

Guía docente

230010 - MOO - Metodología y Programación Orientada a Objetos

Última modificación: 07/07/2015

Unidad responsable: Escuela Técnica Superior de Ingeniería de Telecomunicación de Barcelona

Unidad que imparte: 701 - DAC - Departamento de Arquitectura de Computadores.

Titulación: **Curso:** 2015

Créditos ECTS: 6.0

Idiomas: Catalán, Castellano

PROFESORADO

Profesorado responsable:

Cruellas Ibarz, Juan Carlos

Otros:

Cruellas Ibarz, Joan Carles

Bofill Soliguer, Pablo

Fornes De Juan, Jorge

Rodriguez Luna, Eva

Perello Muntan, Jordi

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Transversales:

1. COMUNICACIÓN EFICAZ ORAL Y ESCRITA - Nivel 1: Planificar la comunicación oral, responder de manera adecuada a las cuestiones formuladas y redactar textos de nivel básico con corrección ortográfica y gramatical.
2. USO SOLVENTE DE LOS RECURSOS DE INFORMACIÓN - Nivel 1: Identificar las propias necesidades de información y utilizar las colecciones, los espacios y los servicios disponibles para diseñar y ejecutar búsquedas simples adecuadas al ámbito temático.

METODOLOGÍAS DOCENTES

- Clases expositivas
- Clases de laboratorio
- Trabajo en grupo (no presencial)
- Trabajo individual (no presencial)
- Pruebas de respuesta corta (Control)
- Pruebas de respuesta larga (Examen Final)
- Práctica de laboratorio
- Prueba de laboratorio

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

Resultado del aprendizaje:

Entender en qué consiste el paradigma de la programación orientada a objetos y sepa utilizar los aspectos más relevantes en el diseño y el desarrollo de software.

Disponer de herramientas metodológicas para el desarrollo de software orientado, y que sepa utilizar correctamente.

Conocer y utilizar los fundamentos básicos de la arquitectura y metodología de diseño de software.

Planificar y lleva a cabo una presentación oral, responde de manera adecuada a las cuestiones formuladas y redacta correctamente textos de nivel básico.

Identificar los objetivos del grupo y puede diseñar un plan de trabajo para alcanzarlos. Identifica las responsabilidades de cada componente del grupo y asume el compromiso de la tarea asignada.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTE

| Tipo | Horas | Porcentaje |
|----------------------------|-------|------------|
| Horas grupo pequeño | 26,0 | 17.33 |
| Horas aprendizaje autónomo | 85,0 | 56.67 |
| Horas grupo grande | 39,0 | 26.00 |

Dedicación total: 150 h

CONTENIDOS

Tema 1. El paradigma de l'orientació a objectes: classes i objectes.

Descripción:

Presentación del paradigma de la orientación a objetos. Conceptos de objeto y clase. Atributos y métodos.

Objetivos específicos:

Formalizar el concepto de objeto como entidad que aglutina datos (atributos) y funciones (métodos). Formalizar el concepto de clase.

Dedicación: 3h 30m

Grupo grande/Teoría: 1h 30m

Aprendizaje autónomo: 2h

Tema 2. Clases y objetos en Java

Descripción:

Sintaxis de definición de clases en Java.

Definición de atributos. Modificadores de acceso: public, private.

Definición de métodos. Métodos especiales: constructores, métodos de acceso a atributos (set y get).

Creación de objetos: operador new. Gestión dinámica de memoria

Atributos y métodos estáticos. Su razón de ser y su uso.

Modificadores constant y final.

Dedicación: 21h

Grupo grande/Teoría: 6h

Aprendizaje autónomo: 15h

Tema 3. Tipos primitivos en Java y clases relevantes

Descripción:

Tipos primitivos en Java: tipos de enteros, tipos de reales, tipo caracter (diferencia con respecto a C), booleano, byte.

Clases correspondientes a los tipos primitivos: Integer, Double, Float, Character, Boolean, Byte.

