Programa Oficial de Asignatura

Fundamentos de Programación

Ficha Técnica

Titulación:	Grado en Ingeniería de Tecnologías y Servicios de Telecomunicaciones		
Plan BOE:	BOE número 108 de 6 de mayo de 2015		
Asignatura:	Fundamentos de Programación		
Módulo:	Informática		
Curso:	1º	**Créditos ECTS:**	6
Tipo de asignatura:	Básica	**Tipo de formación:**	Teórica y Práctica

Presentación

En la asignatura de formación básica denominada “Fundamentos de Programación” se dan las bases necesarias para conocer las técnicas de programación modernas a través de lenguajes de alto nivel estructurados, orientación a objetos y desarrollo de aplicaciones orientadas a eventos. Además, se explicarán los diferentes enfoques y herramientas para afrontar la realización de programas informáticos.

La asignatura consta de cuatro partes diferenciadas. En la primera parte de la asignatura se exponen los conceptos de manera teórica. A través de pseudocódigo, se explicarán las estructuras básicas y reglas fundamentales para la realización de programas.

En la segunda parte de la asignatura se realizará la aplicación de los conceptos adquiridos en la primera parte mediante la utilización de un lenguaje de programación estructurada.

En la tercera parte de la asignatura se realizará una introducción a la metodología de programación orientada a objetos, y se estudiará de forma teórica y práctica los fundamentos del lenguaje Java para este paradigma de programación.

En la parte final de la asignatura se imparten nociones básicas de la programación orientada a eventos, y de la generación de interfaces gráficas de usuario en aplicaciones informáticas.

Como resultado del estudio de la asignatura se espera que el alumno sea capaz de realizar la implementación y diseño de programas informáticos independientemente de la plataforma, tipo de lenguaje o paradigma de programación que se vaya a utilizar, de manera clara y sencilla.

Antes de matricular la asignatura, verifique los posibles requisitos que pueda tener dentro de su plan. Esta información la encontrará en la pestaña "Plan de estudios" del plan correspondiente.

Competencias y/o resultados del aprendizaje

- CE2. Capacidad para adquirir y desarrollar los conocimientos básicos sobre el uso y programación de los ordenadores y programas informáticos con aplicación en ingeniería

Los resultados de aprendizaje que alcanzará el alumno con esta asignatura son:

- Conocer los conceptos básicos de la tecnología de la información y comunicación.
- Conocer los conceptos básicos de la programación utilizando lenguajes de alto nivel.
- Comprensión de tipos abstractos de datos así como su implementación en lenguajes de programación de alto nivel.
- Comprensión de los principales algoritmos así como de las técnicas necesarias para la estimación y

valoración de su complejidad y cálculo.

Contenidos Didácticos

1 Concepto de programa informático
- 1.1 Introducción a las computadoras
- 1.2 La programación
- 1.3 Concepto de programa
- 1.4 El pseudocódigo
  - 1.4.1 Reglas generales del pseudocódigo
- 1.5 Estructuras de control
  - 1.5.1 Secuencia
  - 1.5.2 Condiciones
- 1.6 Estructuras de control: repeticiones
  - 1.6.1 Estructura de tipo Mientras
  - 1.6.2 Estructura de tipo Hasta
  - 1.6.3 Estructura de tipo Para

2 Algoritmos y sistemas de representación de un programa
- 2.1 Algoritmo
  - 2.1.1 Características básicas de un algoritmo
  - 2.1.2 Algoritmo de Euclides
  - 2.1.3 Algoritmos computables y no computables
- 2.2 Diagramas de flujo y ordinogramas
  - 2.2.1 El diagrama de flujo
  - 2.2.2 El ordinograma
- 2.3 La programación estructurada
  - 2.3.1 Diagramas y programa propio
  - 2.3.2 Diagramas estructurados o diagramas privilegiados
  - 2.3.3 Programa estructurado
  - 2.3.4 Teoremas de la programación estructurada
- 2.4 Diagramas estructurados arborescentes
  - 2.4.1 Diagrama arborescente del máximo común divisor
- 2.5 Diagramas estructurados de la metodología Nassi-Schneiderman o de Chapin
  - 2.5.1 Diagrama N-S o de Chapin del máximo común divisor
  - 2.5.2 Ventajas de los diagramas estructurados con respecto a los ordinogramas clásicos

