

Conceptos de programación



Índice

1. Atributo.....	3
2. CASE.....	3
3. Clase y objetos.....	3
4. Mensaje.....	4
5. Modelo UML (Lenguaje de Modelado Unificado).....	4
6. OMG (Object Management Group).....	4
7. OO (Orientación a Objetos).....	5

1. Atributo

- *Investiga qué es un atributo en programación orientada a objetos.*

A groso modo, hace referencia a las propiedades de un objeto dentro de programación, su estructura, sus variables y métodos a groso modo.

- *Describe un ejemplo de atributo en un contexto real.*

Por ejemplo, en java tienes un archivo (clase), que se llama Perro, y su código, es toda una estructura que emula lo que sería las actividades reales de un perro a través de variables y métodos.

2. CASE

- *Investiga qué significa CASE en el desarrollo de software.*

CASE son las siglas de Computer Aided Software Engineering, es un concepto que hace referencia al desarrollo y mantenimiento de proyectos software con herramientas automatizadoras.

- *¿Cuál es la importancia de las herramientas CASE en el proceso de desarrollo?*

Básicamente, son herramientas que se utilizan a la hora de hacer esquemas y documentación general para poder order cada cosa en lo que toca, almenos por lo que he leído un poco por encima.

Una manera de planificar desarrollos de proyectos.

3. Clase y objetos

- *Investiga la diferencia entre una clase y un objeto.*

La clase es digamos la plantilla, el lienzo sobre la que se trabaja con el código, lo que a posteriori, puede llegar a ser un objeto, a groso modo, el objeto es el resultado que se obtiene dentro de una clase a la hora de programar.

- *Proporciona un ejemplo de una clase y su correspondiente objeto.*

Por ejemplo, el caso del ejercicio 1, tenemos una clase a la que llamamos Perro (sería el archivo), para a posteriori, dentro de la clase Perro meter códigos y métodos que compongan el objeto de Perro.

4. Mensaje

- *Investiga qué es un mensaje en programación orientada a objetos.*

A groso modo y según he leído, un mensaje se refiere a la hora de llamar a objetos.

- *¿Cómo se relaciona un mensaje con los métodos de una clase?*

Básicamente, como una solicitud que pide ejecutar el código, o parte del código de un objeto de una clase.

5. Modelo UML (Lenguaje de Modelado Unificado)

- *Investiga qué es UML y su propósito en el desarrollo de software.*

Es una herramienta para moldear esquemas de programación orientada a objetos que ofrece buenas praxis a la hora de armarlos, por lo visto, se usa bastante y es lo suyo de útil.

- *Describe al menos dos tipos de diagramas UML y su utilidad.*

1. **Diagrama de despliegue:** Arquitectura física de un sistema
2. **Diagrama de secuencia:** Muestra como objetos interactúan entre si en sucesión.

6. OMG (Object Management Group)

- *Investiga qué es OMG y su relación con UML.*

Es una organización sin fin de lucro que busca crear estándares para que la gente sea de bien, y no haga churros en software.

Su relación con UML es que, esta organización es directamente su creadora.

- *¿Por qué es relevante OMG en el contexto de estándares de modelado?*

Es una organización mundial ya establecida con diversas herramientas que reúne a gente profesional de todos lados para armar estándares buenos para la gente, más claro, agua.

Están asentados, tienen buena reputación, buen modus operandi, y al final del día, es una organización que lo que busca es promover buenas praxis y herramientas para todo el mundo, lo cuál es la hostia.

7. OO (Orientación a Objetos)

- *Investiga qué significa la orientación a objetos en programación.*

Es un enfoque que se da en programación, para plantear, o mejor dicho, simular cosas y sus comportamientos de la vida real dentro de la programación, como lo pueden ser personas, registros, animales, figuras, etc.

- *¿Cuáles son los principios fundamentales de la programación orientada a objetos?*
 1. **Abstracción:** La creación de conceptos abstractos en el vacío que sirvan como molde para otros archivos.
 2. **Encapsulamiento:** Protección de datos, accesibilidad de unos archivos y datos a otros.
 3. **Polimorfismo:** El polimorfismo es la cosa que permite distinguir si estamos llamando a métodos, también esto aplicando por ejemplo, a métodos de mismo nombre, o que se heredan entre padres e hijos.
 4. **Herencia:** Hace que de unos archivos a otros se hereden ciertos comportamientos de un archivo de base, creando con esto, jerarquías de archivos que se basan de un concepto básico a otro. Por ejemplo, una clase básica llamada Animal que alomejor puede tener varios hijos llamados; Aves, Terrestres, Acuáticos, y con esto hagamos referencia a ciertos datos que tengan esos tipos de animales.