**Programación Orientada a Objetos**

**Nombre:** Gabriela Loyola Romero

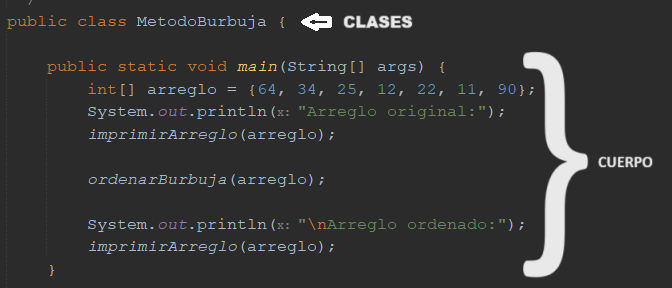
* **La implementación de clases, objetos, constructores**

1. **Clases:**

Las clases se definen como un conjunto de cosas (físicas o abstractas) con el mismo comportamiento o las mismas características.

La implementación de una clase tiene dos componentes: la declaración y el cuerpo de una clase.

Por ejemplo,



1. **Objetos:**

Los objetos al ser declarados dentro de una clase deben ser presentados con una interfaz para que puedan interactuar con el exterior.

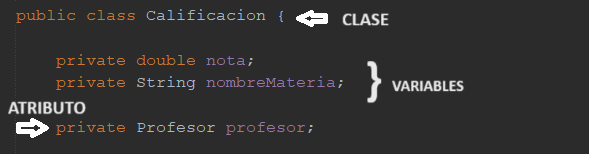
Los objetos constan de:

**Tiempo de vida:**  Los objetos son creados mediante un mecanismo llamado instanciación y cuando dejan de existir se podría decir que son destruidos ya que el objeto solo existe durante una parte de la ejecución de un programa.

**Estado:** Todo objeto posee un estado definido por sus atributos, con los cuales definimos las propiedades del objeto.

**Comportamiento:** Todo objeto debe presentar una interfaz, definida por sus métodos, para que el resto de los objetos que componen los programas puedan interactuar con él.

Ejemplo de objetos,



1. **Constructores:**

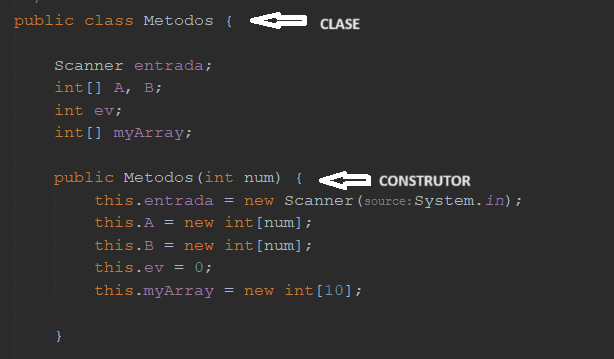
Es un método especial que se utiliza para inicializar un objeto recién creado y asignarle valores iniciales a sus variables de instancia.

No cuenta con valores de retorno ya que su función principal es inicializar un objeto, además puede aceptar argumentos, lo que permite configurar el objeto con valores específicos al momento de su creación."

Para su implementación hay que considerar que el constructor debe tener el mismo nombre que la clase a la que pertenece.

Un constructor puede tener parámetros, mismos que se utilizan para proporcionar valores iniciales a los campos de datos de la clase. Estos parámetros se especifican en la definición del constructor, entre los paréntesis.

Ejemplo,



* **Uso de estructuras de datos en lenguaje de programación Python (list y tuple)**

1. **Listas:**

Las listas en Python son utilizadas para almacenar múltiples valores en una única variable. Esto puede parecer innecesario, pero resuelve muchos desafíos.

Para declarar una lista en Python, se deben seguir las mismas normas básicas que se siguen para declarar una variable cualquiera, con un pequeño cambio en la sintaxis.

Para poder recorrer una lista en Python podemos usar el ciclo repetitivo for ya que este nos ayudará a recorrer la lista en orden.

Las listas son [mutables](https://docs.python.org/es/3/glossary.html#term-mutable), y sus elementos son normalmente homogéneos y se acceden iterando a la lista.

De hecho, las listas son muy parecidas a los arreglos en Java.

1. **Tuplas:**

Las tuplas son estructuras de datos similares a las listas, pero a diferencia de ellas, son inmutables, lo que significa que no se pueden modificar después de ser creadas.

Las tuplas se utilizan para almacenar datos relacionados que no deben cambiar, como las coordenadas de un punto en un plano. Se crean utilizando paréntesis y los elementos se separan por comas.

Wiki:

https://es.slideshare.net/efsolis/5-47311297

<https://blog.hubspot.es/website/que-es-constructor-java>

<https://www.programarya.com/Cursos/Python/estructuras-de-datos/listas>

https://docs.python.org/es/3/tutorial/datastructures.html