Curso 2011-2012
UNIVERSIDAD DE ZARAGOZA
Vicegerencia Académica - Primer y Segundo Ciclo

Curso 2011-2012

CENTROS

Planes Estudio

PLANES

ASIGNATURAS

v. 2.11

Metodología de la programación CÓDIGO:12016

Ingeniero en Informática (en extinción)
Escuela de Ingeniería y Arquitectura, Zaragoza

Departamentos:
Informática e Ingeniería de Sistemas

Áreas:
Lenguajes y Sistemas Informáticos

Curso: 1
Duración: 2º cuatrimestre
Carácter: Troncal
Tipo: Teórica Práctica
Idioma: Español

Horas teóricas: 4
Horas prácticas: 15
Créditos UZ: 7,5
Créditos ECTS: 6,4

Oferta de plazas de libre elección:
Propia Titulación: No
Otras Titulaciones: No
y/u:
Otros Centros: No
Nº Plazas optativas:

Objetivos y Programa | Profesores y Bibliografía | Horario / Observaciones

Objetivos

El alumno debe aprender a especificar formalmente algoritmos (procedimientos y funciones), a diseñar algoritmos iterativos y recursivos correctos, a probar de forma rigurosa la corrección de algoritmos iterativos y recursivos y a analizar el coste de algoritmos iterativos y recursivos.

Estos objetivos generales se concretan en los siguientes objetivos específicos en términos de las tareas que debe aprender a realizar el alumno que la curse.
1. Escribir predicados lógicos que constituyan aserciones de un algoritmo.
2. Especificar procedimientos y funciones mediante una precondición y una postcondición formuladas como predicados lógicos.
3. Diseñar algoritmos iterativos correctos e identificar invariantes de sus bucles que permitan deducir su corrección.
4. Analizar el coste de un algoritmo iterativo calculando analíticamente el orden de su tiempo de ejecución.
5. Diseñar algoritmos recursivos correctos.
6. Transformar un algoritmo recursivo lineal con objeto de que lograr recursividad final y postcondición constante.
7. Transformar un algoritmo recursivo en iterativo.
8. Analizar el coste de un algoritmo recursivo calculando analíticamente el orden de su tiempo de ejecución mediante el planteamiento y la resolución de ecuaciones en recurrencias.

Además, en las prácticas de la asignatura realizadas en el laboratorio de programación el alumno ampliará sus conocimientos sobre el lenguaje de programación Ada, ganará experiencia en su utilización, se ejercitará en la búsqueda y consulta de la información necesaria para resolver los problemas de programación que se le planteen y continuará trabajando en equipo y de forma cooperativa.

Programa

I. Especificación de algoritmos
1. Introducción a la especificación formal de algoritmos
2. Especificación formal de algoritmos que trabajan con registros y tablas
3. Especificación formal de algoritmos que trabajan con ficheros secuenciales

II. Coste de un algoritmo iterativo
4. Introducción al análisis de la eficiencia de un algoritmo
5. Notaciones asintóticas para expresar la eficiencia de algoritmos
6. Cálculo de la complejidad de algoritmos.

III. Corrección de un algoritmo iterativo
7. Bases de la verificación formal de algoritmos
8. Análisis de la corrección de un algoritmo sin bucles
9. Análisis de la corrección de un algoritmo con bucles
10. Análisis de la corrección de un algoritmo con invocaciones a procedimientos y funciones
11. Diseño de algoritmos con bucles

IV. Diseño recursivo de algoritmos
12. Introducción a la recurrencia
13. Fundamentos del diseño recursivo
14. Aplicación al diseño de programas recursivos.
15. Coste de un algoritmo recursivo
16. Diseño recursivo de algoritmos por inmersión
17. Diseño recursivo de algoritmos por inmersión de especificaciones
18. Obtención de algoritmos recursivos finales
19. Transformación de recursivo a iterativo: caso de recursividad lineal
20. Transformación de recursivo a iterativo: caso de recursividad múltiple

V. Aplicación al diseño de algoritmos
21. Algoritmos de búsqueda
22. Algoritmos de ordenación

PROGRAMA DE PRÁCTICAS

Comprenden tres trabajos de programación en Ada que los alumnos realizarán formando grupos de tres. Cada una de las prácticas tiene los contenidos que a continuación de detallan:

1ª. Programación modular. Medida y cálculo experimental del coste de ejecución de algoritmos (2 sesiones)
2ª. Especificación y anotación de programas. Diseño modular de programas iterativos. Programación de excepciones. (2

sesiones)
3ª. Diseño recursivo de programas. (1 sesión)

Para realizar cada práctica se dispone de una o más sesiones de laboratorio de tres horas en las que cada grupo podrá desarrollar su trabajo de programación y consultar con el profesor los problemas que en él se presentan.

Evaluación

Examen escrito teórico-práctico (75%) y los trabajos de programación asociados a cada una de las prácticas de la asignatura (25%). Estos trabajos de programación se evaluarán en base a los materiales escritos y programas que constituyen el resultado de cada práctica y a una entrevista de los alumnos de cada grupo con su profesor sobre los trabajos realizados. Para aprobar la asignatura es necesario obtener una calificación mínima de 3.0 en la evaluación de las prácticas de la asignatura y lograr una nota global de un 5.0 entre el examen escrito y las prácticas. Las calificaciones obtenidas en la primera convocatoria en cualquiera de las dos partes que constituyen la evaluación se conservan para la convocatoria de septiembre.

universidad de zaragoza ©2006