

Facultad de Ciencias

Grado en Ingeniería Informática (Obligatoria)
Grado en Matemáticas (Optativa)

GUÍA DOCENTE DE LA ASIGNATURA

G271 - Métodos de Programación

Curso Académico 2013-2014

1. DATOS IDENTIFICATIVOS DE LA ASIGNATURA

Título/s	Grado en Ingeniería Informática (Obligatoria) Grado en Matemáticas (Optativa)
Centro	Facultad de Ciencias
Módulo / materia	ASIGNATURAS DE PRIMER CURSO MATERIA PROGRAMACIÓN DE COMPUTADORES MENCION EN INFORMÁTICA
Código y denominación	G271 - Métodos de Programación
Créditos ECTS	6
Curso / Cuatrimestre	CUATRIMESTRAL (2)
Web	
Idioma de impartición	Español
Forma de impartición	Presencial

Departamento	DPTO. MATEMATICAS, ESTADISTICA Y COMPUTACION
Profesor responsable	MARIO ALDEA RIVAS
E-mail	mario.aldea@unican.es
Número despacho	Facultad de Ciencias. Planta: + 3. DESPACHO - COMPUTADORES TIEMPO REAL (3054)
Otros profesores	PATRICIA LOPEZ MARTINEZ

2. CONOCIMIENTOS PREVIOS

Haber cursado la asignatura 'Introducción al Software'

3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS DEL PLAN DE ESTUDIOS TRABAJADAS EN LA ASIGNATURA

Competencias Genéricas	Nivel
Capacidad de análisis, síntesis y evaluación.	1
Capacidad de organización y planificación.	1
Capacidad de resolución de problemas aplicando técnicas de ingeniería.	1
Capacidad para argumentar y justificar lógicamente las decisiones tomadas y las opiniones.	1
Capacidad de trabajo en equipo.	1
Capacidad de relación interpersonal.	1
Razonamiento crítico.	1
Aprendizaje autónomo.	1
Adaptación a nuevas situaciones.	1
Creatividad.	1
Tener motivación por la calidad.	1
Competencias Específicas	Nivel
Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.	1
Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.	1
Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.	1
Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.	1

3.1 RESULTADOS DE APRENDIZAJE DE LA ASIGNATURA

- Capacidad de crear aplicaciones sencillas utilizando una metodología orientada a objetos, separando las fases de análisis, diseño e implementación
- Capacidad de codificar en Java un diseño basado en objetos
- Capacidad crear jerarquías de clases para sacar partido herencia y el polimorfismo en el desarrollo de programas
- Capacidad de crear y usar clases que utilicen excepciones
- Capacidad de utilizar módulos de programa predefinidos
- Capacidad de documentar correctamente una clase Java
- Capacidad de guardar y recuperar datos de un fichero
- Capacidad de aplicar estrategias de prueba sencillas para módulos
- Capacidad de crear y comprender diagramas UML de clases sencillos
- Conocimiento y aplicación de los algoritmos básicos de recorrido y búsqueda sobre tablas y secuencias.

4. OBJETIVOS DE LA ASIGNATURA

El principal objetivo de la asignatura es que el alumno domine los principales conceptos de la metodología de programación orientada a objetos y sepa aplicarles al desarrollo de programas de mediana complejidad utilizando el lenguaje Java.

La asignatura pretende dotar al alumno de los siguientes conocimientos:

- Conocimientos básicos sobre el proceso integral de desarrollo de software
- Diferencia y relación entre los conceptos de clase, objeto y referencia.
- Conceptos de clase y tipo primitivo
- Utilización de las clases como elemento básico de estructuración del código
- Conceptos de herencia y polimorfismo
- Tratamiento de errores mediante excepciones
- Modularidad y la abstracción como principios básicos para la construcción de una aplicación.
- Utilización de tablas y secuencias: recorridos y búsquedas.
- Diferencia entre ficheros binarios y de texto.
- Técnicas básicas para la prueba de módulos

La asignatura utilizará el lenguaje Java como lenguaje de programación sobre el que concretar y practicar los conceptos estudiados.

