UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

GUÍA DOCENTE *CURSO: 2007/08*

14107 - FUNDAMENTOS DE LENGUAJES

ASIGNATURA: *14107 - FUNDAMENTOS DE LENGUAJES*
CENTRO: *Escuela de Ingeniería de Telecomunicación y Electrónica*
TITULACIÓN: *Ingeniero de Telecomunicación*
DEPARTAMENTO: *INGENIERÍA TELEMÁTICA*
ÁREA: *Ingeniería Telemática*
PLAN: *13 - Año 200* **ESPECIALIDAD:**
CURSO: *Cuarto curso* **IMPARTIDA:** *Segundo semestre* **TIPO:** *Optativa*
CRÉDITOS: *4,5* **TEÓRICOS:** *3* **PRÁCTICOS:** *1,5*

Descriptores B.O.E.

Diseño de Lenguajes: Conceptos y Paradigmas; valores, almacenamiento, declaraciones, ámbitos, tipos de datos, abstracción y encapsulado; Paradigma imperativo, concurrente, orientado a objetos, funcional y lógico. Procesadores de Lenguajes: traductores e intérpretes: análisis léxico, análisis semántico y generación de códigos.

Temario

Tema 1: Introducción a los lenguajes de programación. Modelos de computación. Características comunes de los lenguajes modernos. Sintaxis y semántica. Gramáticas: clasificación de Chomsky. Compiladores e intérpretes (4 horas)

PRIMERA PARTE: PROCESADORES DE LENGUAJE

Tema 2: Procesadores de lenguajes. Análisis lexico. Separadores, operadores, palabras reservadas e identificadores. Expresiones regulares. Autómatas. Alternativas de implementación (2 horas)

Tema 3: Análisis sintáctico. Métodos descendente. Descenso recursivo. Métodos guiados por tablas. LALR. (4 horas).

Tema 4: Análisis semántico.Atributos y gramáticas con atributos. Atributos heredados y sintetizados. Resolución de nombres (2 horas).

Tema 5: Generación de código. Estructuras de control. Evaluación de expresiones. Dynamic dispatching. Optimización (2 horas).

Tema 6: Soporte de ejecución. Registro de activación de subprogramas. Gestión de ámbitos. Gestión de parámetros. Funciones con número de parámetros variable. Objetos de tamaño dinámico. Gestión de memoria dinámica (2 horas).

SEGUNDA PARTE: DISEÑO DE LENGUAJES

Tema 7: Lenguajes imperativos. Estructuras de control. Tipos de datos y su representación. Declaraciones de tipos y subtipos de datos. Evaluación eficiente de expresiones. Subprogramas y

módulos. Paso de parámetros. Anidación. Soporte para módulos en Ada, Java y C++ (4 horas).

Tema 8: Programación orientada a objetos. Constructores y destructores. Herencia simple. Polimorfismo. Dynamic Dispatching. Clases abstractas. Herencia múltiple. Comparación entre Ada, Java y C++ (2 horas).

Tema 9: Programación con genéricos. Parámetros de genéricos. Instancias. Comparación entre Ada y C++ (2 horas).

Tema 10: Excepciones. Propagación de excepciones. Manejo de excepciones. Ambito de excepciones. Información asociada a excepciones. Comparación entre Ada, C++ y Java (2 horas).

Tema 11: Programación Concurrente. Declaración de tareas. Activación de tareas. Comunicación y sincronización. (2 horas)

Tema 12: Programación funcional y Programación Lógica. Listas. Orden de evaluación. Predicados (2 horas).

Requisitos Previos

Se recomienda que el alumno tenga aprobadas las asignaturas Fundamentos de Programación y Programación de la ETSIT, así como la mayoría de las asignaturas de tercer curso del plan de estudios de la titulación de Ingeniero de Telecomunicación o estudios similares, y que tenga conocimientos básicos de uso de sistemas UNIX (por ejemplo, Linux).

Objetivos

A nivel teórico, el principal objetivo de la asignatura es que el alumno se familiarice con el soporte que proporcionan los lenguajes de programación modernos a los principales paradigmas de programación, así como a los conceptos subyacentes y su uso: tipos de datos, constructores, polimorfismo, concurrencia, genéricos, excepciones, etc.

A nivel práctico, el objetivo es familiarizar al alumno con el procesado de lenguajes. Para ilustrar la importancia de los procesadores de lenguajes en las diversas areas de la Ingeniería de Telecomunicación basta citar algunos ejemplos muy relevantes, tales como: diseño de hardware y software (VHDL, SDL, ADA, C, JAVA, etc…), diseño de algoritmos de procesado de señales (MATLAB, Compiladores para procesadores DSP, etc…), ingeniería telemática (HTML, XML, JAVA Virtual Machine, etc…).

