



Guía docente

340380 - PROP-I4O23 - Proyecto de Programación

Última modificación: 16/05/2024

Unidad responsable: Escuela Politécnica Superior de Ingeniería de Vilanova i la Geltrú
Unidad que imparte: 723 - CS - Departamento de Ciencias de la Computación.

Titulación: GRADO EN INGENIERÍA INFORMÁTICA (Plan 2018). (Asignatura obligatoria).

Curso: 2024 **Créditos ECTS:** 6.0 **Idiomas:** Catalán

PROFESORADO

Profesorado responsable: Orellana Bech, Bernat

Otros: Orellana Bech, Bernat

CAPACIDADES PREVIAS

Conocimientos de programación y de estructuras de datos:

- Capacidad de resolver problemas algorítmicos de dificultad media a partir de una especificación clara, y de implementar las soluciones en un lenguaje de programación imperativo.
- Conocimiento de los mecanismos básicos de estructuración de programas (modularización, encapsulación, tipos abstractos de datos, clases) y capacidad para aplicarlos a problemas pequeños-medios (unos pocos módulos)
- Conocimiento de los elementos de programación orientada a objetos (clases, objetos, mecanismos de ejecución).
- Familiaridad con un lenguaje imperativo orientado a objetos.
- Capacidad de usar y programar estructuras de datos en este lenguaje.
- Capacidad de usar librerías en este lenguaje.
- Dominio de estrategias básicas para encontrar y corregir errores en módulos sencillos.

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:

1. CECO1. Capacidad para tener un conocimiento profundo de los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.
2. CECO2. Capacidad para conocer los fundamentos teóricos de los lenguajes de programación y las técnicas de procesamiento léxico, sintáctico y semántico asociadas, y saber aplicarlas para la creación, diseño y procesamiento de lenguajes.
3. CECO3. Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.
4. CECO4. Capacidad para conocer los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes y analizar, diseñar y construir sistemas, servicios y aplicaciones informáticas que utilicen dichas técnicas en cualquier ámbito de aplicación.



METODOLOGÍAS DOCENTES

En la asignatura se trabajan técnicas algorítmicas de programación a través de clases teóricas y clases de problemas. En las clases de laboratorio nos fijamos en la programación orientada a objetos de forma especialmente práctica, desarrollando actividades de programación para fijar estas técnicas y finalmente desarrollando un proyecto de envergadura media por la que los estudiantes deben desarrollar las técnicas aprendidas en clase de teoría y combinarlas con las técnicas de programación orientada a objetos que han visto en las clases de laboratorio.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

Aprendizaje de técnicas para identificar la complejidad de un problema y aplicar la estrategia de resolución adecuada.

Aprendizaje de la estructura de grafos para representar problemas combinatorios.

Aprendizaje de las diferentes estrategias algorítmicas para solucionar problemas computacionales.

Aprender conceptos avanzados de la programación orientada a objetos.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas grupo grande	30,0	20.00
Horas grupo pequeño	30,0	20.00
Horas aprendizaje autónomo	90,0	60.00

Dedicación total: 150 h

CONTENIDOS

(CAST) Grafs

Descripción:

Repaso teórico de grafos y de su importancia en informática. Representació computacional de grafos: matrices de adyacencia, listas de adyacencia. Problemas habituales sobre grafos. Recorridos de grafos.

Objetivos específicos:

(CAST) Aprenentatge de la importància dels grafs per representar problemes on hi ha relacions entre elements. Aprenentatge de diferents estructures de representació pels grafs i la eficiència espacial i temporal de les mateixes. Disseny d'algorisme de connectivitat entre nodes del graf: Camins, parts connexes d'un graf, cicles, etc.

Dedicación: 1h

Grupo grande/Teoría: 1h

(CAST) Costos Algorísmics

Descripción:

Medida del coste temporal y espacial. Jerarquía de problemas segun su complejidad. Problemas tratables e intratables. Problemas NP: casos conocidos. Técnica de reducción: identificación de problemas NP.

Objetivos específicos:

(CAST) Adquirir i el concepte de cost temporal i espacial d'un problema. Aprenentatge de la jerarquia de complexitat. Capacitat de discernir un problema tractable d'un d'intractable. Aprenentatge de la tècnica de reducció.

Dedicación: 3h

Grupo grande/Teoría: 3h



Algoritmos generativos

Descripción:

Aplicación de los algoritmos generativos. Búsqueda de cualquier solución o de la solución óptima. Descripción del espacio de búsqueda. Esquema de búsqueda general. Lenguajes de definición de dominios y problemas (STRIPS, PDDL). Búsqueda ciega. Funciones heurísticas. Búsqueda informada. Algoritmos de búsqueda en juegos: Minimax, alfa-beta, Montecarlo

Dedicación: 26h

Grupo grande/Teoría: 14h

Grupo pequeño/Laboratorio: 4h

Aprendizaje autónomo: 8h

Algoritmos voraces

Descripción:

Esquema de los algoritmos voraces. Criterio voraz: necesidad de la demostración de optimalidad. Casos de ejemplo.

Objetivos específicos:

(CAST) Aprenentatge de l'estategia dels algorismes voraçs i identificació de problemes que admeten aquestes solucions.

Dedicación: 12h

Grupo grande/Teoría: 4h

Aprendizaje autónomo: 8h

Programación dinámica

Descripción:

Programas recursivos ineficientes. Memorización. Pasos de la programación dinámica. Ecuaciones recursivas.