Clase String. Concepto y métodos más relevantes.

De valores numéricos a su representación textual como Strings y viceversa.

Dedicación: 6h 30m

Grupo grande/Teoría: 2h

Aprendizaje autónomo: 4h 30m

Tema 4. Contenedores

Descripción:

Concepto de objeto contenedor.

Tipos de contenedores: secuencias, diccionarios y conjuntos.

Implementación de secuencias en Java: ArrayList y LinkedList. Métodos más relevantes.

Iteradores. Concepto y su papel en el recorrido de secuencias.

Implementación de diccionarios en Java: HashMap. Métodos más relevantes.

Obtención de vistas parciales de un diccionario: métodos keySet(), values()

Implementación de conjuntos en Java: HashSet. Métodos más relevantes.

Identificación del tipo de contenedor que es preciso utilizar en cada caso.

Dedicación: 21h

Grupo grande/Teoría: 6h

Aprendizaje autónomo: 15h

Tema 5. Introducción al UML. Diagrama de clases.

Descripción:

El UML como un lenguaje de representación gráfica de un programa.

El diagrama de clases.

Relaciones entre clases: asociaciones y dependencias.

Definición completa de una asociación: nombre, direccionalidad, cardinalidad.

Implementación de las asociaciones en el código Java.

Dedicación: 11h 20m

Grupo grande/Teoría: 3h

Aprendizaje autónomo: 8h 20m

Tema 6. Herencia y polimorfismo

Descripción:

Concepto de herencia. Superclases y clases derivadas (subclases). Herencia de atributos y métodos. Constructores en las subclases.

La herencia como relación entre clases. Representación de la herencia en diagramas de clase. Los objetos instancia de una subclase son también objetos instancia de la superclase.

La clase Object como clase raíz de la jerarquía de clases en Java. El método toString() de Object.

Concepto de polimorfismo. Métodos polimórficos.

El Interface en Java. El interface como simulador de herencia múltiple. El interface como mecanismo que permite a un mismo objeto presentar varios tipos de comportamiento.

Dedicación: 21h 20m

Grupo grande/Teoría: 7h

Aprendizaje autónomo: 14h 20m

Tema 7. Excepciones

Descripción:

El concepto de excepción. Generación, lanzamiento y recogida de excepciones en Java.

Tratamiento de excepciones.

Dedicación: 11h 10m

Grupo grande/Teoría: 4h 30m

Aprendizaje autónomo: 6h 40m



Tema 8. Introducción al proceso de desarrollo de software (1): Análisis orientado a objetos

Descripción:

Breve descripción de proceso iterativo de desarrollo de software.
El análisis orientado a objetos: actividades relevantes y resultados.
Concepto de casos de uso. Tipos, identificación y definición.
Modelo conceptual. Concepto y técnicas de construcción a partir de los casos de uso.

Dedicación: 13h 30m

Grupo grande/Teoría: 4h 30m

Aprendizaje autónomo: 9h

Tema 9. Introducción al proceso de desarrollo de software (2): Diseño orientado a objetos

Descripción:

El diseño orientado a objetos: actividades relevantes y resultados.
Diagrama de secuencia UML.
Asignación de responsabilidades a las clases. Patrones GRASP.

Dedicación: 13h 30m

Grupo grande/Teoría: 4h 30m

Aprendizaje autónomo: 9h

ACTIVIDADES

PRUEBAS DE RESPUESTA CORTA (CONTROL)

Dedicación: 2h

Grupo grande/Teoría: 2h

PRÁCTICA DE LABORATORIO 1

Descripción:

Introducción al entorno NetBeans. Creación de proyectos. Edición de código. Depuración.

Dedicación: 2h

Grupo pequeño/Laboratorio: 2h

PRÁCTICA DE LABORATORIO 2

Descripción:

Uso de vectores en Java.

