



PROGRAMA DE LA ASIGNATURA

CURSO ACADÉMICO 2009/2010

ASIGNATURA: Fundamentos de Programación II	PLAN DE ESTUDIOS: 2000/2002
CODIGO: 113	TITULACIÓN: Ingeniería en Informática Ingeniería Técnica en Informática
CRÉDITOS: 4,5	CARÁCTER: Cuatrimestral (2º cuatrimestre)
CURSO: 1º	TIPO : Obligatoria
IDIOMA EN QUE SE IMPARTE: Español	
PROFESORES DE LA ASIGNATURA:	
María Dorrego Luxán	maria.dorrego@upsam.net
Luis Rodríguez Baena	luis.rodriguez@upsam.net

I. OBJETIVOS

- Profundizar en el estudio de los elementos básicos para la programación de sistemas informáticos.
- Aprender el manejo de las estructuras avanzadas de datos.
- Comprender el funcionamiento de los tipos de datos dinámicos.
- Aprender los conceptos básicos de la programación orientada a objetos.

II. PRERREQUISITOS

Conocimientos básicos sobre diseño de algoritmos y estructuras de datos impartidos en la asignatura de Fundamentos de la Programación I

III. CONTENIDO

TEMA 1. ARCHIVOS

- 1.1. Concepto de archivo
 - 1.1.1. El tipo de dato archivo
 - 1.1.2. Estructura física y estructura lógica
- 1.2. Soportes, organización y tipos de acceso
 - 1.2.1. Tipos de soportes
 - 1.2.2. Tipo de acceso
 - 1.2.3. Organización secuencial
 - 1.2.4. Organización directa
 - 1.2.5. Organización indexada
 - 1.2.6. Ventajas e inconvenientes
- 1.3. Instrucciones para el manejo de archivos
 - 1.3.1. Declaración de datos de tipo archivo
 - 1.3.2. Creación, apertura y cierre del archivo
 - 1.3.3. Escritura
 - 1.3.4. Lectura
- 1.4. Operaciones con archivos secuenciales
 - 1.4.1. Recorrido
 - 1.4.2. Mezcla y actualización
 - 1.4.3. Rupturas de control
 - 1.4.4. Ordenación de archivos secuenciales
- 1.5. Operaciones con archivos directos e indexados

TEMA 2. RECURSIVIDAD

- 2.1. Naturaleza de la recursividad
 - 2.1.1. Partes de un algoritmo recursivo
 - 2.1.2. Tipos de recursividad
 - 2.1.3. Llamadas a módulos recursivos
- 2.2. Procedimientos y funciones recursivas
 - 2.2.1. Ventajas e inconvenientes de la recursividad
- 2.3. Resolución de problemas recursivos
 - 2.3.1. QuickSort
 - 2.3.2. La torres de Hanoi



TEMA 3. ESTRUCTURAS LINEALES DE DATOS: PILAS, COLAS, LISTAS ENLAZADAS

- 3.1. Introducción a los tipos abstractos de datos
- 3.2. Datos dinámicos y estructuras de datos dinámicas
 - 3.2.1. Trabajo con punteros
 - 3.2.2. Estructuras de datos dinámicas
- 3.3. Estructuras de datos lineales y no lineales
 - 3.3.1. Listas
 - Listas contiguas*
 - Listas enlazadas*
- 3.4. Pilas
 - 3.4.1. Operaciones primitivas
 - 3.4.2. Implementaciones de pilas
- 3.5. Colas
 - 3.5.1. Operaciones primitivas
 - 3.5.2. Implementaciones de colas
- 3.6. Listas enlazadas simples
 - 3.6.1. Operaciones primitivas
 - 3.6.2. Implementaciones de listas enlazadas
 - Listas ordenadas*
- 3.7. Otros tipos de listas: listas circulares, listas dobles, listas con cabecera
 - 3.7.1. Listas circulares
 - 3.7.2. Listas doblemente enlazadas
 - 3.7.3. Listas con cabecera
 - 3.7.4. Otras realizaciones

TEMA 4. ESTRUCTURAS NO LINEALES DE DATOS: ÁRBOLES

- 4.3. Estructuras de datos no lineales
 - 4.1.1. Árboles
 - 4.1.2. Terminología de árboles
 - 4.1.3. Árboles binarios
 - Terminología de árboles binarios*
 - Implementación de árboles*
- 4.3. Recorridos en un árbol binario
 - 4.3.1. Recorridos en profundidad
 - Implementación de los recorridos en profundidad*
 - 4.3.2. Recorridos en anchura
- 4.4. Árboles binarios de búsqueda
 - 4.4.1. Búsqueda en un árbol binario de búsqueda
 - 4.4.2. Inserción en un árbol binario de búsqueda
 - 4.4.3. Borrado de un nodo en un árbol binario de búsqueda

