

### Ficha Técnica

<b>Titulación:</b>	Grado en Ingeniería Informática		
<b>Plan BOE:</b>	BOE número 108 de 6 de mayo de 2015		
<b>Asignatura:</b>	Metodología de la Programación		
<b>Módulo:</b>	Programación y Estructura de Datos		
<b>Curso:</b>	1º	<b>Créditos ECTS:</b>	6
<b>Tipo de asignatura:</b>	Básica	<b>Tipo de formación:</b>	Teórica y Práctica

### Presentación

En la asignatura de formación básica denominada “Metodología de la Programación” se da la formación necesaria para que el alumno pueda desarrollar programas informáticos de tamaño medio. Esta asignatura constituye la continuación natural de la asignatura “Fundamentos de Programación”, pero incorporando el uso de recursividad y el paradigma de programación orientada a objetos.

Al principio de la asignatura, se obtienen los conocimientos teóricos relacionados con la recursión y de la programación orientada a objetos. Dentro de la programación orientada a objetos, se explican las características básicas tales como herencia, polimorfismo, y reutilización; y se enseña al alumno a diseñar sus propias aplicaciones informáticas. El alumno aprenderá estos conceptos de forma práctica con el lenguaje de programación Java, elegido por su versatilidad entre diferentes plataformas y su éxito como lenguaje orientado a objetos.

Finalmente, el alumno aprende a desarrollar aplicaciones informáticas incorporando gestión de entrada/salida, gestión de eventos e interfaces gráficas de usuario.

### Competencias y/o resultados del aprendizaje

- Realizar programas informáticos.
- Saber implementar las distintas estructuras de datos y sus operaciones.
- Saber escoger la estructura de datos más adecuada para resolver un problema, de acuerdo con criterios de eficiencia temporal y espacial.
- Diseñar y estructurar los programas informáticos de forma modular y robusta usando la programación orientada a objetos.
- Argumentar las ventajas e inconvenientes de los diseños de aplicaciones informáticas.

### Contenidos Didácticos

- 1 Metodología básica de desarrollo de programas:
  - 1.1 Concepto de metodología de programación
  - 1.2 La metodología de desarrollo de programas por diseño descendente ("top-down")
    - 1.2.1 Diseño descendente ("top-down"): descomposición de tareas o subtareas
  - 1.3 Principios y propiedades generales de un diseño descendente
    - 1.3.1 Modularidad de un programa
    - 1.3.2 Acoplamiento entre módulos
    - 1.3.3 Número de aspectos a considerar en cada momento
    - 1.3.4 Cohesión interna de un módulo
    - 1.3.5 Complejidad de un módulo

- 1.3.6 Otros principios
- 1.4 Características básicas de "buen estilo" en programación
  - 1.4.1 Indentación
  - 1.4.2 Nombres de identificadores
  - 1.4.3 Comentarios
- 1.5 La recursión
  - 1.5.1 Concepto de recursión y función recursiva
  - 1.5.2 Ejemplo de función factorial: implementación iterativa y recursiva
  - 1.5.3 Recursividad bien construida. Funciones parciales y totales
  - 1.5.4 Tipos de recursión: recursión final y no final
  - 1.5.5 Cuándo no utilizar la recursión
- 2 Unidad didáctica 2. La programación orientada a objetos (I):
  - 2.1 El modelo orientado a objetos
  - 2.2 Conceptos fundamentales de la programación orientada a objetos
  - 2.3 La abstracción de datos
  - 2.4 Los objetos
    - 2.4.1 Características que definen un objeto
    - 2.4.2 Representación gráfica de un objeto
  - 2.5 Las clases
    - 2.5.1 Estructura de una clase
    - 2.5.2 Representación gráfica de una clase
  - 2.6 Las relaciones
    - 2.6.1 Relación de asociación
    - 2.6.2 Relación de agregación
    - 2.6.3 Relación de generalización/especialización
- 3 Unidad didáctica 3. La programación orientada a objetos (II):
  - 3.1 Los mensajes
  - 3.2 El polimorfismo
    - 3.2.1 Sobrecarga
    - 3.2.2 Polimorfismo paramétrico o genericidad
    - 3.2.3 Polimorfismo dinámico
  - 3.3 La herencia
    - 3.3.1 Herencia y ocultación de información
    - 3.3.2 Herencia y tipado
    - 3.3.3 Herencia múltiple
  - 3.4 Estructura de una aplicación orientada a objetos
  - 3.5 La reutilización y la extensibilidad
  - 3.6 Metodología básica de desarrollo orientado a objetos
    - 3.6.1 Captura de requisitos
    - 3.6.2 Análisis
    - 3.6.3 Diseño
    - 3.6.4 Implementación
    - 3.6.5 Pruebas
- 4 Unidad didáctica 4. El lenguaje Java:
  - 4.1 Introducción a Java
    - 4.1.1 ¿Qué es Java?
    - 4.1.2 Tipos de aplicaciones en Java
  - 4.2 El entorno de desarrollo de Java
  - 4.3 Mi primer programa

