

Paradigmas de Programación IECI

Presentación del ramo: descripción, perfil de egreso,
resultados del aprendizaje

Profesor: Oscar San Martín
osanmartin@ubiobio.cl

Descripción

Esta asignatura permitirá al estudiante **aplicar conocimientos asociados a Paradigmas de Programación en el desarrollo de software y resolver problemas simples utilizando un lenguaje de programación orientado a objeto (Java).**

- Se usará Java como lenguaje, pero el conocimiento es aplicable a cualquier otro lenguaje orientado a objetos.



Perfil de egreso

- **CE6.Construir software** de acuerdo a estándares definidos, acorde a requerimientos y modelos previamente establecidos.
- **CE2.Identificar hardware y software disponibles** para establecer la configuración de la plataforma tecnológica a utilizar en un sistema necesario para la organización.
- **CE3.Planificar** la incorporación y/o adecuación de la **informática en la organización** en base al diagnóstico de los problemas dirigiendo el capital humano y procesos asociados en la utilización e incorporación de TIC en la organización para la adecuada ejecución de los procesos.

Resultados de aprendizaje

1. Resuelve apropiadamente un problema aplicando uno o más paradigmas de programación para la construcción de un software.
2. Interpreta correctamente documentación de análisis y diseño de software para comprender la interrelación entre diseño y programación.
3. Emplea estándares y buenas prácticas de codificación correspondientes al paradigma usado para asimilar e interpretar su aplicación en la programación.
4. Analiza las características asociadas a diferentes paradigmas de programación para seleccionar el más apropiado en la resolución de un problema específico.

Resultado de aprendizaje 1

Conceptual

- Elementos de Diseño de Interfaces Gráficas de Usuario.
- Conceptos de Paradigma Orientado a Objetos:
 - Clase, objetos y su relación con la realidad que representan.
 - Estructura de una clase: atributos y operaciones.
 - Conceptos de abstracción, encapsulamiento, agregación, asociación, herencia y polimorfismo.
- Conceptos de Paradigmas Guiado por Eventos.

Resultado de aprendizaje 1

Procedimental

- Analogías de objetos y sus relaciones en el mundo real.
- Utilización de elementos de diseño de Interfaces Gráficas de Usuario.

Actitudinal

- Valoración del aporte de los enfoques: orientado a objeto, guiado por eventos.
- Trabaja colaborativamente en el desarrollo del proyecto.
- Cumple responsablemente con compromisos adquiridos con su grupo de trabajo.

Resultado de aprendizaje 2

Conceptual

- Propósito de los diagramas de clases y de los Diagramas de Casos de Usos.
- Símbolos del diagrama de clases de un lenguaje de modelado y su significado: clase, atributo, operación, asociación (o uso), agregación, multiplicidad, rol, navegabilidad, herencia, clase abstracta.
- Traduce conceptos OO expresados en UML a su equivalente en lenguaje de programación.

Procedimental

- Aplicación de reglas para interpretar diagramas de clases.

Actitudinal

- Valoración del aporte de las herramientas de documentación para análisis y diseño de software.
- Trabaja colaborativamente en el desarrollo e interpretación de diagramas de clases y de casos de uso.
- Aplica y respeta las reglas de uso, diseño e interpretación de herramientas como UML.

Resultado de aprendizaje 3

Conceptual

- Estilos correctos de programación.
- Modelo Vista Controlador (MVC).
- Diseño por Patrones: Patrones de creación, Patrones estructurales, Patrones de comportamiento

Procedimental

- Valoración de estándares de desarrollo en su aplicación al desarrollo de Sw.
- Valoración de la aplicación de buenos estilos de programación.

Actitudinal

- Aplica estándares de codificación y buenas prácticas y estilos de programación según el paradigma usado.
- Trabaja colaborativamente en la aplicación de buenas prácticas de programación.
- Usa correctamente el enfoque de diseño por patrones.

Resultado de aprendizaje 4

Conceptual

- Elementos de los Paradigmas: Estructurado, Funcional, Lógico, Imperativo.

Procedimental

- Diferenciación de elementos significativos de los Paradigmas.
- Selección del paradigma a utilizar ante un tipo específico de problema.
- Aplicación del paradigma en la resolución de pequeños ejemplos.

Actitudinales

- Rigurosidad en el trabajo
- Criterios para el trabajo en equipo
- Compromiso en la fecha de entrega de trabajos y tareas.

Evaluación

- Primer Certamen teórico - práctico (20%)
- Segundo Certamen teórico - práctico (20%)
- Promedio de Test-Trabajos 20% (4 test)
- Proyecto de Desarrollo de Software en Java con POO (40%)
- Este proyecto de programación podrá ser en grupo de hasta 2 alumnos. Las copias se calificarán con nota 1. Las instrucciones se entregarán posterior al certamen 1.
- El funcionamiento de los programas entregados por los alumnos debe poder ejecutarse y funcionar para que se puedan evaluar.

Bibliografía

- Deitel, Paul. Java, cómo programar. 9 Edición, Pearson Educación, 2012.
- Weitzenfeld, Alfredo. Ingeniería de software orientada a objetos con UML, Java e internet. Thomson, 2005.
- Deitel, Harvey M. “Cómo programar en Java”. 5 Ed, Pearson Educación, 2004.
- Braude, Eric J. Ingeniería de software: una perspectiva orientada a objetos. Alfaomega, 2003.
- Arnow, David M. Introducción a la programación con JAVA TM :un enfoque orientado a objetos. Primera Edición, Addison-Wesley, 2001.
- Larman, Craig. UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado. Segunda Edición, Pearson, 2003.