Metodología

- En las clases de teoría se explican los conceptos fundamentales de programación y su disponibilidad en varios lenguajes de programación modernos.
- En las clases de teoría también se proponen ejercicios para facilitar la comprensión de dichos conceptos.
- En el laboratorio se realiza un ejercicio práctico diseñado para familiarizar al alumno con los detalles internos de la implementación de procesadores de lenguaje.
- En internet se dispone de material de la asignatura accesible a traves de la siguiente dirección http://www.iuma.ulpgc.es/users/jmiranda, y también en el servidor de Campus Virtual de la ULPGC disponible en http://www.ulpgc.es

Criterios de Evaluación

Actividades que liberan materia:
- Las prácticas liberan hasta un máximo del 40% de la nota final. La evaluación de las prácticas se realizará en el laboratorio de la asignatura de forma interactiva junto al profesor (no es una prueba escrita).
- La presentación y defensa de un trabajo teórico consistente en la exposición y defensa de un artículo técnico libera hasta un máximo del 20% de la nota final.

Actividades que no liberan materia materia:
- La realización de los ejercicios propuestos en clase

Consideraciones generales:
- En las convocatorias oficiales de la asignatura el alumno deberá realizar un examen para obtener el 40% de la nota final asociada a la parte teórica. También puede realizar un segundo examen escrito para obtener el 40% asociado a la parte práctica, así como defender su trabajo teórico para obtener el 20% restante.

La nota final de la asignatura se obtiene mediante la suma de la nota de teoría y la de prácticas. Por tanto, no es necesario aprobar las tres partes para aprobar la asignatura.

Descripción de las Prácticas

Las prácticas se desarrollan en el Laboratorio de Arquitecturas del Departamento de Ingeniería Telemática.

Al comenzar el curso se realiza una práctica introductoria que no puntua. Su objetivo es familiarizar al alumno con los conceptos que va a utilizar a lo largo del curso. Para esta práctica se utilizan herramientas basadas en lex y yacc.

A lo largo del curso se realiza una única práctica consistente en el desarrollo de un pequeño lenguaje y su correspondiente traductor. El lenguaje (diseñado por los alumnos) debe tener un soporte mínimo para declaraciones de variables, gestión de expresiones aritméticas y lógicas, registros, arrays y procedimientos anidados. Además los alumnos pueden incorporar en su lenguaje características de programación orientada a objectos, excepciones, etc. El objetivo de la práctica es aplicar los aspectos fundamentales de diseño y desarrollo de procesadores de lenguajes.

Las prácticas se realizan en grupos de dos personas.

La duración estimada de la práctica: 15 horas.

Bibliografía

[1 Básico] Compilers: principles, techniques and tools /

Alfred V. Aho ...[et al.].
Addison Wesley,, Boston : (2007) - (2nd ed.)
0321491696 (ed. int.)

[2 Básico] Compiladores: principios, técnicas y herramientas /

Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman.
Addison-Wesley Iberoamericana,, Argentina : (1990)
0201629038

[3 Básico] Concepts of programming languages /
Robert W. Sebesta.
Addison-Wesley,, Boston (Massachusetts) : (2002) - (5th ed.)
0201752956

[4 Recomendado] Programming language pragmatics /
Michael L. Scott.
Morgan Kaufmann,, San Francisco : (2000)
1-55860-578-9

[5 Recomendado] Programming languages: concepts and constructs /
Ravi Sethi.
Addison-Wesley,, Reading (Massachusetts) : (1996) - (2nd ed.)
0-201-59065-4

[6 Recomendado] Programming languages: design and implementation /
Terrence W. Pratt, Marvin V. Zelkowitz.
Prentice Hall,, Upper Saddle River, New Jersey : (2000) - (4th. ed.)
0-13-027678-2

[7 Recomendado] Comparative programming languages.
Wilson, Leslie B.
Addison-Wesley,, Wokingham, England : (1993) - (2nd ed.)
0201568853

Equipo Docente

FRANCISCO JAVIER MIRANDA GONZÁLEZ *(COORDINADOR)*
Categoría: TITULAR DE UNIVERSIDAD
Departamento: INGENIERÍA TELEMÁTICA
Teléfono: 928451240 **Correo Electrónico:** javier.miranda@ulpgc.es
WEB Personal: http://www.iuma.ulpgc.es/users/jmiranda

Resumen en Inglés

Descriptor

Languages design: concepts and paradigms: values, storage, declarations, scopes, types, abstraction. Imperative programming, concurrent programming, object-oriented programming. Language processors: translators and interpreters: scanner, parser, semantic analyzer and code generator.

Requirements

The student should have passed the subjects Fundamentos de Programacion and Programacion of the first course of the ETSIT. It is also recommended to have passed most of the subjects of the third course of the ETSIT, and be familiar with the use of some variant of the UNIX Operating System (ie. Linux) at the user level.

Methodology:

- The instructor will present the theoretical concepts in class.
- The instructor will propose exercises to facilitate understanding such concepts.
- The students will write a language processor in the laboratory to understand the implementation

of these concepts.
- The students will have further information in the web at http://www.iuma.ulpgc.es/users/jmiranda, and also in the Virtual Campus Server of this University (available at http://www.ulpgc.es).

Evaluation:
- The defense of a technical report values 20% of the final punctuation.
- The work done in the laboratory values 40% of the final punctuation.
- The final exam values 40% of the final punctuation.
The final punctuation consists of the addition of these three values. Hence, it is not required to pass the three parts to pass the whole subject.

Laboratory Assignments

We well work in the Architectures Laboratory of the Departamento de Ingenieria Telematica. The goal of this practical work is to know implementation details of language processors. In groups composed of two students, they will work in the development of a language processor. The estimated length of this work is 15 hours.