- 4.3.1 Compilar en consola
  - 4.3.2 Utilizar un IDE
  - 4.4 Sintaxis básica de Java
    - 4.4.1 El método main
    - 4.4.2 Componentes básicos
    - 4.4.3 Variables
    - 4.4.4 Arrays
  - 4.5 Expresiones y operadores
    - 4.5.1 Operador de asignación
    - 4.5.2 Operadores sobre tipos básicos
    - 4.5.3 Operadores de Strings
    - 4.5.4 Operador de selección
    - 4.5.5 Operador de comparación de tipo (instanceof)
    - 4.5.6 Precedencia de operadores
  - 4.6 Sentencias y bloques
  - 4.7 Estructuras de control de flujo
    - 4.7.1 Sentencias condicionales
    - 4.7.2 Bucles
    - 4.7.3 Sentencia return
  - 4.8 Conversión de tipos (casting)
- 5 Unidad didáctica 5. Programación orientada a objetos con Java:
- 5.1 Clases
    - 5.1.1 Declaración de clases
    - 5.1.2 Declaración de atributos
    - 5.1.3 Declaración de métodos
    - 5.1.4 Creación de instancias
    - 5.1.5 La palabra reservada this
    - 5.1.6 Constructores y destructores de clase
    - 5.1.7 Atributos y métodos de clase
  - 5.2 Paquetes
  - 5.3 Metodología del main
  - 5.4 Objetos y referencias en Java
  - 5.5 Herencia
    - 5.5.1 Herencia de implementación
    - 5.5.2 Herencia de interfaz
- 6 Unidad didáctica 6. Utilidades y Entrada/Salida:
- 6.1 Enumerados
  - 6.2 La documentación de la API de Java
  - 6.3 Utilidades
    - 6.3.1 Manejo de fechas
    - 6.3.2 Análisis de cadenas de texto
    - 6.3.3 Generación de números aleatorios
  - 6.4 Excepciones
    - 6.4.1 Tipos de errores en Java
    - 6.4.2 Manejo de excepciones
    - 6.4.3 Creación de excepciones particulares
  - 6.5 Entrada/Salida
    - 6.5.1 La clase abstracta InputStream
    - 6.5.2 La clase abstracta Reader
    - 6.5.3 La clase abstracta OutputStream

- 6.5.4 La clase abstracta Writer
- 6.5.5 Object Streams
  
- 7 Unidad didáctica 7. Genéricos en Java:
  - 7.1 Introducción a los genéricos en Java
    - 7.1.1 Definición de clases genéricas
    - 7.1.2 Uso de clases genéricas
  - 7.2 Métodos y constructores genéricos
  - 7.3 Restricción de parámetros de tipo
    - 7.3.1 Restricción de argumentos genéricos
  - 7.4 Borrado de tipos
  - 7.5 Colecciones en Java
    - 7.5.1 Recorrer colecciones
    - 7.5.2 Listas
    - 7.5.3 Conjuntos
    - 7.5.4 Colas
    - 7.5.5 Mapas
  
- 8 Unidad didáctica 8. La programación orientada a eventos:
  - 8.1 Modelos de programación
  - 8.2 La programación dirigida por el control
  - 8.3 La programación dirigida por los datos
  - 8.4 La programación dirigida por los eventos
    - 8.4.1 ¿Qué es un evento?
    - 8.4.2 ¿Quién puede generar un evento?
    - 8.4.3 Características de la programación dirigida por eventos
  
- 9 Unidad didáctica 9. La Programación orientada a eventos en Java:
  - 9.1 Modelos de programación
  - 9.2 La programación dirigida por el control
  - 9.3 La programación dirigida por los datos
  - 9.4 La programación dirigida por los eventos
    - 9.4.1 ¿Qué es un evento?
    - 9.4.2 ¿Quién puede generar un evento?
    - 9.4.3 Características de la programación dirigida por eventos
  
- 10 Unidad didáctica 10. Interfaces gráficas de usuario:
  - 10.1 Introducción a AWT
  - 10.2 Implementación de interfaces gráficas
  - 10.3 Componentes básicos de AWT
    - 10.3.1 Botones
    - 10.3.2 Listas desplegables
    - 10.3.3 Etiquetas
    - 10.3.4 Campos de texto
    - 10.3.5 Áreas de texto
  - 10.4 Contenedores y distribución de componentes visuales
    - 10.4.1 La clase de FlowLayout
    - 10.4.2 La clase GridLayout
    - 10.4.3 La clase BorderLayout
  - 10.5 Eventos
  - 10.6 Swing

## Contenidos Prácticos

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

- Realización de diseños de jerarquía de clases y estructuras de datos según los requisitos disponibles.
- Desarrollo de software con programación orientada a objetos.
- Desarrollo de software con programación orientada a eventos.
- Desarrollo de interfaces gráficos sencillos.

## 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)	30%
Examen final presencial	50%
<b>TOTAL</b>	<b>100%</b>

## Bibliografía

- Fernando Alonso Amo, José Crespo del Arco, Ángel Lucas González Martínez, Miguel Jiménez Gañán y Daniel Manrique Gamo. (2009) "Metodología de la Programación". Editorial: UDIMA.
- Bruce Eckel. 2002. *Thinking in Java* (3rd ed.). Prentice Hall Professional Technical Reference.
- Bruce Eckel, Chuck D. Allison, and Chuck Allison. 2003. *Thinking in C++, Vol. 2* (2 ed.). Pearson Education.
- Flórez Fernández, H. A. (2012). Programación orientada a objetos usando java. Colombia: Ecoe Ediciones.
- Casanova, A., & Marqués, F. (2012). Empezar a programar usando java. España: Editorial de la Universidad Politécnica de Valencia
- Ruiz, R. R. (2014). Fundamentos de la programación orientada a objetos: una aplicación a las estructuras de datos en Java. Argentina: El Cid Editor. Retrieved from <http://www.ebrary.com>
- Pilgrim, P. A. (2013). Java EE 7 Developer Handbook. Olton, Birmingham, GBR: Packt Publishing