OBJECT-ORIENTED PROGRAMMING: PART 1

TBC

📖 PROGRAMMING PARADIGMS

🔧 Imperative

📜 Declarative

📖 IMPERATIVE PARADIGM

- 👨‍💻 Procedural
- 📦 Structural
- ✍️ Aspect Oriented
- 🚗 Object oriented

📖 OBJECT ORIENTED

👷 Agent-oriented

📦 Component-oriented

🗿 Prototype-oriented

TBC

📖 DECLARATIVE PARADIGM

🔧 Functional

⚖️ Logical

📖 OOP

A programming methodology based on the representation of a program as a collection of objects, each of which is an instance of a particular class, and the classes form an inheritance hierarchy.

TBC

 PRINCIPLES OOP

🚗 Abstraction

📦 Encapsulation

🐢 Inheritance

🧙‍♂️ Polymorphism

TBC

📖 CLASS

A universal, complex data type consisting of a thematically single set of “fields” (variables of more elementary types) and “methods” (functions for working with these fields), that is, it is a model of an information entity with internal and external interfaces for operating its content (field values).

TBC

 OBJECT

An entity in the address space of a computing system that appears when a class is instantiated.

TBC

📖 INTERFACE TYPES

⚙️ Inner interface

📞 External interface

📖 INNER INTERFACE

These are properties and methods that can be accessed only from other methods of the object, they are also called “private”.

TBC

📖 EXTERNAL INTERFACE

These are the properties and methods available outside the object, they are called “public".

TBC

📖 CREATE OBJECT

```
1 function Car() {
2   console.log('Car was created');
3 };
4
5 const car = new Car();
6
```

TBC

📖 PUBLIC PROPERTY

```
1 function Car(color) {
2   this.color = color;
3 };
4
5 const car = new Car('red');
6
7 console.log(car.color);
8
```

TBC

📖 PUBLIC METHOD

```
1 function Car(color) {
2   this.color = color;
3
4   this.getColor = function( ) {
5     return this.color;
6   }
7 };
8
9 const car = new Car('red');
10
11 console.log(car.getColor( ));
12
```

TBC

📖 PRIVATE PROPERTY

```
1 function Car(color) {
2   const wheels = 4;
3 };
4
5 const car = new Car('red');
6
7 console.log(car.color);
8 console.log(car.wheels);
9
```

TBC

PRIVATE METHOD

```
1  function Car(color) {
2    let mileage = 0;
3
4    const changeMileage = function() {
5      mileage++
6    }
7
8    this.start = function() {
9      changeMileage();
10   }
11
12   this.getMileage = function() {
13     return mileage;
14   }
15 };
16
17 const car = new Car('red');
18 car.start();
19 console.log(car.getMileage());
```

TBC

HERE GOES!

TBC