

Métodos com Parâmetros

Carro

```
+Marca: String
+Cor: String
+Placa: String
+Velocidade_atual: Inteiro
+Marcha_atual: Caractere
+Freio_de_mao_puxado: booleano
+chave_virada: booleano

+Ligar()
+Acelerar_ate(velocidade: Inteiro)
+Mudar_Marcha(marcha: Inteiro)
+Parar()
```

Métodos com Retorno

Conta
+numero: int +saldo: double +limite: double +nome: String
+saca(valor: double): boolean +deposita(valor: double) +transfere(destino: Conta, valor: double)

MÉTODO toString()

```
1 public class Empregado {  
2     private String nome;  
3     private int idade;  
4  
5     public Employee(String nome, int idade) {  
6         this.nome = nome;  
7         this.idade = idade;  
8     }  
9  
10    @Override  
11    public String toString() {  
12        return this.nome + ": " + this.idade;  
13    }  
14 }
```

SOBRECARGA DE MÉTODOS

Soma
<pre>+soma(x:int,y:int): int +soma(x:string,y:string): string +soma(x:double,y:double): double</pre>

Exemplo

Criar a classe Carro, com as informações do modelo, ano, cor, ligado, velocidade_maxima, velocidade_atual.

Devemos ter os métodos ligar, acelerar(velocidade), desligar e desacelerar(velocidade).

ligar vai alterar o atributo ligado para True.

acelerar vai receber a velocidade, será acelerado apenas com o carro ligado e se a velocidade não ultrapassar o máximo.

desligar vai alterar o atributo ligado para false.

Desacelerar vai receber a velocidade, será desacelerado com o carro ligado e se a velocidade for menor que a velocidade atual.

Exercício

Criar a classe Jogador, terá os atributos força e nome.

Teremos os métodos chute, soco, beberAgua e comerPao.

O método chute vai utilizar 10 pontos de força.

O método soco vai utilizar 5 pontos de força.

O método beberAgua, vai adquirir 15 pontos de força.

O método comerPao, vai adquirir 30 pontos de força.

Crie uma aplicação que permite criar o jogador e utilizar os métodos indicados.