Guía docente
270086 - CAP - Conceptos Avanzados de Programación

Última modificación: 13/07/2023

Unidad responsable: Facultad de Informática de Barcelona
Unidad que imparte: 723 - CS - Departamento de Ciencias de la Computación.

Titulación: GRADO EN INGENIERÍA INFORMÁTICA (Plan 2010). (Asignatura optativa).

Curso: 2023 **Créditos ECTS:** 6.0 **Idiomas:** Catalán

PROFESORADO

Profesorado responsable: JORDI DELGADO PIN

Otros: Primer quadrimestre:
JORDI DELGADO PIN - 11, 12

CAPACIDADES PREVIAS

Los alumnos deberían tener conocimientos suficientes de estructuras de datos y algoritmos y de diseño y programación orientada a objetos.

REQUISITOS

- Pre-requisito IES
- Pre-requisito PROP

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:
CES1.1. Desarrollar mantener y evaluar sistemas y servicios software complejos y/o críticos.
CES1.3. Identificar, evaluar y gestionar los riesgos potenciales asociados a la construcción de software que pudiesen presentarse.
CES1.7. Controlar la calidad y diseñar pruebas en la producción de software.

Genéricas:
G5. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

METODOLOGÍAS DOCENTES

La docencia de la asignatura está estructurada en clases de teoría y clases de laboratorio.

En las clases de teoría los profesores presentan los contenidos esenciales de la asignatura. En las clases de laboratorio se practican los contenidos de la asignatura (los presentados en clase y los adquiridos autónomamente) mediante la realización de problemas prácticos. Las clases de laboratorio serán una continuación de las clases teóricas, donde los conceptos nuevos se implementarán a medida que vayan apareciendo.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

1.Conocer las los conceptos básicos y generales de la programación basada en prototipos, de manera que el estudiante sea capaz de entender cómo están implementados estos conceptos en diferentes lenguajes de programación
2.Conocer de manera suficiente un lenguaje basado en prototipos como para ser capaces de ilustrar los conceptos generales en una implementación concreta (Javascript)
3.Conocer qué es un cierre (closure) y algunas técnicas asociadas a su utilización. Entender el concepto de Continuación y usarlo para implementar diversas estructuras de control
4.Conocer un lenguaje de programación dinámico orientado a objetos como Smalltalk.
5.Conocer los conceptos básicos y generales de la reflexión computacional, de manera que el estudiante sea capaz de entender cómo están implementados estos conceptos en diferentes lenguajes de programación
6.Conocer cómo está implementada la reflexión computacional en Smalltalk: acceso y modificación de los miembros de clases en tiempo de ejecución, creación de código en tiempo de ejecución, etc.
7.Ser capaz de desarrollar en equipo un programa de tamaño pequeño-medio que utilice parte de lo que se enseña en el curso

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas actividades dirigidas	6,0	4.00
Horas grupo grande	30,0	20.00
Horas grupo pequeño	30,0	20.00
Horas aprendizaje autónomo	84,0	56.00

Dedicación total: 150 h

CONTENIDOS

Programación basada en prototipos.

Descripción:
Clases vs. prototipos. Herencia vs. delegación

Estructuras de Control: Continuaciones

Descripción:
Las continuaciones son el resultado de poder reificar y eventualmente modificar la pila de ejecución de un programa mientras éste se ejecuta, pero desde el mismo lenguaje. Exploraremos las posibilidades que esto ofrece y las técnicas asociadas.

Iniciación a Smalltalk

Descripción:
Filosofía y orígenes. Encapsulación, miembros privado / (protected) / public, herencia, polimorfismo, early / late binding. Patrones. Smalltalk, aprendizaje y utilización de un entorno Smalltalk

Reflexión: Conceptos generales

Descripción:
¿Qué es la reflexión?. Tipo de reflexión: Introspección, intercesión, reificación. Implicaciones de la reflexión para un lenguaje de programación. Reflexión en programación orientada a objetos.

Implementación en Smalltalk de la reflexión

Descripción:
Acceso a los miembros de clases en tiempo de ejecución. Patrones y Reflexión. Construir objetos de manera reflexiva. Generación de código en tiempo de ejecución.

ACTIVIDADES

Programación basada en prototipos.

Descripción:
El estudiante debería estar atento en clase y trabajar los ejercicios propuestos por el profesor.

Objetivos específicos:
1, 2

Dedicación: 26h
Grupo grande/Teoría: 6h
Grupo pequeño/Laboratorio: 6h
Aprendizaje autónomo: 14h

Estructuras de Control. Continuaciones

Descripción:
Es necesario que el estudiante esté atento en clase y realice los ejercicios propuestos.

Objetivos específicos:
3

Dedicación: 26h
Grupo grande/Teoría: 6h
Grupo pequeño/Laboratorio: 6h
Aprendizaje autónomo: 14h

Control teoría

Objetivos específicos:
5, 6

Dedicación: 16h
Actividades dirigidas: 2h
Aprendizaje autónomo: 14h

Práctica

Descripción:
Fecha de entrega; A determinar una vez hayan acabado las clases.

Objetivos específicos:
7

Competencias relacionadas:
G5. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Introducción a Smalltalk

Descripción:
El estudiante debería estar atento en clase y trabajar los ejercicios propuestos por el profesor.

Objetivos específicos:
4

Dedicación: 18h
Grupo grande/Teoría: 6h
Grupo pequeño/Laboratorio: 6h
Aprendizaje autónomo: 6h

Reflexión: Conceptos generales

Descripción:
El estudiante debería estar atento en clase y trabajar los ejercicios propuestos por el profesor.

Objetivos específicos:
5

Dedicación: 14h
Grupo grande/Teoría: 2h
Grupo pequeño/Laboratorio: 2h
Aprendizaje autónomo: 10h

Reflexión enSmalltalk

Descripción:
El estudiante debería estar atento en clase y trabajar los ejercicios propuestos por el profesor.

Objetivos específicos:
6, 7

Competencias relacionadas:
G5. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 36h
Grupo grande/Teoría: 8h
Grupo pequeño/Laboratorio: 8h
Aprendizaje autónomo: 20h

Control final

Objetivos específicos:
1, 2, 3

Dedicación: 14h
Actividades dirigidas: 2h
Aprendizaje autónomo: 12h

SISTEMA DE CALIFICACIÓN

El método de evaluación de la asignatura consistirá en dos pruebas de carácter teórico (T1 y T2), una a mediados de curso y otra al final y una práctica de tamaño pequeño-medio (Practica).

Entonces, el método de evaluación sería:
0.8 * Teoría + 0.2 * Practica
donde:
Teoría: 0.5 * T1 + 0.5 * T2

Competencia transversal "Trabajo en equipo":

Se evalúa usando una rúbrica simple en que el tutor de cada grupo puntúa los diferentes aspectos del trabajo en equipo de cada miembro de los grupos.

BIBLIOGRAFÍA

Básica:
- Black, A.P.; Ducasse, S.; Nierstrasz, O.; Pollet, D.; Cassou, D.; Denker, M. Pharo by example. Square Bracket Associates, 2009. ISBN 9783952334140.
- Noble, J.; Taivalsaari, A.; Moore, I. (Editors). Prototype-based programming: concepts, languages, and applications. Springer, 1999. ISBN 9814021253.

Complementaria:
- Proceedings of OOPSLA '87, ACM SIGPLAN Notices. 22(12):147--155, 1987.

RECURSOS

Enlace web:
- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.5354&rep=rep1&type=pdf- https://developer.mozilla.org/es/JavaScript