**Aluno:** Anderson José dos Santos(back-end – terça/quinta-noite)

Crie uma fábrica de veículos utilizando o padrão Prototype com base no exemplo apresentando 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;

-cire 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 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.

Reposta em JavaScript:

// Classe abstrata Veículo

class Veiculo {

    constructor(*modelo*, *marca*, *cor*, *numeroRodas*) {

*this*.modelo = *modelo*;

*this*.marca = *marca*;

*this*.cor = *cor*;

*this*.numeroRodas = *numeroRodas*;

    }

    clone() {

      return *Object*.create(*this*);

    }

    represent() {

      return `Modelo: ${*this*.modelo}, Marca: ${*this*.marca}, Cor: ${*this*.cor}, Número de Rodas: ${*this*.numeroRodas}`;

    }

  }

  // Subclasse Carro que herda de Veículo

  class Carro extends Veiculo {

    constructor(*modelo*, *marca*, *cor*, *numeroRodas*, *numeroPortas*) {

*super*(*modelo*, *marca*, *cor*, *numeroRodas*);

*this*.numeroPortas = *numeroPortas*;

    }

    represent() {

      return `${*super*.represent()}, Número de Portas: ${*this*.numeroPortas}`;

    }

  }

  // Subclasse Moto que herda de Veículo

  class Moto extends Veiculo {

    constructor(*modelo*, *marca*, *cor*, *numeroRodas*, *cilindrada*) {

*super*(*modelo*, *marca*, *cor*, *numeroRodas*);

*this*.cilindrada = *cilindrada*;

    }

    represent() {

      return `${*super*.represent()}, Cilindrada: ${*this*.cilindrada}`;

    }

  }

  // Classe Aplicação

  class Aplicacao {

    static criarVeiculos() {

      const veiculo1 = **new** Carro('Sedan', 'Toyota', 'Preto', 4, 4);

      const veiculo2 = **new** Carro('SUV', 'Ford', 'Branco', 4, 2);

      const veiculo3 = **new** Moto('Esportiva', 'Honda', 'Vermelho', 2, '600cc');

      const veiculo4 = **new** Carro('Hatchback', 'Volkswagen', 'Azul', 4, 5);

      const veiculo5 = **new** Moto('Custom', 'Harley-Davidson', 'Preto', 2, '1200cc');

      const veiculo6 = **new** Carro('SUV', 'Jeep', 'Verde', 4, 4);

      return [veiculo1, veiculo2, veiculo3, veiculo4, veiculo5, veiculo6];

    }

    static clonarVeiculos(*veiculos*) {

      return *veiculos*.map((*veiculo*) => *veiculo*.clone());

    }

    static imprimirVeiculos(*veiculos*) {

*veiculos*.forEach((*veiculo*) => {

        console.log(*veiculo*.represent());

      });

    }

  }

  // Uso da classe Aplicação

  const veiculosOriginais = Aplicacao.criarVeiculos();

  const veiculosClonados = Aplicacao.clonarVeiculos(veiculosOriginais);

  Aplicacao.imprimirVeiculos(veiculosClonados);