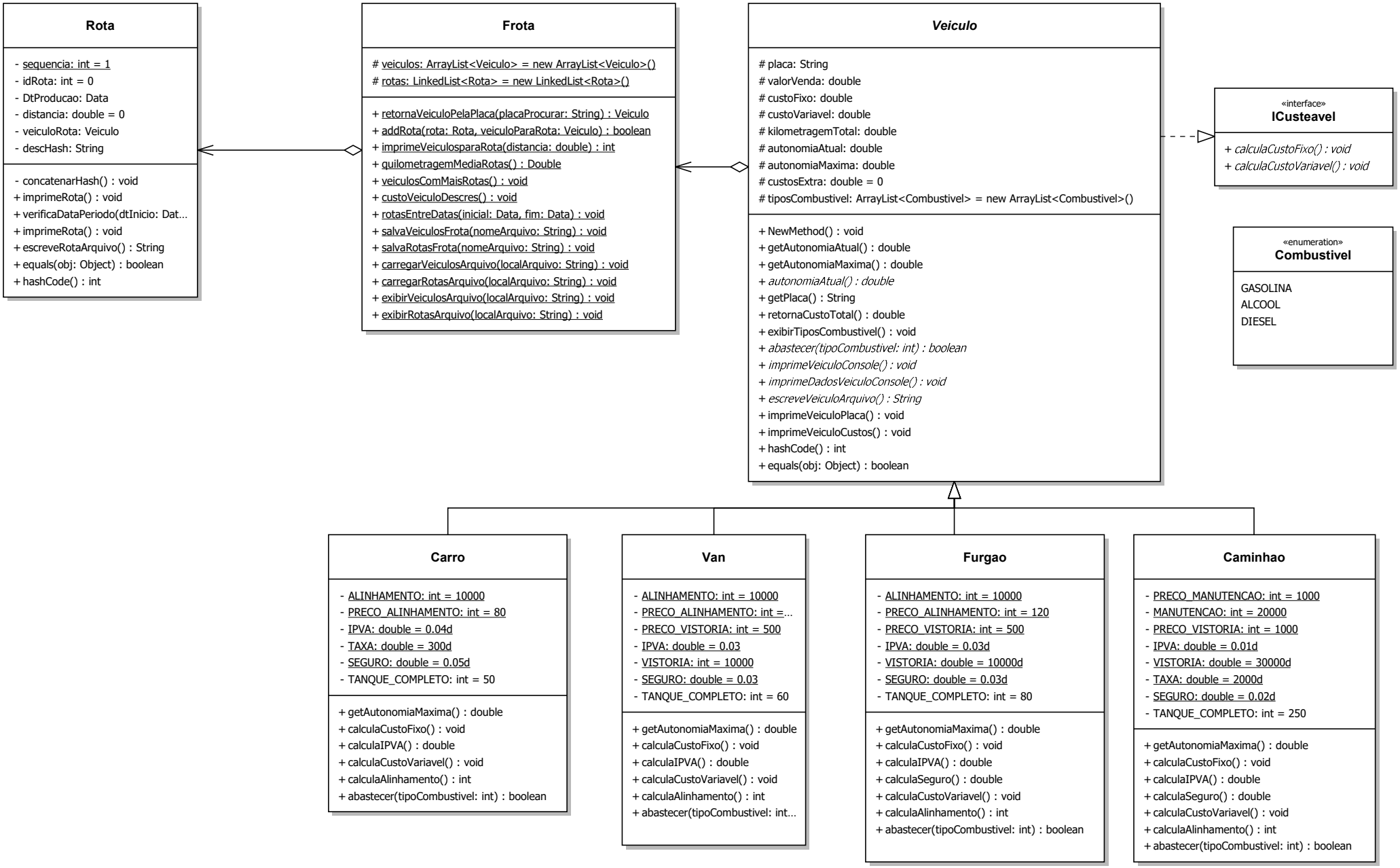
Rota
- sequencia: int = 1
- idRota: int = 0
- DtProducao: Data
- distancia: double = 0
- veiculoRota: Veiculo
- descHash: String
- concatenarHash() : void
+ imprimeRota() : void
+ verificaDataPeriodo(dtInicio: Dat…
+ imprimeRota() : void
+ escreveRotaArquivo() : String
+ equals(obj: Object) : boolean
+ hashCode() : int
Frota
veiculos: ArrayList<Veiculo> = new ArrayList<Veiculo>()
rotas: LinkedList<Rota> = new LinkedList<Rota>()
+ retornaVeiculoPelaPlaca(placaProcurar: String) : Veiculo
+ addRota(rota: Rota, veiculoParaRota: Veiculo) : boolean
+ imprimeVeiculosparaRota(distancia: double) : int
+ quilometragemMediaRotas() : Double
+ veiculosComMaisRotas() : void
+ custoVeiculoDescres() : void
+ rotasEntreDatas(inicial: Data, fim: Data) : void
+ salvaVeiculosFrota(nomeArquivo: String) : void
+ salvaRotasFrota(nomeArquivo: String) : void
+ carregarVeiculosArquivo(localArquivo: String) : void
+ carregarRotasArquivo(localArquivo: String) : void
+ exibirVeiculosArquivo(localArquivo: String) : void
+ exibirRotasArquivo(localArquivo: String) : void
Veiculo
placa: String
valorVenda: double
custoFixo: double
custoVariavel: double
kilometragemTotal: double
autonomiaAtual: double
autonomiaMaxima: double
custosExtra: double = 0
tiposCombustivel: ArrayList<Combustivel> = new ArrayList<Combustivel>()
+ NewMethod() : void
+ getAutonomiaAtual() : double
+ getAutonomiaMaxima() : double
+ autonomiaAtual() : double
+ getPlaca() : String
+ retornaCustoTotal() : double
+ exibirTiposCombustivel() : void
+ abastecer(tipoCombustivel: int) : boolean
+ imprimeVeiculoConsole() : void
+ imprimeDadosVeiculoConsole() : void
+ escreveVeiculoArquivo() : String
+ imprimeVeiculoPlaca() : void
+ imprimeVeiculoCustos() : void