3 Tratamiento informático de un problema
- 3.1 Introducción
- 3.2 Definición de los requisitos del problema
- 3.3 Análisis
- 3.4 Diseño
  - 3.4.1 Diseño general
  - 3.4.2 Diseño detallado
- 3.5 Codificación
- 3.6 Pruebas
- 3.7 Mantenimiento
- 3.8 La programación
- 3.9 Paradigmas de programación
- 3.10 Paradigma imperativo
- 3.11 Los lenguajes de programación
- 3.12 El lenguaje imperativo C++

4 Expresiones y Sentencias Básicas

7.1 Concepto de metodología de programación
7.2 La metodología de desarrollo de programas por diseño descendente ("top-down")
7.2.1 Diseño descendente ("top-down"): descomposición de tareas o subtareas
7.2.2 Descripción
7.3 Principios y propiedades generales de un diseño descendente
7.3.1 Modularidad de un programa
7.3.2 Acoplamiento entre módulos
7.3.3 Número de aspectos a considerar en cada momento
7.3.4 Cohesión interna de un módulo
7.3.5 Complejidad de un módulo
7.3.6 Otros principios
7.4 Características básicas de "buen estilo" en programación
7.4.1 Indentación
7.4.2 Nombres de identificadores
7.4.3 Comentarios
7.5 La recursión
7.5.1 Concepto de recursión y función recursiva
7.5.2 Ejemplo de función factorial: implementación iterativa y recursiva
7.5.3 Recursividad bien construida. Funciones parciales y totales
7.5.4 Tipos de recursión: recursión final y no final
7.5.5 Cuándo no utilizar la recursión

8 La programación orientada a objetos
8.1 El modelo orientado a objetos
8.2 Conceptos fundamentales de la programación orientada a objetos
8.3 La abstracción de datos
8.4 Los objetos
8.4.1 Características que definen un objeto
8.4.2 Representación gráfica de un objeto
8.5 Las clases
8.5.1 Estructura de una clase
8.5.2 Representación gráfica de una clase
8.6 Las relaciones
8.6.1 Relación de asociación
8.6.2 Relación de agregación
8.6.3 Relación de generalización/especialización
8.7 Los mensajes
8.8 El poliformismo
8.9 La herencia
8.10 Estructura de una aplicación orientada a objetos
8.11 La reutilización y la extensibilidad
8.12 Metodología básica de desarrollo orientado a objetos
8.12.1 Captura de requisitos
8.12.2 Análisis
8.12.3 Diseño
8.12.4 Implementación
8.12.5 Pruebas

9 Lenguajes orientados a objetos, utilidades E/S y genéricos
9.1 Introducción a Java
9.2 El entorno de desarrollo Java
9.3 Mi primer programa
9.4 Sintaxis básica de Java
9.5 Expresiones y operadores

Contenidos Prácticos

Durante el desarrollo de la asignatura se realizarán las siguientes actividades prácticas:

- Ejercicios sobre conceptos de programación
- Ejercicios prácticos de programación imperativa
- Ejercicios prácticos de expresiones, entrada/salida y funciones
- Ejercicios prácticos de programación estructurada
- Debate/coloquio sobre un diseño orientado a objetos
- Realización de un sistema informático con programación orientada a objetos en Java
- Ejercicio de Programación Orientada a Eventos
- Resolución de un caso práctico con interfaces gráficas

Evaluación

El sistema de evaluación del aprendizaje de la UDIMA contempla la realización de diferentes tipos de actividades de evaluación y aprendizaje. El criterio de valoración establecido se detalla a continuación:

Actividades de aprendizaje	10%
Controles	10%
Actividades de Evaluación Continua (AEC)	20%
Examen final presencial	60%
TOTAL	**100%**

Bibliografía

- *Fundamentos de Programación* (2015). Madrid: Ed. CEF
- *Metodología de la Programación* (2015). Madrid: Ed. CEF