

Lenguajes de Programación

Introducción

Dr. Mauricio Araya

Primer Semestre 2014

Agenda

Índice

1. Sobre el Curso	1
2. Introducción a los Lenguajes de Programación	2
3. Bibliografía	6

1. Sobre el Curso

Al finalizar el curso se espera que el alumno sea capaz de cumplir los siguientes objetivos:

Objetivos

- Comprender los principales paradigmas de programación
- Entender las herramientas conceptuales para realizar una evaluación crítica de los lenguajes existentes y futuros
- Ser capaz de seleccionar el lenguaje más apropiado para resolver un determinado tipo de problema
- Preparar a los alumnos para cursos más avanzados

Contenidos del Curso

- Conceptos Fundamentales
- Lenguajes Imperativos
- Lenguajes Orientados a Objetos
- Lenguajes Funcionales
- Lenguajes Lógicos
- Lenguajes de Scripting

Metodología

- *Cátedra*: Presentación de paradigmas y constructores relevantes a los lenguajes de programación
- *Ayudantía*: Revisión y aplicación de los conceptos de cátedra a lenguajes reales
- *Certámenes y Controles*: Evaluación de los contenidos pasados en clase
- *Tareas*: Aplicación individual de los conceptos a 5 lenguajes seleccionados

Cálculo de la Nota Final

$$N_{Final} = 0,66 \cdot N_C + 0,34 \cdot N_A$$

$$N_C = \frac{C_1 + C_2 + N_Q}{3}$$

$$N_Q = \frac{\sum_1^5 Q_i - \min_{j \in [1,5]} \{Q_j\}}{4}$$

$$N_A = \frac{\sum_1^5 T_i - \min_{j \in [1,5]} \{T_j\}}{4}$$

IMPORTANTE: si $N_C < 50$ o $N_A < 50 \rightarrow$ reprobación

2. Introducción a los Lenguajes de Programación

¿Qué es un Lenguaje de Programación?

“To me programming is more than an important practical art. It is also a gigantic undertaking in the foundations of knowledge” — Grace Hopper

Wikipedia: Un lenguaje de programación es una técnica estándar de comunicación que permite expresar las instrucciones que han de ser ejecutadas en una computadora.

¿Por qué Estudiar Lenguajes?

“Para qué aprender tantos, ¿si igual vamos a ocupar solo Java en la vida real!”

- Incrementar capacidad de expresar ideas
- Mejor base para elegir lenguajes apropiados
- Incrementar habilidad de aprender nuevos lenguajes
- Mejorar el uso de los lenguajes
- Mejor comprensión del significado de la implementación

Dominios de Programación

“Para el que tiene un martillo, todo tiene cara de clavo”

- Aplicaciones de negocio
- Programación de sistemas
- Aplicaciones científicas
- Inteligencia Artificial
- Scripting (Shell, PERL, etc.)
- Lenguajes de propósitos especiales (e.g. SQL, Simula, etc.)

Nivel de Abstracción de los Lenguajes

“Los programadores de verdad pueden escribir código ensamblador en cualquier lenguaje.” - Larry Wall

- Lenguajes de Bajo Nivel
 - Lenguaje de Máquina
 - Assembler
- Lenguajes de Alto Nivel
 - C, Pascal, Cobol, Fortran
 - Java, C++, Smalltalk, C#
 - Perl, Python, Ruby, Shell
 - Scheme, Lisp, Prolog
- Lenguajes de otros niveles...
 - de Medio Nivel (C, Fortran, Cobol)
 - de ¿Altísimo Nivel? (Matlab, Kodu, LabVIEW)

Métodos de Implantación

- Lenguajes Compilados
 - Código de Fuente → Binarios ejecutables
 - como C
- Lenguajes Interpretados
 - Código de Fuente → Interpretado línea por línea
 - como Python
- Lenguajes Híbridos
 - Se compilan a un lenguaje intermedio
 - Como al Bytecode de Java

Ambientes de Programación

- Sistema de Archivos
- Editores
- Enlazadores
- Cargadores
- Compiladores
- Depuradores

Paradigmas de Programación

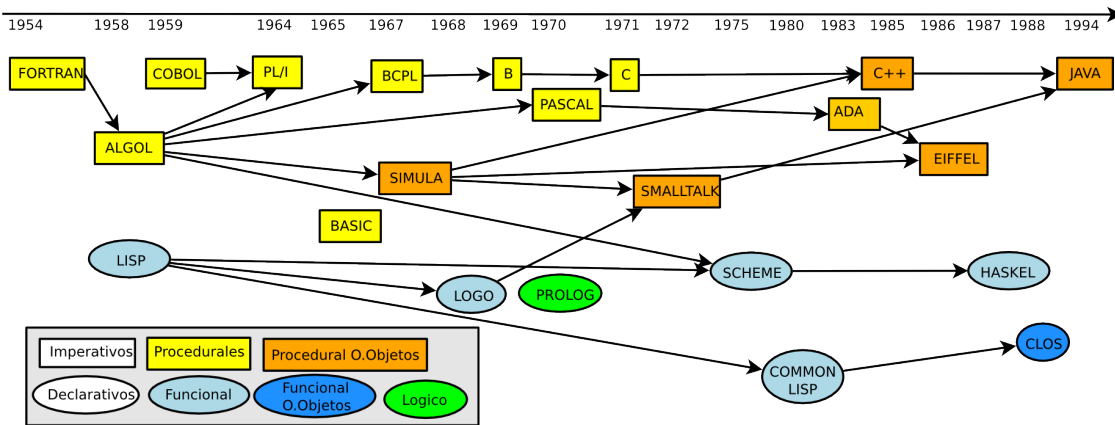
***Paradigma:** Conjunto de opiniones, valores y métodos compartidos por los integrantes de un colectivo.*