+ hashCode() : int
+ equals(obj: Object) : boolean
«interface»
ICusteavel
+ calculaCustoFixo() : void
+ calculaCustoVariavel() : void
«enumeration»
Combustivel
GASOLINA
ALCOOL
DIESEL
Carro
- ALINHAMENTO: int = 10000
- PRECO_ALINHAMENTO: int = 80
- IPVA: double = 0.04d
- TAXA: double = 300d
- SEGURO: double = 0.05d
- TANQUE_COMPLETO: int = 50
+ getAutonomiaMaxima() : double
+ calculaCustoFixo() : void
+ calculaIPVA() : double
+ calculaCustoVariavel() : void
+ calculaAlinhamento() : int
+ abastecer(tipoCombustivel: int) : boolean
Van
- ALINHAMENTO: int = 10000
- PRECO_ALINHAMENTO: int =…
- PRECO_VISTORIA: int = 500
- IPVA: double = 0.03
- VISTORIA: int = 10000
- SEGURO: double = 0.03
- TANQUE_COMPLETO: int = 60
+ getAutonomiaMaxima() : double
+ calculaCustoFixo() : void
+ calculaIPVA() : double
+ calculaCustoVariavel() : void
+ calculaAlinhamento() : int
+ abastecer(tipoCombustivel: int…
Furgao
- ALINHAMENTO: int = 10000
- PRECO_ALINHAMENTO: int = 120
- PRECO_VISTORIA: int = 500
- IPVA: double = 0.03d
- VISTORIA: double = 10000d
- SEGURO: double = 0.03d
- TANQUE_COMPLETO: int = 80
+ getAutonomiaMaxima() : double
+ calculaCustoFixo() : void
+ calculaIPVA() : double
+ calculaSeguro() : double
+ calculaCustoVariavel() : void
+ calculaAlinhamento() : int
+ abastecer(tipoCombustivel: int) : boolean
Caminhao
- PRECO_MANUTENCAO: int = 1000
- MANUTENCAO: int = 20000
- PRECO_VISTORIA: int = 1000
- IPVA: double = 0.01d
- VISTORIA: double = 30000d
- TAXA: double = 2000d
- SEGURO: double = 0.02d
- TANQUE_COMPLETO: int = 250
+ getAutonomiaMaxima() : double
+ calculaCustoFixo() : void
+ calculaIPVA() : double
+ calculaSeguro() : double
+ calculaCustoVariavel() : void
+ calculaAlinhamento() : int
+ abastecer(tipoCombustivel: int) : boolean