# **Programación Orientada a Objetos**

Breve repaso de los paradigmas de la programación:

## **Programación imperativa:**

Se basa en la idea de que el programador describe paso a paso lo que debe hacer el programa para resolver un problema.  
suele estar compuesto de instrucciones básicas, como asignar valores a variables, realizar operaciones matemáticas y controlar el flujo de ejecución.

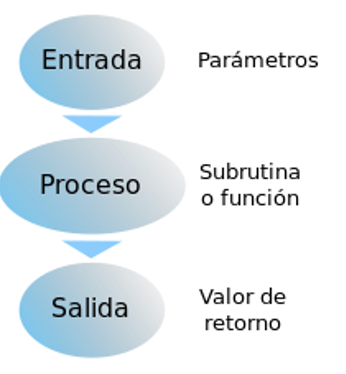
## **Programación estructurada:**

Es cuando planteamos funciones que agrupan actividades a desarrollar y luego dentro del programa llamamos a dichas funciones que pueden estar dentro del mismo archivo (modulo) o en una librería separada.

O, es una forma de programación imperativa, o es un subconjunto de la programación imperativa o tipo de programación imperativa, que utiliza un conjunto limitado de estructuras de control para organizar el flujo de ejecución del programa. Las estructuras de control básicas son:

* Secuencia
* Selección
* Iteración

Las declaraciones se ponen en procedimientos, que se pueden volver a llamar cuando sea necesario.



## **Programación Orientada a objetos:**

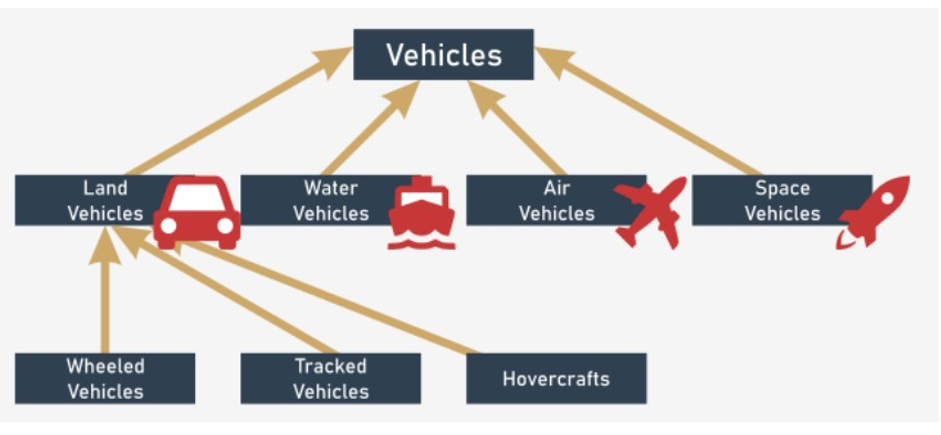
Es un paradigma de programación que basa su estructura en los objetos, que son entidades o elementos que poseen datos y comportamientos, y que pertenecen a la situación que se esta analizando, teniendo una participación fundamental.

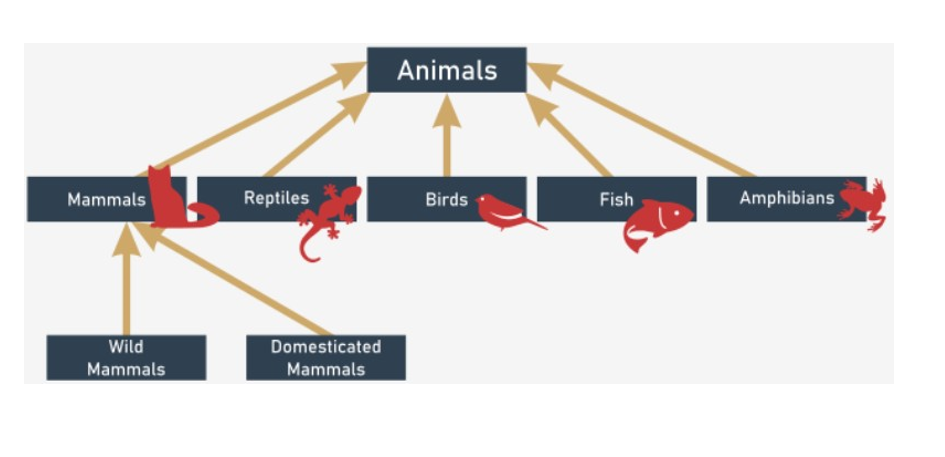
La manera en que los objetos se relacionan e interactúan entre sí, permite identificar la dinámica del contexto.

Los objetos se agrupan en clases, que son plantillas que definen los atributos y métodos que comparten los objetos de una misma clase, además de un mismo objetivo.

**Explicación2:** Es un estilo que trata los datos como objetos con atributos y métodos que pueden aplicarse a estos objetos y también ser heredados por otros objetos.

**Ejemplo1 de herencia:**



**Ejemplo2 de herencia:**

# **Programación Estructurada**

**Ejemplo1:**

clientes= [{"Nombre":"Hector", "Apellidos":"Costa Guzman","Rut":"11111111-1"},

{"Nombre":"Juan","Apellidos":"Gonzales Marquez","Rut":"22222222-2"}]

for cliente in clientes:

print(f'"{cliente["Nombre"]},"{cliente["Apellidos"]}","{cliente["Rut"]}"')

-

def mostrar\_cliente(clientes,dni):

for c in clientes:

if (dni==c["Rut"]):

print("{} {}".format(c["Nombre"],c["Apellidos"]))

return

print("Cliente no encontrado")

mostrar\_cliente(clientes,"11111111-1")

**Ejemplo2:**

clientes= [{"Nombre":"Hector", "Apellidos":"Costa Guzman","Rut":"11111111-1"},

{"Nombre":"Juan","Apellidos":"Gonzales Marquez","Rut":"22222222-2"}]

for cliente in clientes:

print(f'"{cliente["Nombre"]},"{cliente["Apellidos"]}","{cliente["Rut"]}"')

def borrar\_cliente(clientes, dni):

for i,c in enumerate(clientes):

if (dni ==c["Rut"]):

del (clientes[i])

print(str(c),"-->BORRADO")

return

print("Cliente no encontrado")

borrar\_cliente(clientes,"22222222V")

borrar\_cliente(clientes,"22222222-2")

print(clientes)

**Ejemplo3:**

lista=[]

while True:

sabor = input("De que sabor es la galleta?: ")

tamaño = input("De que tamaño en la galleta?: ")

galleta = "Sabor: " + sabor + " Tamaño: " + tamaño

lista.append(galleta)

print()

for i in range(len(lista)):

print(f"galleta {i+1}: {lista[i]})")

**Ejemplo4:**

sabor = "Salada"

chocolate = True

print("Se ha creado la galleta")