Análisis del lenguaje LISP y sus respectivos paradigmas

Lisp (LISt Processing) fue creado por John McCarthy en 1958 mientras trabajaba en el MIT[1]. Lisp es el segundo lenguaje de programación de alto nivel más antiguo que aún se usa comúnmente, después de Fortran. El nombre *LISP* deriva de "LISt Processor"..

Funcional

Imperativa

Lisp fue uno de los primeros lenguajes en adoptar el paradigma funcional, con características como funciones de orden superior, recursión y evaluación perezosa. [2]

Además, consideramos que Lisp también tiene bases en los paradigmas funcional e imperativo. [3]

Clisp fue creado en 1987 Bruno Haible y Michael Stoll (Common Lisp) este lenguaje es muy utilizado en la robótica y la ingeniería artificial, fue desarrollado como parte de la estandarización de Lisp y para tener más eficiente que el lenguaje lisp este incluye un intérprete y un compilador para el desarrollo. El Common Lisp utiliza los paradigmas funcionales, el imperativo y la orientada a objetos [4]

Funcional

Orientada a

Objetos

Imperativa

Referencias

[1]. My WordPress, “▷【 Lisp 】 lenguaje de programación en 2025,” *CodificaYA -*, May 09, 2023. <https://codificaya.com/lisp/>

[2] J. L. Bustos, “¿Qué son los paradigmas de programación y qué tipos hay?,” *KeepCoding Bootcamps*, Apr. 25, 2024. <https://keepcoding.io/blog/paradigmas-de-programacion/>

[3] Anier, “LISP,” *EcuRed*, Dec. 13, 2011. <https://www.ecured.cu/LISP>

[4] “Common Lisp,” *EduRed*. <https://www.ecured.cu/Common_Lisp>