



Universitat de Lleida

GUÍA DOCENTE **PROGRAMACIÓN II**

Coordinación: GIMENO ILLA, JUAN MANUEL

Año académico 2021-22

Información general de la asignatura

Denominación	PROGRAMACIÓN II			
Código	102001			
Semestre de impartición	2o Q(SEMESTRE) EVALUACIÓN CONTINUADA			
Carácter	Grado/Máster	Curso	Carácter	Modalidad
	Doble titulación: Grado en Ingeniería Informática y Grado en Administración y Dirección de Empresas	1	TRONCAL	Presencial
	Grado en Ingeniería Informática	1	TRONCAL	Presencial
Número de créditos de la asignatura (ECTS)	6			
Tipo de actividad, créditos y grupos	Tipo de actividad	PRALAB	TEORIA	
	Número de créditos	3	3	
	Número de grupos	4	2	
Coordinación	GIMENO ILLA, JUAN MANUEL			
Departamento/s	INFORMATICA E INGENIERIA INDUSTRIAL			
Distribución carga docente entre la clase presencial y el trabajo autónomo del estudiante	20% Presencial 20% Virtual 60% Trabajo autónomo			
Información importante sobre tratamiento de datos	Consulte este enlace para obtener más información.			
Idioma/es de impartición	Preferentemente en Catalán (Castellano si algún estudiante muestra dificultades con el Catalán).			
Distribución de créditos	Xavier Domingo (6) Juan Manuel Gimeno (9)			

Profesor/a (es/as)	Dirección electrónica\nprofesor/a (es/as)	Créditos impartidos por el profesorado	Horario de tutoría/lugar
DOMINGO ALBIN, JAVIER JUAN	xavier.domingo@udl.cat	6	Con cita previa
GIMENO ILLA, JUAN MANUEL	juanmanuel.gimeno@udl.cat	9	Con cita previa
ORTEGA MARMOL, ALVARO	alvaro.ortega@udl.cat	3	

Información complementaria de la asignatura

Asumimos que los alumnos han adquirido todos los conceptos presentados en la asignatura de Programación I ya que, a partir de ellos, los extendemos en dos direcciones: la programación orientada a objetos y el diseño recursivo.

Objetivos académicos de la asignatura

- Aplicar el paradigma de Orientación a Objetos a problemas simples.
- Utilizar los tipos básicos de ficheros en Java.
- Diseñar algoritmos recursivos simples.
- Utilizar la documentación estándar de la plataforma Java.
- Manejar un entorno integrado de desarrollo.

Competencias

- **Competencias transversales EPS**
 - **EPS1:** Capacidad de resolución de problemas y elaboración y defensa de argumentos dentro de su área de estudios.
 - **EPS5:** Capacidad para la abstracción y el razonamiento crítico, lógico y matemático.
 - **EPS9:** Capacidad de trabajo en equipo, tanto unidisciplinar como multidisciplinar.
 - **EPS12:** Tener motivación por la calidad y la mejora continua.
- **Competencias específicas de la titulación**
 - **GII-FB3:** Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería.
 - **GII-FB4:** Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
 - **GII-FB5:** Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.
 - **GII-FB7:** Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más

- adecuados a la resolución de un problema.
- **GII-FB9**: Capacidad de conocer, comprender y evaluar la estructura y arquitectura de los computadores, así como los componentes básicos que los conforman.

Contenidos fundamentales de la asignatura

1. Introducción

- 1.1 De C a Java
- 1.2 La biblioteca de la ACM Java Task Force
- 1.3 El programa principal
- 1.4 Uso de funciones auxiliares
- 1.5 Tratamiento de vectores (arrays)
- 1.6 Tratamiento de cadenas de caracteres

2. Programación Orientada a Objetos

- 2.1 Objetos y referencias
- 2.2 Introducción a las clases gráficas de las bibliotecas de la ACM
- 2.3 La clase String
- 2.4 Definición de clases en Java

3. Tratamiento de ficheros

- 3.1 Clases de ficheros
- 3.2 Ficheros secuenciales de texto
- 3.3 Ficheros binarios de acceso directo
- 3.4 Algoritmo de ordenación MergeSort

4. Diseño recursivo

- 4.1 Funcionamiento de las llamadas a funciones
- 4.2 Pensar recursivamente
- 4.3 Recursividad usando índices
- 4.4 Búsqueda binaria
- 4.5 Recursividad múltiple

Ejes metodológicos de la asignatura

Grupos Grandes: Clases Teóricas (3 créditos)

- Parte teórica: clases soportadas por apuntes.
- Parte de aplicación práctica: se trabaja siempre sobre ejemplos.