5. MODALIDADES ORGANIZATIVAS Y MÉTODOS DOCENTES

ACTIVIDADES	HORAS DE LA ASIGNATURA
ACTIVIDADES PRESENCIALES	
HORAS DE CLASE (A)	
- Teoría (TE)	22
- Prácticas en Aula (PA)	8
- Prácticas de Laboratorio (PL)	30
- Horas Clínicas (CL)	
Subtotal horas de clase	60
ACTIVIDADES DE SEGUIMIENTO (B)	
- Tutorías (TU)	10
- Evaluación (EV)	5
Subtotal actividades de seguimiento	15
Total actividades presenciales (A+B)	75
ACTIVIDADES NO PRESENCIALES	
Trabajo en grupo (TG)	20
Trabajo autónomo (TA)	55
Tutorías No Presenciales (TU-NP)	
Evaluación No Presencial (EV-NP)	
Total actividades no presenciales	75
HORAS TOTALES	150

6. ORGANIZACIÓN DOCENTE DE LA ASIGNATURA												
CONTENIDOS		TE	PA	PL	CL	TU	EV	TG	TA	TU-NP	EV-NP	Semana
1	Presentación de la asignatura	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1
2	1. Introducción a la especificación y diseño de programas. Ingeniería del software. Actividades en el desarrollo del software. Modelo clásico: desarrollo en cascada. Objetivos de los nuevos modelos. Introducción al análisis de requisitos. Diagrama de Clases en UML. Diseño orientado a objetos. Especificación de los métodos. Programación orientada a objetos.	2,00	0,00	4,00	0,00	0,70	0,20	2,20	5,00	0,00	0,00	1,2
3	2. Clases, referencias y objetos. Concepto de clase y objeto. Creación e inicialización de objetos. Tipos primitivos, referencias y objetos. Recolector de basura. Comparación de objetos. Métodos y campos de clase (o estáticos). Anidamiento de clases.	3,00	2,00	6,00	0,00	1,60	1,10	3,60	8,50	0,00	0,00	2,3,4
4	3. Modularidad y abstracción: aspectos avanzados. Conceptos de modularidad y abstracción. Modificadores de acceso básicos. Paquetes. Módulos genéricos. Programación con módulos predefinidos. Documentación del código fuente.	3,00	0,50	2,00	0,00	1,30	0,20	2,20	6,00	0,00	0,00	5, 6
5	4. Herencia y Polimorfismo. Herencia. Clases abstractas. Polimorfismo. La clase Object.	3,00	1,00	4,00	0,00	1,30	0,40	3,60	8,50	0,00	0,00	6, 7, 8
6	5. Tratamiento de errores. Tratamiento de errores por paso de parámetros. Excepciones. Bloques de tratamiento excepciones. La cláusula finally. Patrones de tratamiento de excepciones. Jerarquía de las excepciones. Lanzar excepciones. Usar nuestras propias excepciones. Utilización de excepciones.	3,00	1,00	4,00	0,00	1,30	0,90	2,20	8,50	0,00	0,00	9, 10
7	6. Entrada/salida con ficheros. Conceptos básicos. Flujos de datos (streams). E/S de texto. E/S de texto con formato. E/S binaria.	4,00	1,50	6,00	0,00	1,60	1,10	4,00	8,50	0,00	0,00	11, 12, 13
8	7. Prueba de programas. Verificación y validación. Pruebas del software. Caja negra: particiones de equivalencia. Herramienta JUnit.	2,00	1,00	2,00	0,00	1,00	0,40	2,20	5,00	0,00	0,00	13, 14
9	Revisión general de la asignatura	1,00	1,00	2,00	0,00	1,20	0,70	0,00	5,00	0,00	0,00	15
TOTAL DE HORAS		22,00	8,00	30,00	0,00	10,00	5,00	20,00	55,00	0,00	0,00	

Esta organización tiene carácter orientativo.

TE	Horas de teoría
PA	Horas de prácticas en aula
PL	Horas de prácticas de laboratorio
CL	Horas Clínicas
TU	Horas de tutoría
EV	Horas de evaluación
TG	Horas de trabajo en grupo
TA	Horas de trabajo autónomo
TU-NP	Tutorías No Presenciales
EV-NP	Evaluación No Presencial