TEMA 5. INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS

- 5.1. Programación procedimental y POO
- 5.2. Objetos y clases
 - 5.2.1. Miembros de una clase
 - Mensajes*
- 5.3. Declaración de clases
- 5.4. Declaración de objetos
 - 5.4.1. Objetos e instancias
 - 5.4.2. Constructores
 - Sobrecarga*
 - Destruyores*
- 5.5. Visibilidad de los miembros de una clase
 - 5.5.1. Modificadores de visibilidad
 - 5.5.2. Métodos ayudantes
- 5.6. Representación de clases mediante UML
- 5.7. Agregación



- 5.8. Jerarquía de clases: especialización / generalización
 - 5.8.1. Herencia
 - 5.8.2. Implementación de la herencia
 - 5.8.3. Clases y métodos abstractos
 - 5.8.4. Polimorfismo

IV. BIBLIOGRAFÍA

A. Básica

Teoría:

JOYANES, L.: *Fundamentos de la programación. Algoritmos y Estructura de Datos*, 4ª Edición. Madrid: McGraw-Hill, 2008.

Problemas:

JOYANES, L.; RODRIGUEZ, L; FERNANDEZ, M.: *Fundamentos de programación Libro de problemas*. 2ª Edición. Madrid: McGraw-Hill, 2003.

B. Bibliografía complementaria.

AHO, Alfred V.; HOPCROFT, John E.; ULLMAN, Jeffrey D.: *Estructuras de datos y algoritmos*. México: Addison Wesley, 1998.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I.: *El lenguaje unificado de modelado*. Madrid: Addison-Wesley, 2000

BOOCH, G: *Análisis y diseño orientado a objetos con aplicaciones*. Madrid: Addison-Weasley, 1995.

BRASSARD, G.; BRATLEY, P.: *Fundamentos de Algoritmia*. Madrid: Prentice-Hall, 1997.

COLLADO MACHUCA, M.; MORALES FERNÁNDEZ, R.; MORENO NAVARRO, J. J.: *Estructuras de datos. Realización en Pascal*. Madrid: Ediciones Díaz de Santos, 1987.

EGE, Raimund K.: *Programming in a Object-Oriented Environment*. San Diego (CA): Academic Press, 1992.

GARCÍA MOLINA, J. J.; MONTOYA DATO, F. J.; FERNÁNDEZ ALEMÁN, J. L.; MAJADO ROSALES, M. J.: *Una introducción a la programación. Un enfoque algorítmico*. Madrid: Thomson-Paraninfo, 2005.

JOYANES, Luis: *Programación orientada a objetos*. 2ª edición. Madrid: McGraw-Hill, 1998.

JOYANES, Luis.: *Problemas de Metodología de la Programación*. Madrid: McGraw-Hill, 1990.

JOYANES, L.: *Fundamentos de la programación. Algoritmos y Estructura de Datos*. Madrid: McGraw-Hill, 1988.

RUMBAUGH, J.; BLAHA, M; PREMERLANI, W. L.; LORENSEN, W.: *Modelado y diseño orientados a objetos (Metodología OMT)*. Madrid: Prentice-Hall, 1998.

WIRTH. N.: *Algoritmos + Estructuras de datos = programas*. Madrid: Ediciones del Castillo S. A., 1986.

V. MÉTODOS DOCENTES

La asignatura se impartirá mediante sesiones en el aula de teoría. Las sesiones se compondrán de explicaciones teóricas de los conceptos de la asignatura acompañadas del desarrollo práctico de algoritmos tanto resueltos por el profesor como realizados por los alumnos.



VI. EVALUACIÓN

Además del examen de la asignatura en la convocatoria ordinaria de junio, los alumnos y alumnas tendrán la oportunidad de evaluarse mediante una evaluación continua. En este último caso, la nota final de la asignatura se calculará mediante la evaluación del trabajo realizado por el alumno durante el periodo lectivo de la asignatura y la obtenida en el examen de la convocatoria de junio (*véanse* las Normas Académicas de la Asignatura en documento aparte). En caso de suspender la convocatoria ordinaria de junio, el alumno deberá realizar el examen correspondiente en la convocatoria de septiembre.

Modelo de examen

El examen constará de dos partes: una parte teórico-práctica con preguntas teóricas y pequeños ejercicios relacionados y una parte práctica en la que el alumno deberá resolver con un lenguaje algorítmico algún supuesto de programación. El peso para cada una de las partes sería de un 45-50% para las preguntas teórico-prácticas y de un 50-55% para el supuesto práctico.

En Madrid, 20 de mayo de 2009