UAB
Universitat Autònoma de Barcelona

To POO or not to POOp

WavesLab Workshop – 17-07-25

Aleix Valdivieso Gonzalez

To POO or not to POOp

Struct

- Advantages
  - Easy to use
  - Flexible
- Disadvantages
  - Non-encapsulation
  - Spread logic
  - Difficult maintenance on big projects.
  - No validations.

```
car.brand = 'Toyota';
car.model = 'Corolla';
car.year = 2020;
car.speed = 0;

% Función para acelerar
function car = accelerate(car, amount)
    car.speed = car.speed + amount;
end

% Función para frenar
function car = brake(car, amount)
    car.speed = max(car.speed - amount, 0);
end
```

POO

Structs

```
car.brand = 'Toyota';
car.model = 'Corolla';
car.year = 2020;
car.speed = 0;

% Función para acelerar
function car = accelerate(car, amount)
    car.speed = car.speed + amount;
end

% Función para frenar
function car = brake(car, amount)
    car.speed = max(car.speed - amount, 0);
end
```

POO

```
classdef Car
    properties
        brand
        model
        year
        speed = 0
    end
    methods
        function obj = Car(brand, model, year)
            obj.brand = brand;
            obj.model = model;
            obj.year = year;
        end

        function obj = accelerate(obj, amount)
            obj.speed = obj.speed + amount;
        end

        function obj = brake(obj, amount)
            obj.speed = max(obj.speed - amount, 0);
        end
    end
end
```

```
myCar = Car("Toyota", "Corolla", 2020);
myCar = myCar.accelerate(50);
myCar = myCar.brake(20);
```

POO Inheritance

- Add flexibility and legibility.

```
classdef ElectricCar < Car
    properties
        batteryLevel = 100;  % Porcentaje de batería
    end

    methods
        function obj = ElectricCar(brand, model, year)
            obj@Car(brand, model, year);  % Llamada al constructor padre
        end

        function obj = charge(obj)
            obj.batteryLevel = 100;
            disp('Batería cargada al 100%.');
        end

        function obj = brake(obj, amount)
            brake@Car(obj, amount);  % Llamada al método de la clase padre
            % Solo frena si hay batería suficiente
            regen = 0.2 * amount;  % regeneracion de batería ficticio
            if obj.batteryLevel < 100

                obj.batteryLevel = max(obj.batteryLevel + regen, 0);
            else
                warning('Batería completa. No se puede regenerar.');
            end
        end
    end
end
```

POO vs Structs

POO	Structs
Better scalability and legibility.	High flexibility, uncontrollable on big systems.
Details can be encapsulated (prevent modification of internal details of the code).	All data is accessible (accidental modifications).
Handle possibility: Modify the object parameters without needing to return it on functions.	No Handle: Struct return needed to update its values inside functions.
Unified data and functions.	Data and functions are separated one to each other.
Automatic validations (create restricted properties to assure the correct functioning).	Manual validations.
Inheritance, code reutilization.	

Exercise

- From this STRUCT example, develop its OBJECT homonymous.
- Create 2 new objects, Car, ElectricCar.
- Both need to have return the same outputs as the struct.

```
classdef ClassName < handle
    properties
        Property1
    end
    methods
        function obj = ClassName(inputArg1)
            obj.Property1 = inputArg1^2;
        end
        function return1 = exampleFunction(obj, input)
            return1 = input + obj.Property1;
        end
    end
end
```

```
classdef ClassName_son < ClassName
    properties
        Propertyx
    end
    methods
        function obj = ClassName_son(inputArg1, inputArg2)
            obj@ClassName(inputArg1);
            obj.Propertyx = inputArg2+2;
        end
    end
end
```