7. MÉTODOS DE LA EVALUACIÓN				
Descripción	Tipología	Eval. Final	Recuper.	%
Prácticas	Evaluación en laboratorio	No	Sí	5,00
Calif. mínima	0,00			
Duración	2			
Fecha realización	A lo largo del curso			
Condiciones recuperación	Entrega y presentación de las prácticas antes del examen final (la penalización por entrega fuera de plazo no es recuperable)			
Observaciones	<p>La nota se obtiene en base a:</p> <ul style="list-style-type: none"> - Presentación de las prácticas: a lo largo del curso el profesor requerirá al alumno la presentación de alguna de las prácticas realizadas. - Informes de prácticas: deberán entregarse dentro del plazo establecido. Una entrega fuera del plazo supondrá una penalización en la nota. También supondrá una penalización la entrega de una práctica con errores u omisiones graves. 			
Exámenes prácticos	Evaluación en laboratorio	No	Sí	45,00
Calif. mínima	4,50			
Duración	2 horas			
Fecha realización	A lo largo del curso			
Condiciones recuperación	Exámenes finales de prácticas en los periodos ordinario y de recuperación			
Observaciones	<p>Exámenes realizados en el laboratorio. Está permitido llevar apuntes.</p> <p>Se celebrarán entre 2 y 4 exámenes de prácticas a lo largo del curso.</p> <p>La nota de este apartado se calculará como la media (ponderada) de las notas de los exámenes realizados. Lo normal será que el peso de todos los exámenes sea el mismo, aunque podría darse el caso de primar unos sobre otros si se considera que hay razones para ello.</p> <p>La nota de los exámenes prácticos podrá verse mejorada si el alumno realiza satisfactoriamente los problemas y ejercicios que podrían proponerse a lo largo del curso.</p>			
Examen final	Examen escrito	Sí	Sí	50,00
Calif. mínima	4,50			
Duración	3,5 horas			
Fecha realización	Periodo ordinario de exámenes			
Condiciones recuperación	Examen final en el periodo de recuperación			
Observaciones	<p>Compuesto por varias cuestiones y uno o más problemas.</p> <p>Las cuestiones son preguntas cortas que pretenden evaluar los conocimientos del alumno en alguno de los contenidos de la asignatura.</p> <p>Los problemas pretenden evaluar el dominio de la metodología de programación que tiene el alumno.</p> <p>Para ello se planteará el desarrollo de una aplicación similar a las realizadas en prácticas.</p> <p>Se podrán utilizar apuntes y bibliografía.</p> <p>Para poder presentarse a este examen es obligatorio entregar y presentar las prácticas de la asignatura.</p>			
TOTAL				100,00
Observaciones				
<p>Para aprobar la asignatura es necesario superar la nota mínima de los 'Exámenes prácticos' y del 'Examen final'. En caso de no superar alguna de esas notas mínimas, la nota final será el mínimo de 4.5 y la media obtenida.</p> <p>Para poder presentarse a los exámenes finales (escrito y de prácticas) de los periodos ordinario y de recuperación es obligatorio entregar y presentar las prácticas de la asignatura.</p>				
Observaciones para alumnos a tiempo parcial				

En el caso de estudiantes en regímenes de dedicación a tiempo parcial que no puedan realizar alguno de los exámenes prácticos podrán recuperar dichos exámenes en los exámenes finales de prácticas de los periodos ordinario y/o de recuperación.

Para poder realizar los exámenes finales (escrito y de prácticas) de los periodos ordinarios y de recuperación será obligatorio que, con anterioridad, hayan entregado y presentado las prácticas.

Se espera de estos alumnos que vayan realizando las prácticas de la asignatura al ritmo que se van proponiendo y que las vayan entregando en los plazos que a tal efecto se establezcan. La entrega fuera de plazo de las prácticas les supondrá la misma penalización que a los alumnos que asisten habitualmente a las clases de la asignatura.

8. BIBLIOGRAFIA

BÁSICA

Ken Arnold, James Gosling, David Holmes, "El lenguaje de programación Java", 3ª edición. Addison-Wesley, 2000.

Eitel, Harvey M. y Deitel, Paul J., "Cómo programar en Java", quinta edición. Pearson Educación, Mexico, 2004.

Complementaria

D.J. Barnes y M. Kölling, "Objects first with Java: A practical introduction using Bluej", Prentice Hall, 2003.

Eric J. Braude, "Ingeniería de Software". Alfaomega, 2003.

E. Bueno, y otros. "Algoritmos y Ejercicios resueltos en Java". Prentice Hall 2003. ISBN 84-705-4024-2

The Java Tutorials. <http://java.sun.com/docs/books/tutorial/>

9. SOFTWARE

PROGRAMA / APLICACIÓN	CENTRO	PLANTA	SALA	HORARIO
Entorno de desarrollo Eclipse	Facultad de Ciencias			
Sistema operativo Linux	Facultad de Ciencias			
Compilador Java de Sun, versión 7	Facultad de Ciencias			

10. COMPETENCIAS LINGÜÍSTICAS

- Comprensión escrita Comprensión oral
 Expresión escrita Expresión oral
 Asignatura íntegramente desarrollada en inglés

Observaciones

Lectura de manuales técnicos en inglés