Grupos Medianos: Clases de Laboratorio (3 créditos)

- Clases dirigidas a la resolución de problemas por parte de los alumnos (se dispone de una colección de problemas y de exámenes de cursos anteriores).
- Resolución de dificultades y seguimiento personalizado de las prácticas.
- Uso de un entorno integrado de desarrollo.

Trabajo Autónomo (no presencial)

- Las prácticas se realizarán en horas no presenciales.
- Se recomienda que el alumno resuelva por su cuenta los problemas de la colección de problemas para practicar y obtener feedback por parte del profesor.

Plan de desarrollo de la asignatura

Setmana	Presencial GG	Presencial GM	No Presencial
1	Presentación + De C a Java (1 a 3)	Netbeans	Estudio y resolución de problemas
2	De C a Java (resto)	Probs 3, 4 i 6	Estudio y resolución de problemas
3	Introducción OO (1, 2)	Probs 1, 2, 5	Estudio y resolución de problemas Práctica 1
4	Introducción OO (3, 4)	Probs 1, 2	Estudio y resolución de problemas Práctica 1
5	Introducción OO (5, 6, 7)	Probs 2, 4, 5	Estudio y resolución de problemas Práctica 1
6	Ampliación OO (8, 9)	Probs 8, 9	Estudio y resolución de problemas Práctica 2
7	Ampliación OO (10, 11)	Probs 10, 11, 12	Estudio y resolución de problemas Práctica 2
8	Ampliación OO (12, 13, 14)	Exámenes anteriores	Estudio y resolución de problemas
9	Evaluación		
10	Tractamiento ficheros (1, 2, 3)	Javadoc	Práctica 2
11	Tractamiento ficheros (4, 5, 6)	Probs 2, 3, 4	Estudio y resolución de problemas Práctica 2
12	Tractamiento ficheros (7, 8)	Probs 5, 6, 7	Estudio y resolución de problemas Práctica 3
13	Recursividad (1, 2, 3)	Probs 8, 9 10	Estudio y resolución de problemas Práctica 3
14	Recursividad (4, 5, 6)	Probs 1 i 2	Estudio y resolución de problemas Práctica 3
15	Recursividad (9, 10)	Probs 3, 4, 5 Exámenes anteriores	Estudio y resolución de problemas
16	Evaluación		
17	Evaluación		
18	Tutorías		Estudio y resolución de problemas Práctica 3

Setmana	Presencial GG	Presencial GM	No Presencial
19	Recuperación		

- Los números en los paréntesis asociados a los temas, se refieren a la sección de los apuntes que se presentará.
- Los correspondientes a los problemas, al número de los problemas a comentar dentro de la lista de problemas.

Sistema de evaluación

Acr	Activitat	Ponderació	Nota mínima	Obligatoria	Recuperable
Par1	Examen 1er parcial	25%	4,0	SI	SI (con el 2do parcial)
Par2	Examen 2do parcial	25%	4,0	SI	SI (examen recuperación, con nota máxima de 7)
Pract1	Práctica 1	15%	NO	NO	Si (al final de curso, con nota máxima de 5)
Pract2	Práctica 2	20%	NO	NO	Si (al final de curso, con nota máxima de 5)
Pract3	Práctica 3	15%	NO	NO	NO

$$\text{Nota final} = 0,25 * \text{Par1} + 0,25 * \text{Par 2} + 0,15 * \text{Pract1} + 0,20 * \text{Pract2} + 0,15 * \text{Pract3}$$

- La asignatura se aprueba con una nota final igual o superior a 5
- Un segundo parcial superado permite recuperar el primer parcial
- La nota del primer parcial sólo se tiene en cuenta si es superior a la del segundo parcial (si no, se coge como nota la del segundo parcial)
- En caso de que el alumno haya de realizar la recuperación, la nota del primer parcial no se tendrá en cuenta y la nota máxima de la parte de teoría será un 7.
- Las dos primeras prácticas son recuperables, pero en caso de re-entrega, la nota máxima será un 5.
- Una práctica detectada como copia (o trabajo no original del alumno) tendrá una nota de 0 i no será recuperable.

Bibliografía y recursos de información

- Básica:
 - Apuntes de la asignatura (en español).
 - Eric S. Roberts, The Art & Science of Java: An Introduction to Computer Science, PearsonEducation, 2008. (hay una versión preliminar disponible en pdf).
 - Eric S. Roberts, Thinking Recuersively with Java, John Wiley & Sons, 2006.
- Complementaria:
 - Documentación de la biblioteca ACM Java Task Force <http://jtf.acm.org/>
 - Kathy Sierra y Bert Bates, Head First Java, O'Reilly, 2003.
 - Jorge A. Villalobos y Rubby Casallas, Fundamentos de Programación. Aprendizaje Activo Basado en Casos. Pearson Pentice-Hall, 2006