Programación orientada a objetos

Objeto: debe ser funcional, y posee atributos. Debe ser el objeto quien realice una acción, no la persona que mira el objeto.

En este concepto estricto la fe y el amor no son objetos.

Todo lo del mundo es un objeto (?)

Un objeto se puede describir, en apariencia y el “qué hace”.

Para definir un objeto este debe tener:

-Atributos

-Comportamiento

Al encontrarse con un nuevo objeto se interactúa con él, y se analizan los comportamientos que presentan.

Ejemplo de los huevos, se debe dar un contexto o características

**Pasos:**

1. **Defino los objetos**
2. **Defino las características y comportamientos de los objetos**
3. **Defino las relaciones entre los objetos**

Ejemplo:

Puertas de carro NO son un atributo

La cantidad de puertas SÍ son un atributo (o el color, o el cómo se abre)

Esto debido a que atributo = variable y las puertas no son variables.

**Paradigma**

Es un patrón o modelo de cómo hacer algo, es decir, una forma de hacer las cosas. Una forma en que yo entiendo teorías o metodologías con respecto a un tema específico en un contexto científico

En nuestro contexto: resolver problemas mediante el desarrollo de software

Paradigma

|-Se puede decir además que es el estilo fundamental de programación

|-Forma de construir estructuras y elementos de los programas

|-Es cómo el lenguaje ve al mundo. Ej: Si el lenguaje es del paradigma orientado a objetos nos sugiere resolver el problema por medio de objetos

Existen varios paradigmas…

1. Funcional

2. Imperativo

3. Lógico

4. Orientado a objetos

Principal elemento para resolver el problema y ejemplos de lenguajes que usan esos paradigmas

Orientado objetos:

Ejemplos:

-Java

-C++

-SmallTalk

-C#

Lógico:

El menos conocido de todos, se define a través de proposiciones lógicas, qué es verdad y en qué condiciones, resulta como una base de datos de Verdad o Falso

Ejemplos:

-Prolog

Imperativo:

Utiliza estructuras para resolver problemas

Ejemplos:

-C

-C++

-PHP

Funcional:

Recurre a funciones para resolver los problemas

Ejemplos:

-Scheme

-Lisp

-F#

**Paradigma orientado a objetos**

Nos permite analizar, diseñar y construir programas bajo el hecho de analizar objetos.

Conceptos \*Buscar funcionamiento y código de esto en Java.

Abstracción: Es una **representación** no exacta de la realidad.

Encapsulamiento:

Clases:

Atributos:

Métodos:

Instancia:

Herencia:

Polimorfismo:

Consideraciones

-El código debe estar identado

-Atributos y métodos significativas