Apuntes java

Important definitions.

Class: Define tipo de objetos.

Static: Siempre está en memoria. / No necesitamos hacer una estancia.

Public: Es quien puede acceder (public es todo el mundo), private(solo un número determinado o alguien en específico puede acceder)

Diferencia entre método y variable: El método es algo que puede hacer una instancia y las variables son características de ese objeto.

Block: Un "bloque" en programación generalmente se refiere a un conjunto de declaraciones de código que están agrupadas y se ejecutan juntas. Los bloques son utilizados en lenguajes de programación para definir la estructura de control de flujo, como bucles y condicionales. También son utilizados para delimitar el alcance de variables y funciones. Por ejemplo, en lenguajes como C, C++, Java o JavaScript, los bloques de código están definidos entre llaves {}.

Statement: en programación es una instrucción que realiza una acción específica. Es una unidad básica de ejecución en un programa.

Pointers and nulls: Java works with pointers. This means that the variables don't hold "directly" the data that we are looking for. They have the address (the pointer) of where the data is. If there is no address to any part of the virtual memory the value is **null**.

PC: Personal computer.

Interface: Comunicarse.

GUI: Graphic User Interface.

CLI: Interfaz de la línea de comandos.

High-level and low level abstraction: El nivel alto es un nivel de abstracción muy general y el nivel bajo es una abstracción muy específica.

Procedural programming: Se centra en secuencias de instrucciones para la resolución de problemas, el uso de funciones y el flujo de control. El control de procesos se refiere a gestionar la ejecución de un programa y determinar el orden de ejecución de las instrucciones en función de condiciones y bucles.

Class: Objeto en Java.

Refactorize: Cambiar las variables de un código, para que sea más fácil de leer.

OOP: Object-oriented programming se trata de crear objetos que contengan tanto datos como métodos.

Object: Un fragmento de código incluye un nombre, una serie de datos y una serie de métodos o funciones o pequeños programas que podemos llamar.

Final: Significa que este valor no se cambiará durante la ejecución.

Ejemplo de estático en texto:

UML

- Nombre de la clase
- Variables
- Métodos

Coche
int numero Motor motor
void avanzar() void frenar() void cambiarMarcha() void ponerRadio()

Moto
int numeroRuedas Motor motor
void HacerCaballito -> es un metodo

Vehiculo
Motor motor
void avanzar() void frenar() void cambiarMarcha()

Vehículo es una clase grade(parent) de Coche y Moto

Coche y Moto son Vehículos.

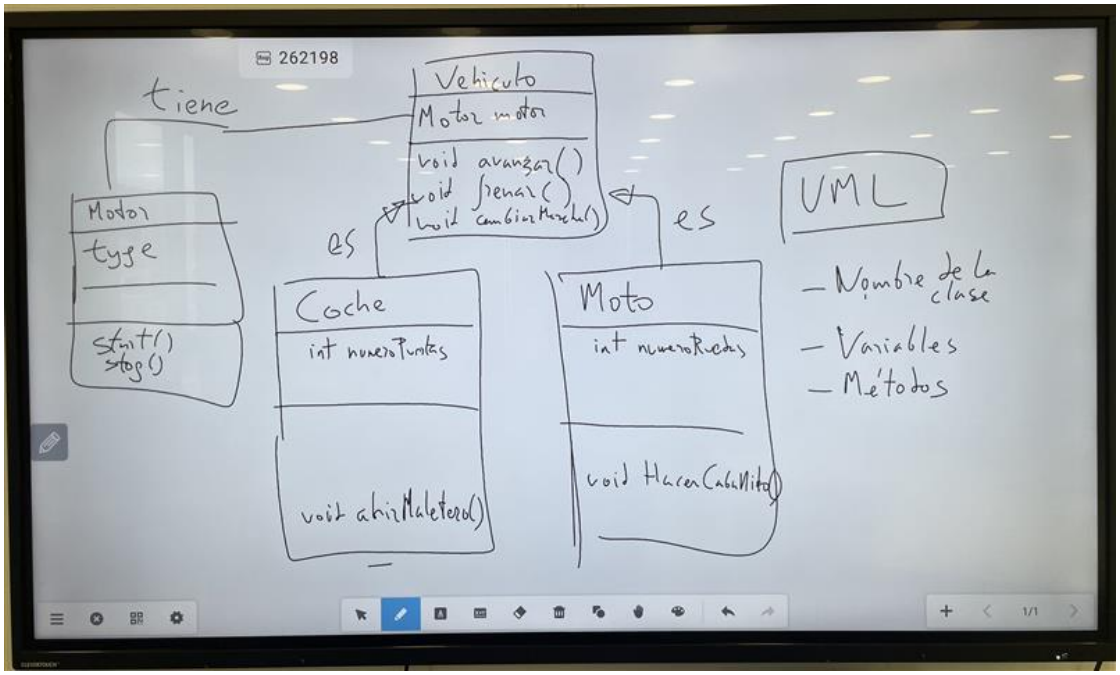
Coche y Moto extienden a Vehículo

Vehículo tiene Motor

Coche y Moto, por herencia, también tienen Motor -> Agregación

Es herencia

Ejemplo de estático en imagen:



¡Enlace a mi GitHub por si queréis consultar algo!

- https://github.com/axckzz/J25-Progamming