Dedicación: 6h

Grupo grande/Teoría: 6h

Técnicas de agregación y divide y vencerás

Descripción:

Esquema del divide y vencerás. Esquema de agregación y eliminación.

Objetivos específicos:

(CAST) Aprenentatge de l'estategia d'agregació i de divideix i venç i identificació de problemes que admeten aquestes solucions.

Dedicación: 2h

Grupo grande/Teoría: 2h

Lenguaje orientado a objetos: Java

Descripción:

Aprendizaje práctico del lenguaje Java

Objetivos específicos:

Aprendizaje práctico del lenguaje Java

Dedicación: 5h

Grupo pequeño/Laboratorio: 5h



Programación Orientada a Objetos Avanzada

Descripción:

Aprendizaje de la utilidad de las características avanzadas de la programación orientada a objetos.

Objetivos específicos:

Aprendizaje de la utilidad de las características avanzadas de la programación orientada a objetos.

Dedicación: 3h

Grupo pequeño/Laboratorio: 3h

ACTIVIDADES

Actividad 1

Descripción:

Trabajo con clases preprogramadas y extensión de las mismas, aplicadas a juegos y búsquedas. Uso de Javadoc. Uso de herencia y polimorfismo.

Competencias relacionadas:

I_CECO2. CECO2. Capacidad para conocer los fundamentos teóricos de los lenguajes de programación y las técnicas de procesamiento léxico, sintáctico y semántico asociadas, y saber aplicarlas para la creación, diseño y procesamiento de lenguajes.

I_CECO4. CECO4. Capacidad para conocer los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes y analizar, diseñar y construir sistemas, servicios y aplicaciones informáticas que utilicen dichas técnicas en cualquier ámbito de aplicación.

Dedicación: 21h

Grupo pequeño/Laboratorio: 5h

Aprendizaje autónomo: 16h

Actividad 2

Descripción:

Programación Orientada a Objetos de algoritmos combinatorios para juegos.

Competencias relacionadas:

I_CECO1. CECO1. Capacidad para tener un conocimiento profundo de los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.

I_CECO4. CECO4. Capacidad para conocer los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes y analizar, diseñar y construir sistemas, servicios y aplicaciones informáticas que utilicen dichas técnicas en cualquier ámbito de aplicación.

I_CECO2. CECO2. Capacidad para conocer los fundamentos teóricos de los lenguajes de programación y las técnicas de procesamiento léxico, sintáctico y semántico asociadas, y saber aplicarlas para la creación, diseño y procesamiento de lenguajes.

Dedicación: 23h

Grupo pequeño/Laboratorio: 5h

Aprendizaje autónomo: 18h



Proyecto

Descripción:

Programación de una IA de juego, usando algoritmos basados en MiniMax, poda alfa-beta, Iterative deepening, tablas de transposición y cálculos heurísticos que incluyen el uso de algoritmos de programación dinámica y/o combinatorios y/o voraces. Uso de herramientas de modelado UML y Javadoc. Uso de repositorios Git para trabajar en grupo.

Competencias relacionadas:

I_CECO4. CECO4. Capacidad para conocer los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes y analizar, diseñar y construir sistemas, servicios y aplicaciones informáticas que utilicen dichas técnicas en cualquier ámbito de aplicación.

I_CECO1. CECO1. Capacidad para tener un conocimiento profundo de los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.

I_CECO3. CECO3. Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.

Dedicación: 48h 40m

Grupo pequeño/Laboratorio: 8h 40m

Aprendizaje autónomo: 40h

SISTEMA DE CALIFICACIÓN

Nota Teoría = $\max(0,5 \text{ Parcial} + 0,5 \text{ Final}, \text{Reevaluación})$

Nota Actividades = $0,5 \text{ Actividad1} + 0,5 \text{ Actividad2}$

Si Nota Teoría ≥ 3 ; NOTA FINAL = $(0,5 \text{ Teoría} + 0,3 \text{ Proyecto} + 0,2 \text{ Actividades})$

Sino NOTA FINAL = $(0,7 \text{ Teoría} + 0,2 \text{ Proyecto} + 0,1 \text{ Actividades})$

BIBLIOGRAFÍA

Básica:

- Cormen, Thomas H. ; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford. Introduction to algorithms [en línea]. 3rd. Cambridge: MIT Press, 2009 [Consulta: 15/02/2024]. Disponible a: <https://search-ebSCOhost-com.recursos.biblioteca.upc.edu/login.aspx?direct=true&AuthType=ip,uid&db=nlebk&AN=2932690&site=ehost-live&ebv=EK&ppid=Page--1>. ISBN 9780262033848.
- Levitin, Anany. Introduction to the design and analysis of algorithms. 3rd. Boston: Pearson Addison-Wesley, 2012. ISBN 9780273764113.
- Edmonds, Jeff. How to think about algorithms. 2nd ed. Cambridge: Cambridge University Press, 2024. ISBN 9781009302135.
- Skiena, Steven S. The Algorithm design manual. 3th ed. Cham: Springer, 2020. ISBN 9783030542559.
- Savitch, Walter. Java : an introduction to problem solving and programming. 8th ed. New York: Pearson, 2019. ISBN 9781292247472.
- Wu, C. Thomas. An Introduction to object-oriented programming with JAVA. 5th. Boston [etc.]: McGraw-Hill, 2009. ISBN 9780073523309.

RECURSOS

Otros recursos:

El siguiente material está disponible en el campus digital:

- Diapositivas de clase
- Ejemplos de código
- Exámenes resueltos de cursos anteriores