

PROGRAMA DE LA ASIGNATURA

Programación Orientada a Objetos

Postgrado de Física Fundamental

Facultad de Ciencias

Universidad de Los Andes

1. IDENTIFICACIÓN Y DESCRIPCIÓN.

UBICACIÓN: Electiva.

UNIDAD CRÉDITO: 4.

PRELACIÓN: Ninguna.

2. JUSTIFICACIÓN.

Desde hace ya varias décadas se está desarrollando una clase de física teórica que se vale de las simulaciones numéricas usando computadoras. Esta forma de hacer física, que se denomina física computacional, se basa sobre una infraestructura de software cada vez más complejo y una de las formas más eficaces que se han encontrado para el desarrollo de esta plataforma es mediante la Programación Orientada a Objetos.

El análisis y diseño orientado a objetos, u OO, es un paradigma de programación relativamente nuevo que permite tratar la complejidad mediante la división del programa en módulos manejables, reducir la complejidad mediante la abstracción, soportar los nuevos desarrollos con los desarrollos previos y reducir el tiempo y esfuerzo de pruebas.

3. REQUISITOS.

El estudiante debe abordar esta materia con buenas bases conceptuales y destreza en programación y diseño algorítmico. La aprobación de este curso es recomendable para elaborar trabajos de investigación en los que se requiera el desarrollo de software para simulación, como por ejemplo en el área de Sistemas Complejos, Caos y Sistemas No Lineales.

4. OBJETIVOS GENERALES.

- Adquirir los conceptos básicos de la programación orientada a objetos.
- Diseñar un software relativamente complejo utilizando las técnicas orientadas a objeto.
- Implementar, utilizando algún lenguaje de programación orientado a objeto, una aplicación.

5. CONTENIDO.

Programación orientada a objetos (OO)

Implementación de la programación OO

Instalación y Compilación del Software

Componentes básicos de la tecnología OO

Diseño de clases orientadas a objeto

Manipulación y optimización de código OO

Diseño de Interfaces Gráficas de Usuario

Manejo de Entrada/Salida y Redes

6. BIBLIOGRAFIA Y MATERIALES

Bibliografía

- Cualquier libro de OO
- <http://www.javasoft.com>
- <http://www.python.org>
- <http://gcc.gnu.org>

Materiales

- <http://sai.ula.ve>
- <http://webdelprofesor.ula.ve/ciencias/kay>

7. CRONOGRAMA DE CLASES

Semana 1 : Programación orientada a objetos (OO)

- Comprender la abstracción de clases
- Identificar objetos
- Comprender las ventajas del encapsulamiento
- Comprender la relación entre objetos y clases
- Comprender la herencia
- Comprender las subclases, su especialización y el polimorfismo
- Comprender las clases abstractas

Semana 2 : Implementación de la programación OO

- Comprender el código de declaración de clases
- Declarar variables de tipos primitivos y de clases
- Usar las clases abstractas y la herencia
- Usar la declaración `import` para incluir clases
- Usar la declaración `package` para agrupar clases
- Definir el comportamiento de una clase
- Explicar la sobrecarga de métodos
- Usar un constructor para instanciar a un objeto

Semana 3 : Instalación y Compilación del Software

- Describir las facilidades fundamentales y los objetivos de la tecnología OO
- Configurar un ambiente de desarrollo para programar OO
- Usar la documentación ofrecida por el ambiente de desarrollo

- Escribir, compilar y correr un ejemplo simple

Semana 4 : Componentes básicos de la tecnología OO

- Crear arreglos de diferentes tipos de datos
- Diseñar una clase con atributos y métodos
- Implementa una interface
- Usar colecciones para almacenar y manipular datos

Semana 5 : Diseño de clases orientadas a objeto

- Crear clases utilizando el concepto de encapsulamiento
- Crear clases que hereden características de su clase padre
- Explicar algunos conceptos avanzados del diseño de clases como polimorfismo y clases internas

Semana 6 : Manipulación y optimización de código OO

- Manejar las excepciones que ocurran durante la ejecución de un programa
- Crear un clase propia para el manejo de excepciones
- Usar hebras (*threads*) en los códigos, extendiendo la clase `Thread` e implementando una interface `Runnable`

Semana 7-8 : Diseño de Interfaces Gráficas de Usuario

- Crear Interfaces Gráficas de Usuario (GUI)
- Escribir código para la manipulación de eventos en las GUI
- Explicar el concepto de *applets*
- Crear aplicaciones y *applets* basados en GUI

Semana 9-10 : Manejo de Entrada/Salida y Redes

- Explicar los conceptos de *streaming* y *networking*
- Describir las características y el propósito del *streaming*
- Explicar cómo se maneja la E/S de bytes y de caracteres
- Implementar el modelo de *networking* para establecer conexiones por la red.

Semana 11-16 :Desarrollo y evaluación del proyecto