Paradigmas de Lenguajes de Programación

Universidad Nacional de la Patagonia Austral
Unidad Académica Río Gallegos

Primeros Lenguajes

- Surgen en el Siglo XIX, por el matemático Charles Babagge y Ada Lovedby.
- Máquina Analítica que programaba tarjetas perforadas.

Primeros Lenguajes

- Diseño propuesto por Babagge, cinco unidades:
  - Unidad de Entrada
  - Memoria
  - Unidad de Control
  - Unidad Aritmético-Lógica
  - Unidad de Salida
  - En los años 40 surgen las primeras computadoras modernas.

Primeros Lenguajes

- Denominados código o lenguaje de máquina se basaban en código binario y se especializaban según las aplicaciones.
- Programación lenta y tediosa, datos e instrucciones en código binario, acceso a posiciones de memoria.

Primeros Lenguajes

- A principios de los 50, surge nueva simbología, código de ensamblaje.
- Operaciones representadas mediante abreviaturas: ADD, STORE, etc. ASSEMBLER
- 1954 FORTRAN (FORmula TRANslation)
- COBOL: registros, estructuras de datos separada de la ejecución.
- Algol60: descripción de algoritmos. Pascal, C y Ada. Bloques inicio-fin, declaración de tipos de variables, recursión, paso de parámetros. Ejecución basado en Pilas.

Los 60

ADAM
AED
AESOP
AIMACO
ALGOL
ALGY
ALTRAN
AMBIT
AMTRAN
APL
BACIAC
BASEBALL
BASIC
BUGSYS
C-10
CLIP
CLP
COBOL
COGENT
COGO
COLASL
COLINGO
COMIT
CORAL
CORC
CPS
DAS
DATA-TEXT
DIALOG
DEACON
DIAMAG
DIMATE
DOCUS
DSL
DYANA
DYNAMO
DYSAC
FLAP
FLOW-MATIC
FORMAC
FORTRAN
FORTRANSIT
FSL
GAT
GECOM
GPL
GPSS
GRAF
GRAF
IDS
IT
JOSS
JOVIAL
L
LDT
LISP
LOLITA
LOTIS
MAD
MADCAP
MAP
MATHLAB
MATH-MATIC
META
MILITRAN
MIRFAC
NELIAC
OCAL
OMNITAB
OPS
OPS
PENCIL
PRINT
QUIKTRAN
SFD-ALGOL
SIMSRIPT
SIMULA
SNOBOL
SOL
SPRINT
STRESS
STROBES
TMG
TRAC
TRANDIR
TREET
UNCOL
UNICODE

Los 70

- Los diseñadores se enfocaron en la simplicidad y consistencia.
- PASCAL (1971): pequeño, simple, eficiente y estructurado.
- C (1972): reduce la complejidad del sistema de tipos y el entorno en tiempo de ejecución, mayor acceso a máquina adyacente.
- Mecanismos de abstracción de datos, concurrencia y verificación.

Los 80

- Consolidación de Lenguajes Imperativos.
- Crecimiento de la programación orientada a objetos. (Objetive C, Objetive Pascal, Modula-3)
- Sistemas a gran escala de unidades de código, sistema de módulos relacionados a construcciones de programación genérica.

Los años 90

- Crecimiento de Internet y del mercado de las PC.
- Sistemas Operativos basados en ventanas.
- Hasta 1993 C, luego en 1995 aparece Java.
- Surgen las bibliotecas de scripts.
- Programas breves, dinámicos, estructuras de alto nivel de datos, sintaxis extensible, carencia de verificación de datos.

Tendencias Actuales

- Investigación e industria, la tendencia de los LP se enfoca en :
  - Apoyo a la programación concurrente y distribuida.
  - Mecanismos de verificación al lenguaje, seguridad y confiabilidad.
  - Desarrollo orientado a componentes.
  - Programación Orientada a Aspectos.
  - Integración de bases de datos.