Dedicación: 2h

Grupo pequeño/Laboratorio: 2h



PRÁCTICA DE LABORATORIO 3

Descripción:

Uso de objetos en Java: String y StringBuffer. Sesión con métodos relevantes de String: split(), indexOf(), etc

Dedicación: 2h

Grupo pequeño/Laboratorio: 2h

PRÁCTICA DE LABORATORIO 4

Descripción:

Definición e instanciado de una clase.

Dedicación: 2h

Grupo pequeño/Laboratorio: 2h

PRUEBAS DE RESPUESTA LARGA (EXAMEN FINAL)

Descripción:

Examen final

Prueba corta de laboratorio

Dedicación: 1h

Grupo pequeño/Laboratorio: 1h

PRÁCTICA DE LABORATORIO 5

Descripción:

Usando los contenedores en Java.

Dedicación: 2h

Grupo pequeño/Laboratorio: 2h

PRACTICA DE LABORATORIO 7

Descripción:

Pequeña aplicación con contenedores y herencia.

Dedicación: 4h

Grupo pequeño/Laboratorio: 4h

PRACTICA DE LABORATORIO 7

Descripción:

Pequeña aplicación con excepciones y polimorfismo

Dedicación: 4h

Grupo pequeño/Laboratorio: 4h

PRACTICA DE LABORATORIO: ENTREGABLE 1

Descripción:

Trabajo en primer entregable del proyecto

Dedicación: 2h

Grupo pequeño/Laboratorio: 2h

PRÀCTICA DE LABORATORI: ENTREGABLE 2

Descripción:

Sesión de trabajo sobre el entregable 2 del proyecto.

Dedicación: 2h

Grupo pequeño/Laboratorio: 2h

SISTEMA DE CALIFICACIÓN

- Prácticas de laboratorio, Trabajo en grupo y Prueba de laboratorio 30%
- Prueba corta 20%
- Examen final 50%

En esta asignatura se evaluarán las competencias genéricas:

- Comunicación eficaz oral y escrita (Nivel Elemental)
- Uso solvente de los recursos de información (Nivel Elemental)

NORMAS PARA LA REALIZACIÓN DE LAS PRUEBAS.

Para poder aprobar la asignatura hay que realizar las dos entregas del trabajo en grupo y realizar la prueba de laboratorio obligatoriamente.

BIBLIOGRAFÍA

Básica:

- Eckel, B. Piensa en Java. 4a ed. Madrid: Prentice Hall, 2007. ISBN 9788489660342.
- García de Jalón de la Fuente, J. [et al.]. Aprenda Java como si estuviera en primero [en línea]. San Sebastián: Universidad de Navarra. Escuela Superior de Ingenieros Industriales, 1999 [Consulta: 11/11/2011]. Disponible a: <http://www1.ceit.es/asignaturas/Informat1/ayudainf/aprendainf/Java/Java2.pdf>.
- Craig, L. UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado. 2a ed. Madrid: Prentice Hall, 2003. ISBN 8420534382.

Complementaria:

- Documentació d'API de Java [en línea]. [Consulta: 11/11/2011]. Disponible a: <http://java.sun.com/reference/api/index.html>.
- Java de Sun [en línea]. [Consulta: 11/11/2011]. Disponible a: <http://www.java.com/es/>.
- Tutorías de Java [en línea]. [Consulta: 04/02/2015]. Disponible a: http://www.programacion.com/java/tutorial/java_basico/.
- Jacobson, I.; Booch, G.; Rumbaugh, J. El proceso unificado de desarrollo de software. Madrid: Addison-Wesley, 2000. ISBN 8478290362.
- Booch, G.; Rumbaugh, J.; Jacobson, I. El lenguaje unificado de modelado: guía del usuario. 2a ed. Madrid: Addison-Wesley, 2006. ISBN 9788478290765.



RECURSOS

Material informàtico:

- Java Development Kit. Java Development Kit
- <https://netbeans.org/>. NetBeans