

A8 Criterios que debe cumplir la sintaxis de un lenguaje

Note Title

5/2/2010

Los criterios generales son:

Prof. L.Ortiz

- Fácil de leer

- Una vez escrito el programa, las oraciones deben reflejar que hace el programa sin la necesidad de añadirle comentarios (un lenguaje con muchos adornos o palabras osciosas en su sintaxis).
- La sintaxis debe reflejar claramente la semántica de la oración. Por ejemplo, una oración para sumar dos números no debe parecer una instrucción de multiplicación.

- Fácil de escribir

- Lo que hace un lenguaje fácil de leer, provoca que el lenguaje sea difícil de escribir.
- Las estructuras sintácticas deben ser concisas (ir al punto sin dar muchos rodeos). En otras palabras, las estructuras deben ser claras, precisas y naturales.
- Los identificadores no deben tener restricciones (largo, tipo, caracteres válidos, etc).
- Para los operadores se deben usar símbolos y no palabras.

- Fácil de traducir

- Los criterios que hacen un programa fácil de leer y fácil de escribir van en contra de la traducción. Por ejemplo, LISP no es fácil de leer ni fácil de escribir pero es fácil de traducir. Esto se debe a que LISP no tiene una sintaxis compleja para describir sus programas. Eso permite que la traducción se haga directamente. Por ejemplo, COBOL es un lenguaje que su sintaxis es fácil de leer, usa muchos adornos, lo que causa que el traductor tenga que remover los adornos antes de hacer la traducción final.

- Sin ambigüedad

- Una construcción ambigua permite que se interprete en distintas maneras. Por ejemplo, en FORTRAN se usan los paréntesis para indicar un elemento de un arreglo y la llamada a una función. Por lo tanto, $a(3,2)$ es un arreglo o la llamada a una función. Eliminar la ambigüedad provoca que las estructuras no sean naturales debido a la introducción de símbolos extraños. Por ejemplo, C++ usa $[]$ para indicar un elemento de un arreglo.