/\*

desenvolva um código  para criar uma fábrica de veículos utilizando o padrão Prototype em javascript, com os seguintes requisitos:

1- implemente uma classe abstrata Veículos com  um construtor padrão e os métodos clone e represent.

2- o construtor da classe veículos deve receber os parâmetros: modelo, marca, cor e numeroRodas.

3- crie pelo menos duas subclasses da classe veículos, que acrescentem um ou mais atributos, por exemplo:

carro que tem dois campos a mais, como numeroRodas e numeroPortas.

4- 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.

5- na classe aplicação, implemente um mpetodo que retorne um array com o mesmo tamanho do array de veículos,

onde cada elemento deve ser um clone dos elementos do array veículos.

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

\*/

// Classe abstrata Veículos

class Veiculo {

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

*this*.modelo = *modelo*;

*this*.marca = *marca*;

*this*.cor = *cor*;

*this*.numeroRodas = *numeroRodas*;

    }

    clone() {

      const clone = *Object*.create(*this*);

      return clone;

    }

    represent() {

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

    }

  }

  // Subclasse Carro

  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

  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}`;

    }

  }

  class Caminhao extends Veiculo {

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

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

*this*.tracao = *tracao*;

    }

    represent(){

        return `${*super*.represent()}, Tracao: ${*this*.tracao}`;

    }

  }

  // Classe de aplicação

  class Aplicacao {

    constructor() {

*this*.veiculos = [];

    }

    criarVeiculos() {

      const carro1 = **new** Carro('Sedan', 'Toyota', 'Azul', 4, 4);

      const carro2 = **new** Carro('SUV', 'Honda', 'Vermelho', 4, 5);

      const moto1 = **new** Moto('Sport', 'Yamaha', 'Preto', 2, '600cc');

      const moto2 = **new** Moto('Cruiser', 'Harley-Davidson', 'Prata', 2, '1200cc');

      const caminhao1 = **new** Caminhao('Cavalo Mecânico', 'Scania', 'Azul Marinho', 2, 'Trucado');

      const caminhao2 = **new** Caminhao('Mercedinha', 'Mercedes Benz', 'Vermelho', 2, 'Toco');

*this*.veiculos.push(carro1, carro2, moto1, moto2, caminhao1, caminhao2);

      const veiculosClonados = *this*.cloneVeiculos();

      for (const veiculo of veiculosClonados) {

        console.log(veiculo.represent());

      }

    }

    cloneVeiculos() {

      const clonedVeiculos = [];

      for (const veiculo of *this*.veiculos) {

        const clone = veiculo.clone();

        clonedVeiculos.push(clone);

      }

      return clonedVeiculos;

    }

  }

  // Instância da aplicação

  const app = **new** Aplicacao();

  app.criarVeiculos();