiene varios dialectos, entre los que se destacan Common Lisp, Scheme, Emacs Lisp y Clojure. Cada dialecto tiene sus propias características y aplicaciones, pero comparten la sintaxis homoicónica basada en listas entre paréntesis, conocida como expresiones S. Una expresión S puede representar tanto datos como código, lo que permite a los programadores de Lisp crear macros y lenguajes específicos de dominio.

Lisp se puede clasificar como un lenguaje de programación funcional, ya que permite definir y aplicar funciones de manera abstracta y general, sin efectos secundarios ni asignaciones de variables. La programación funcional se basa en el concepto de que las funciones son valores de primera clase que se pueden pasar como argumentos, devolver como resultados o almacenar en estructuras de datos. La programación funcional también hace uso de la recursión, la inmutabilidad, las funciones de orden superior, las funciones anónimas y las funciones puras.

La programación funcional se diferencia de la programación orientada a objetos (OOP) en que esta última se basa en el concepto de que los datos y el comportamiento se agrupan en objetos que se comunican mediante mensajes. La programación orientada a objetos se enfoca en el estado y la identidad de los objetos, mientras que la programación funcional se enfoca en las transformaciones y las operaciones de los datos. La programación orientada a objetos facilita el encapsulamiento, la abstracción, el polimorfismo y la herencia, mientras que la programación funcional facilita la composición, el modularidad, la referencialidad transparente y la concurrencia.

El Java Collections Framework es un conjunto de interfaces y clases que proporcionan una arquitectura unificada para almacenar y manipular colecciones de objetos. Las colecciones son estructuras de datos que agrupan varios elementos, como listas, conjuntos, mapas, colas, etc. El Java Collections Framework ofrece varias ventajas, como reducir el esfuerzo de programación, aumentar el rendimiento, facilitar la interoperabilidad, fomentar la reutilización y proporcionar algoritmos útiles.

El Java Collections Framework se compone de los siguientes elementos:

Interfaces de colección. Representan diferentes tipos de colecciones, como Set, List, Queue, Deque y Map. Estas interfaces forman la base del framework y definen las operaciones comunes que se pueden realizar sobre las colecciones, como agregar, eliminar, buscar, iterar, etc.

Implementaciones de colección. Son clases concretas que proveen las funcionalidades de las interfaces de colección. Existen varias implementaciones de propósito general, como ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap, etc. También existen implementaciones de propósito especial, como las que proveen sincronización, inmutabilidad, ordenamiento, etc.

Algoritmos de colección. Son métodos estáticos que realizan funciones útiles sobre las colecciones, como ordenar, mezclar, buscar, copiar, etc. Estos métodos se encuentran en la clase Collections y se pueden aplicar a cualquier tipo de colección que implemente la interfaz Collection o Map.

Infraestructura de colección. Son interfaces y clases que proveen soporte esencial para las interfaces de colección, como Iterator, Comparator, Comparable, etc.

**Referencias**

Azcoitia, S. S. (2023, 29 junio). Historia de Lisp y su uso en redes neuronales - Parte I. *Telefónica Tech*. <https://telefonicatech.com/blog/historia-lisp-uso-redes-neuronales>

Guerra-Hernández, A. (2020). Programación para la Inteligencia Artificial - Introducción a Lisp. Universidad Veracruzana. Recuperado de <https://www.uv.mx/personal/aguerra/files/2020/09/pia-06.pdf>

Velasco, J. (2015, 9 julio). Historia de la tecnología: Lisp. *Hipertextual*. <https://hipertextual.com/2011/10/historia-de-la-tecnologia-lisp>