Crie uma fábrica de veículos utilizando o padrão Prototype com base no exemplo apresentado no Hipertexto 2. Os requisitos do projeto devem ser:

- implemente uma classe abstrata **Veículo**com um construtor padrão e os métodos *clone*e *represent*;

- o construtor da classe **Veículo**deve receber **modelo**, **marca**, **cor**e **numeroRodas**como parâmetros;

- crie pelo menos duas subclasses da classe **Veículo**, que acrescentem um ou mais atributos, por exemplo: carro que tem dois campos a mais, como **numeroRodas**e **numeroPortas**;

- desenvolva uma classe **Aplicação**que deve criar um *array*com seis veículos com dois tipos de veículos diferentes (subclasses), utilizando o método clone dos objetos para preencher o *array*;

- na classe **Aplicação**, implemente um método que retorne um *array*com o mesmo tamanho do *array*de veículos, onde cada elemento deve ser um clone dos elementos do *array*veículos;

- no final, deve imprimir na tela o retorno da função *represent*de cada elemento desse novo *array*de clones de veículos.

class Veiculo{

    constructor(modelo, marca, cor, numeroRodas){

        this.modelo = modelo;

        this.modelo = marca;

        this.cor = cor;

        this.numeroRodas = numeroRodas;

    }

    clone(obj){

        obj.\_\_proto\_\_;

    }

    represent(){

        console.log(`Modelo: ${this.modelo}, Marca: ${this.marca}, Cor: ${this.cor}, Rodas: ${this.numeroRodas}`);

    }

}

class Carro extends Veiculo{

    constructor(modelo, marca, cor, numeroRodas){

        super(modelo, marca, cor, numeroRodas)

        this.numeroPortas = this.numeroPortas

        let cloneCarro = {

            modelo: this.modelo,

            marca: this.marca,

            cor: this.cor,

            numeroRodas: this.numeroRodas,

            numeroPortas: this.numeroPortas

        }

        Veiculo.prototype.clone = function(obj){

            obj.prototype = cloneCarro

        }

        Veiculo.prototype.represent = function(){

            console.log(`Modelo: ${this.modelo}, Marca: ${this.marca}, Cor: ${this.cor}, Rodas: ${this.numeroRodas}`)

        }

    }

}

class Onibus extends Veiculo{

    constructor(modelo, marca, cor, numeroRodas){

        super(modelo, marca, cor, numeroRodas)

        this.numeroPortas = this.numeroPortas

        let cloneBike = {

            modelo: this.modelo,

            marca: this.marca,

            cor: this.cor,

            numeroRodas: this.numeroRodas,

            numeroPortas: this.numeroPortas

        }

        Veiculo.prototype.clone = function(obj){

            obj.prototype = cloneOnibus

        }

        Veiculo.prototype.represent = function(){

            console.log(`Modelo: ${this.modelo}, Marca: ${this.marca}, Cor: ${this.cor}, Rodas: ${this.numeroRodas}, ${this.numeroPortas}`)

        }

    }

}

let jeep = new Carro('Renegade','Jeep','Vermelho', 4, 4)

jeep.represent()

jeep.clone(jeep)

let chevrolet = new Carro('Onix','Chevrolet','Branco', 4, 4)

chevrolet.represent()

chevrolet.clone(chevrolet)

let Honda = new Carro('Accord','Honda','Preto', 4, 4)

Honda.represent()

Honda.clone(Honda)

let Mercedes = new Carro('O-400RSD','Mercedes','Branco', 4, 1)

Mercedes.represent()

Mercedes.clone(Mercedes)

let fiat = new Carro('Uno','Fiat','Prata', 4, 2)

fiat.represent()

fiat.clone(fiat)

let nissan = new Carro('Civilian','Nissan','Preto', 4, 1)

nissan.represent()

nissan.clone(nissan)

const veiculos = []

veiculos.push(jeep)

veiculos.push(chevrolet)

veiculos.push(Honda)

veiculos.push(Mercedes)

veiculos.push(fiat)

veiculos.push(nissan)