**Programación Orientada A Objetos**

La programación orientada a objetos es un paradigma de la programación en el que se crean objetos para la manipulación de datos y donde, por lo general, cada objeto ofrece una funcionalidad especial.

Las bases de la programación orientada a objetos son: abstracción, encapsulación, modularidad y jerarquización.

La abstracción es un proceso mental de extracción de las características esenciales, ignorando los detalles.

La encapsulación es ocultar los detalles que dan soporte a un conjunto de características esenciales de una abstracción. Existirán dos partes, una visible que todos tienen acceso y se aporta la funcionalidad, y una oculta que implementa los detalles internos.

El modularidad es descomponer un sistema en un conjunto de partes. Aparecen dos conceptos muy importantes: acoplamiento y cohesión.

* El acoplamiento entre dos módulos mide el nivel de asociación entre ellos; nos interesa buscar módulos poco acoplados.
* La cohesión de un módulo mide el grado de conectividad entre los elementos que los forman; nos interesa buscar una cohesión alta.

**Variables:**

es un identificador que representa un espacio de memoria que contiene información. La información es del tipo con que se declaró la variable. Hay 2 categorías de tipos para variables, los tipos primitivos (como int, short, byte, long, char, boolen y double) y las referencias a objetos (como String, Array y otros objetos).

Java tiene, además, tres tipos de variables, que son: De instancia De clase Locales Las variables de instancia, se usan para guardar los atributos de un objeto particular.

Las variables de clase son variables que guardan el mismo valor, para todos los objetos de una clase determinada.

Las variables locales, se utilizan dentro de los métodos, en Java, las variables locales pueden declararse en el momento en que serán utilizadas, y una buena costumbre es inicializar las variables al momento de declararlas.

**Constantes:**

es una variable que mantiene un valor inmutable durante la ejecución del programa, se define utilizando el modificador "final".

**Operadores:**

Los operadores en JavaScript son:

* operadores aritméticos

+ suma

-resta

\* multiplicación

/división

%residuo

++var preincremento

--var predecremento

var++ postincremento

var-- postdecremento

* Operadores de asignación

= asignación

+= asignación de suma

-= asignación de resta

\*= asignación de multiplicación

/= asignación de división

%= asignación de residuo

* Operadores relacionales

< menor que

<= menor o igual que

> mayor que

>= mayor o igual que

== igual a

!= no igual

* Operadores lógicos

&& AND

|| OR

! NOT

^ OR exclusivo

**Expresiones:**

Las expresiones aparecen dentro del cuerpo de los métodos y están formadas por operandos y operadores. Los operandos pueden ser atributos, parámetros, literales, constantes o llamadas de métodos, mientras que los operadores son los que indican la manera de calcular el valor de la expresión. Los operadores que se pueden utilizar en una expresión dependen del tipo de los datos de los operandos que allí aparezcan.

**Ciclos:**

Existen tres tipos diferentes de ciclos:

While

do...while

for

* Los cuales funcionan siempre y cuando, la condición que evalúa sea verdadera.
* El ciclo do...while se ejecuta cuando menos una vez, si la condición es verdadera se vuelve a ejecutar.
* Por el contrario, while y el ciclo for evalúan primero la condición, en caso de ser falsa, los ciclos nunca se ejecutan.
* Para que la condición vaya cambiando, incluimos un contador dentro del código del ciclo, el cual avanza cada vez que se ejecuta el ciclo. Y dentro de la condición se evalúa a la variable del contador.

Imagen que contiene Diagrama

Descripción generada automáticamente