- Distintas visiones (o escuelas) para programar
- A veces para distintos propósitos, a veces distintas aproximaciones
- Los más conocidos:
 - **Imperativos:** C, Pascal, Basic
 - **Orientados a Objetos:** Java, C++, Smalltalk
 - **Scripting:** Perl, Shell, Python
 - **Funcionales:** Scheme, Lisp
 - **Lógicos:** Prolog

Evolución de los Paradigmas

- **Lenguajes de Máquina:** primeros programas; Babbage (1837), Turing (1936), Zuse (1941), ENIAC (1946).
- **Lenguajes de Ensamblaje** (assembly): código simbólico y herramientas de software; IBM (1954).
- **Primeros Lenguajes de Alto Nivel:** independiente de máquina; FORTRAN (1957), LISP (1958), COBOL (1959).
- **Lenguajes Estructurados:** ALGOL (1960), ALGOL 68, PASCAL (1970), C (1972) y ADA (1979).
- **Lenguajes Orientados a Objeto:** Simula (1967), Smalltalk (1980), C++ (1983), Eiffel (1986), Java (1995).
- **Lenguajes de Scripting:** JCL (1964), RUNCOM (1964), SH (1971); GREP (1973), AWK (1977); TCL (1988); PERL (1987), PYTHON (1991); JavaScript (1995), PHP (1995), Ruby (1995).

Árbol de Evolución



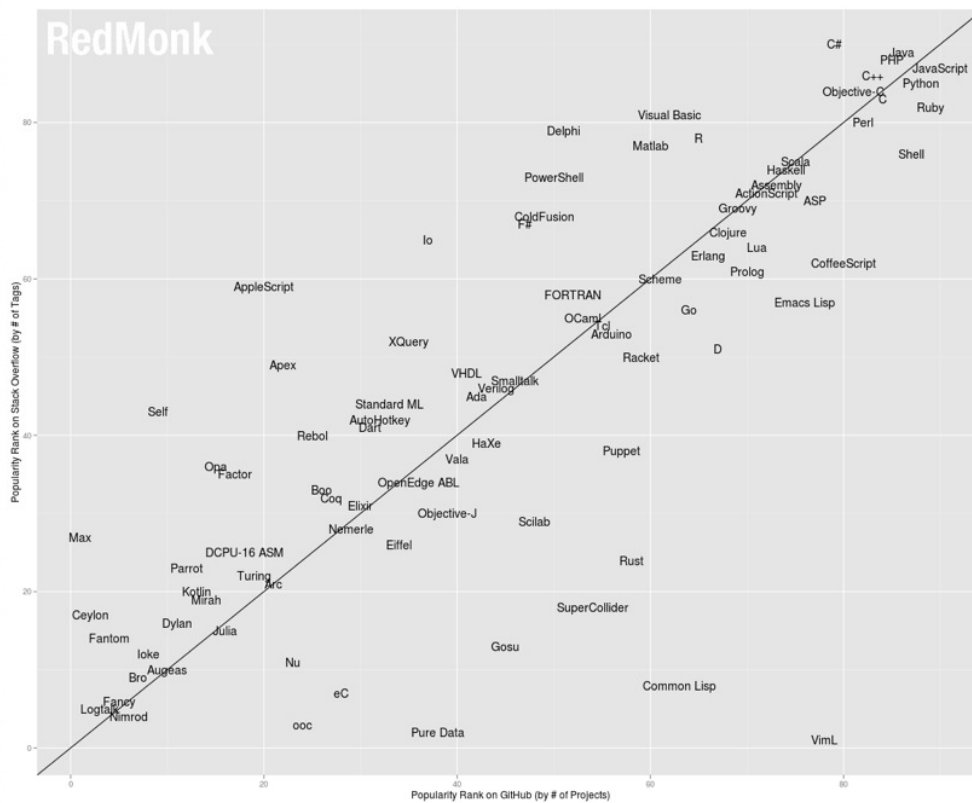
Estándares

- Estándares de Jure
- Estándares de Facto
- Ejemplos:
 - ANSI C
 - ISO C++
 - J2EE Standard
 - ANSI SCHEME
 - ISO PROLOG

Otros Modelos de Programación

- Programación basada en Eventos : flujo del control está determinado por eventos que procesa el manejador de eventos.
- Programación basada en Aspectos : programación basada en los conceptos y la incumbencia de sus relaciones.
- Programación Concurrente : conjunto de procesos cooperativos que se pueden ejecutar en paralelo. Se requiere sincronización en el acceso a recursos compartidos
- Programación Real-Time : programación basada en tiempos límites de ejecución. Normalmente usado para procesos críticos o con consecuencias catastróficas
- Programación Visual : Permite crear programa manipulando objetos gráficos. Normalmente se integra con otros lenguajes
- etc..

Popularidad de los Lenguajes



3. Bibliografía

Bibliografía General

Referencias

- [1] Robert Sebesta, *Concepts of Programming Languages*, 10th Edition, Addison Wesley, 2011.
- [2] Kenneth C. Loudon, Kenneth A. Lambert, *Programming Languages: Principles and Practices*, Course Technology, 3rd Edition, 2011.
- [3] Allen Tucker, Robert Noonan, *Programming Languages*, McGraw-Hill, 2nd edition, 2006.

Bibliografía por Lenguaje

Referencias

- [1] Brian W. Kernighan & Dennis M. Ritchie, *The C Programming Language*.
- [2] Ken Arnold & James Gosling, *The Java Programming Language*.
- [3] Mark Lutz, *Programming Python*.
- [4] R. Kent Dybvig, *The Scheme Programming Language: ANSI Scheme*.

[5] Ivan Bratko, *Prolog: Programming for Artificial Intelligence*.