



Unidad II: El Paradigma Orientado a Objeto.

Tema 1: Paradigmas de Programación

Coordinación de Introducción a la Computación

Contenido

1. Paradigmas de Programación.

1. Programación Imperativa
2. Programación Lógica
3. Programación Funcional
4. Programación Declarativa
5. Programación Estructurada
6. Programación Dirigida por Eventos
7. Programación Orientada a Aspectos
8. Programación Orientada a Objetos

2. Paradigma Orientado a Objeto (POO)

➤ Definición y Ventajas.

➤ Características.

➤ Elementos básicos de la POO:

- ✓ Clase.
- ✓ Objeto.
- ✓ Atributo.
- ✓ Método.

➤ Propiedades:

- ✓ Abstracción
- ✓ Encapsulamiento/Ocultamiento.
- ✓ Modularidad.

3. Clases, Objetos y sus interrelaciones dentro del mundo real.

4. Lenguaje Formal

- ✓ Palabras claves de uso frecuente en lenguaje formal.
- ✓ Elementos propios del lenguaje formal usado
- ✓ Constructor
- ✓ **Funciones**
 - ✓ Elementos de una función
 - ✓ Funciones que no retornan valor. Estructura básica
 - ✓ Funciones que retornan valor. Estructura básica.
- ✓ **Parámetros**
 - ✓ Definición
 - ✓ Estructura básica
 - ✓ Parámetros por referencia
 - ✓ Parámetros por valor
- ✓ **Métodos**
 - ✓ Sintaxis
- ✓ **Diferencia entre función y método**
- ✓ **Forma de escribir:**
 - ✓ Identificadores
 - ✓ Comentario

1. Paradigmas de Programación.

Definición de Paradigma

Un paradigma es una teoría, patrón o modelo.



El concepto fue introducido por Thomas Kuhn (1975) para explicar los cambios o "revoluciones científicas". Para Thomas Kuhn el motor de toda evolución implica un cambio de paradigma, por lo cual la nueva teoría no tiene por qué falsificar a la anterior, puede ser una alternativa, un nuevo modelo o teoría modelo, un paradigma nuevo.

1. Paradigmas de Programación.

Tipos de Paradigmas

- ✓ Programación Imperativa
- ✓ Programación Lógica
- ✓ Programación Funcional
- ✓ Programación Declarativa
- ✓ Programación Estructurada
- ✓ Programación Dirigida por Eventos
- ✓ Programación Orientada a Aspectos
- ✓ Programación Orientada a Objetos



1. Paradigmas de Programación.

Diferencias entre paradigmas

Programación Imperativa

- Describe la programación en términos del estado del programa y sentencias que cambian dicho estado.
- Los programas imperativos son un conjunto de instrucciones que le indican al computador cómo realizar una tarea.
- La implementación de hardware de la mayoría de computadores es imperativa; prácticamente todo el hardware de los computadores está diseñado para ejecutar código de máquina, que es nativo al computador, escrito en una forma imperativa.

Programación Lógica

- Consiste en la aplicación del corpus de conocimiento sobre lógica para el diseño de lenguajes de programación.
- Es un tipo de paradigma de programación dentro del paradigma de programación declarativa.
- La programación lógica gira en torno al concepto de predicado, o relación entre elementos.

1. Paradigmas de Programación.

Diferencias entre paradigmas

Programación Funcional

- Es un paradigma de programación declarativa basado en la utilización de funciones aritméticas que no maneja datos mutables o de estado. Enfatiza la aplicación de funciones, en contraste con el estilo de programación imperativa, que enfatiza los cambios de estado.
- Las hojas de cálculo también pueden ser consideradas lenguajes de programación funcional.
- JavaScript, uno de los lenguajes más ampliamente utilizados en la actualidad, también incorpora capacidades de programación funcional.

Programación Declarativa

- Está basado en el desarrollo de programas especificando o "declarando" un conjunto de condiciones, proposiciones, afirmaciones, restricciones, ecuaciones o transformaciones que describen el problema y detallan su solución. La solución es obtenida mediante mecanismos internos de control, sin especificar exactamente cómo encontrarla (tan sólo se le indica a la computadora que es lo que se desea obtener o que es lo que se está buscando). No existen asignaciones destructivas, y las variables son utilizadas con Transparencia referencial

1. Paradigmas de Programación.

Programación Estructurada

Significa escribir un programa de acuerdo a las siguientes reglas :

- El programa tiene un diseño modular.
- Los módulos son diseñados de modo descendente.
- Cada módulo se codifica de acuerdo a tres estructuras de control básicas: secuencia, selección y repetición.

Programación Orientada a Objeto

- Se basa en organizar el software como una colección de objetos que incorporan tanto características como comportamiento.
- La programación orientada a objetos es una evolución de la programación estructurada.

1. Paradigmas de Programación.

Diferencias entre paradigmas

Programación Dirigida Por Eventos

- Es un paradigma de programación en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurran en el sistema, definidos por el usuario o que ellos mismos provoquen.

Programación Orientada a Aspectos

- Es un paradigma de programación relativamente reciente cuya intención es permitir una adecuada modularización de las aplicaciones y posibilitar una mejor separación de incumbencias.
- Gracias a la POA se pueden encapsular los diferentes conceptos que componen una aplicación en entidades bien definidas, eliminando las dependencias entre cada uno de los módulos.
- El término POA es usado para referirse a varias tecnologías relacionadas como los métodos adaptativos, los filtros de composición, la programación orientada a sujetos o la separación multidimensional de competencias.

.



Unidad II: El Paradigma Orientado a Objeto.

Tema 1: Paradigmas de Programación

Coordinación de Introducción a la Computación