UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

CONCEPTOS DE LENGUAJES DE PROGRAMACION

Año 2002

Carrera: ***Licenciatura en Informática***
Año: ***3°***
Duración: ***Anual***
Profesor: ***Lic. María B. Rodríguez***

1) Motivos para estudiar los conceptos de los lenguajes de Programación. Objetivos del estudio de los conceptos de lenguajes de programación. Influencia del lenguaje de programación en el proceso de desarrollo de software. Clasificación de lenguajes. Paradigmas. Metodologías de desarrollo de software y lenguajes de programación. Objetivos al diseñar lenguajes. Criterios de evaluación: Facilidad de escritura, legibilidad, confiabilidad, mantenibilidad. Perspectiva y evolución histórica. Aporte de los distintos lenguajes: desde Assembler hasta Java. Estandarización de un lenguaje.
2) Estructuras de lenguajes de programación. Qué mirar en un lenguaje. Sintaxis y semántica. Sintaxis. Como se puede definir. Gramáticas. BNF. Gramática ambigua. Gramáticas libres y sensibles al contexto. BNF extendida. Diagramas sintácticos. Semántica estática y dinámica. Semántica. Semánticas formales. Procesamiento de un lenguaje, interpretación y traducción. Pasos de la traducción.
3) Semántica operacional. Concepto de ligadura y momentos de ligadura. Variables: atributos, r-valor y l-valor. Constantes e inicialización.
   Unidades de programa: atributos, instancia. Unidades genéricas. Definición de Procesador semántico abstracto. Estructura de los lenguajes de programación: Modelos de ejecución: estructura estática, basada en pila, dinámica. Ejemplos de cada tipo.
   Datos compartidos. Ambiente de referencia. Cadena dinámica. Cadena estática. Ambiente no local. Pasaje de parámetros.
   Parámetros. Modelos semánticos: parámetros de entrada, parámetros de salida, parámetros de entrada/salida – valor, referencia, por nombre. Rutinas como parámetros.
4) Tipos de datos. Concepto de tipo. Abstracción de datos en lenguajes primitivos. Datos predefinidos y definidos por el usuario. Constructores: producto cartesiano, correspondencia finita, secuencias, recursión, unión discriminada, conjunto de partes. Tipos provistos por Algol, Pascal, Simula y Ada. Implementaciones. Arreglos. Punteros. Coerción y compatibilidad. Areas conflictivas y soluciones propuestas. Sistema tipado, chequeo de tipos.

5) Abstracción de control. Variables, expresiones, sentencias. Expresiones y sentencia de asignación. Estructuras de control a nivel de sentencia: secuencia, selección,

iteración. Estructuras de control a nivel de unidad, llamadas explícitas a unidades subordinadas, unidades llamadas implícitamente.

6) Abstracción de datos. Definición de tipo. Tipos definidos por el usuario. Mecanismos para proveer abstracción: Simula67-Clases, Modula II-módulo, Clu-cluster, Ada-paquetes. Tipos genéricos y parametrizados.
7) Estructuras de control no jerárquicas. Llamadas implícitas y explícitas. Modelo maestro esclavo. Manejo de excepciones: modelo de terminación y de reasunción. Manejadores, alcance, parametrización. Distintos mecanismos Pl/l, Clu, Ada, JAVA.
8) Estructura de un programa. Modularidad. Encapsulamiento. Interfase. Implementación.
9) Paradigma funcional. La Arquitectura de Von Newman. Características imperativas. Variable, asignación, repetición. Funciones matemáticas. Función como algoritmo. Formas de evaluación: evaluación más interna, evaluación truncada, evaluación selectiva. Efectos laterales. Funciones matemáticas vs. Funciones de programación. Origen de la programación funcional. De la expresión al lenguaje funcional. Componentes de un lenguaje funcional: Funciones primitivas. Formas Funcionales (construcción, inserción, aplicar todo, condición), Aplicación, Estructura para almacenar datos. Lisp. ML. Comparación con la programación imperativa.
10) Paradigma orientado a objetos. Que hace que un sistema o un lenguaje sea OO?: Clases. Herencia. Polimorfismo. Simulación: Método Protocolo, TAD: objeto, Clasificación: instancia, Polimorfismo: ligadura dinámica – ligadura estática, Herencia: Comparación con la programación imperativa.

Bibliografía

GHEZZI C. – JAZAYERI M.: Programming languaje concepts. John Wilwy and Sons. (1998) 3er. Ed.
PRATT: Programming Languajes. Design and Implementation. Prentice Hall (1998) 3er. Ed.
SETHI R.: Programming languajes: concepts and constructs. Addison – Wesley (1996) 2nd. Ed.
SEBESTA: Concepts of Programming languajes. Benjamin/Cumming. (1998) 2nd. Ed.
HOROWITZ: Funddamentals of Programins Languajes. Spring- Verlar (1984).
WILSON – CLARK: Comparative Programming Languaje. Addison -Wesley (1993).
KAMIN: Programming languajes. Addison – Wesley (1991).
BAL H. – GRUNE D.: Programming Languaje Essentials. Addison – Wesley (1994).
FINEKL R.: Advanced Programming languajes design. Addison – Wesley (1996).
BUDD: An Introduction to object-Oriented programming. Addison – Wesley (1991).
FIELD – HARRISON: Functional programming. Addison – Wesley (1989).
STROUSTRUP: C++. El lenguaje de programación. Addison – Wesley (1993).
KERNIGHAN – RITCHIE: The C Programming Languaje.
LALONDE – Inside Smalltalk (introducción).
HERMANN: Algol 68 Languaje algorithmique.
HABERMAN: Ada for experienced Programmers. Addison – Wesley (1983).
LISKOV B. – Clu.