Lenguajes de Programación de los 2000

- ActionScript
- C#
- D
- Visual Basic Net
- Groovy
- Grace
- Assembly language
- Python
- Ruby

Tendencias Actuales

- Evolución de Lenguajes de Programación:
  - Recursos y tipos de computadoras
  - Aplicaciones y necesidades de los usuarios
  - Métodos de programación
  - Estudios teóricos
  - Estandarización

Definición

- *Un LP es un conjunto de caracteres, reglas para su combinación y reglas especificando sus efectos cuando son ejecutadas por una computadora, las cuales deben cumplir con las siguientes características.*

Definición: Características

- El usuario no requiere conocimientos de lenguaje de máquinas.
- Es independiente de la máquina.
- Es posible traducirlo al lenguaje de máquina.
- Es cercano al problema específico que se resuelve en el lenguaje de máquina.

Definición

- *Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras.*
- *Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.*
- *Programación: escribir, probar, compilar , depurar y mantener código.*

Características y Elementos de los Lenguajes de Programación

- Definen un proceso que se ejecuta en una computadora.
- Es de alto nivel, cercano a los problemas que se quieren resolver. (Abstracción)
- Permite crear nuevas abstracciones que se adapten al dominio que se programa.
- Ideas simples en ideas complejas: expresiones primitivas, mecanismos de combinación, mecanismos de abstracción.

Paradigmas de Programación

- Un paradigma es un conjunto de reglas, patrones y estilos de programación utilizados por un grupo de Lenguajes de Programación.

Paradigma Imperativo

- Se describen las sentencias que permiten modificar el estado de un programa. Secuencia de acciones que expresan cómo debe resolverse un problema. (subrutinas o funciones)

Programación Estructurada

- Datos y procedimientos separados, sin relación, se busca el procesamiento de un conjunto de datos de entrada para obtener otros de salida.

  - Primero : funciones y procedimientos
  - Segundo: estructura de los datos

Paradigma Declarativo

- No es necesario definir un algoritmo, se detalla la solución del problema, en lugar de *Cómo* llegar a la solución.
- Solución a problemas específicos a través de mecanismos internos de control, no se especifica exactamente.
  - Paradigma funcional: evaluación de expresiones y funciones matemáticas.
  - Paradigma Lógico: reglas y sentencias lógicas.

Paradigma Orientado a Objetos

- Los programas se definen en términos de clases de objetos.
- Los programas se expresan como un conjunto de objetos que colaboran entre ellos para realizar tareas.
  - Encapsulamiento
  - Herencia
  - Polimorfismo

Paradigma Orientado a Aspectos

- Surge en concepto de *Aspecto*, un módulo que encapsula determinada funcionalidad que se encuentra dispersa por diferentes partes del Sistema.
- *AspectJ*.

Paradigmas Actuales

- Combinación de varios paradigmas en un mismo lenguaje:
  - Programación lógico-funcional: Curry, TOY
  - Programación funcional-concurrente: Erlang (80'), Concurrent Haskell (87'- 10')
  - Programación funcional-paralela: GpH, Eden (90'), Data Parallel Haskell (02')
  - Programación lógico-funcional-concurrente-orientado a objetos: Oz (99 - 05')

Paradigmas Actuales

- Lenguajes O.O. incorporan construcciones funcionales:
  - JavaScript: maps, folds, closures
  - Java 8: maps, folds, lambda, abstractions
  - Ruby: lambda abstractions, closures, firts-class, continuations
  - Python: maps, reduce, filters, list comprehensions, lambda abstractions

Paradigmas Actuales

- Los lenguajes han evolucionado enfocándose en los siguientes aspectos:
  - Creciente nivel de abstracción
  - Mecanismos de modularidad
  - Mecanismos de seguridad
  - Reutilización

Lenguajes de Programación de los 2020+

- Python
- Java
- Javascript
- C#
- PHP
- C/C++
- R
- Objective-C
- Swift
- Matlab