MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO TRIÂNGULO MINEIRO
CAMPUS UBERLÂNDIA CENTRO

TECNOLOGIA EM SISTEMAS PARA INTERNET
Turma: **3º PERÍODO**
Unidade Curricular: **PROGRAMAÇÃO ORIENTADA A OBJETOS**
Professor: **WILL ROGER PEREIRA**

LISTA 2-4

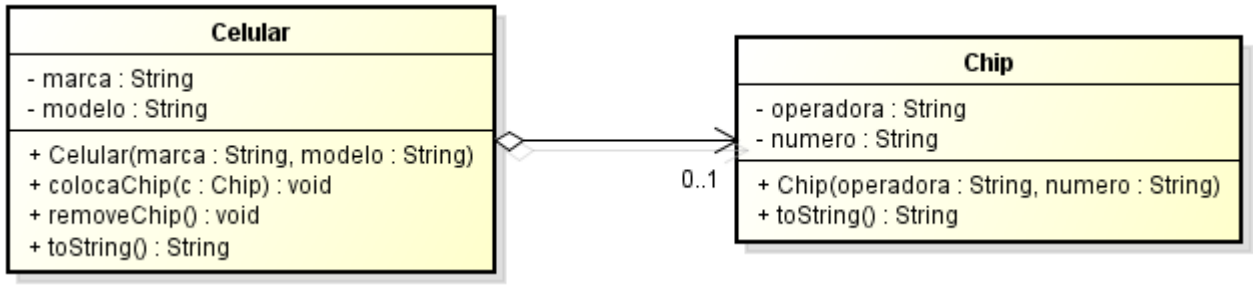
Obs: Para todos os exercícios, proceda conforme a aula. Construa objetos, contemple a multiplicidade e execute os métodos. Caso a multiplicidade contemplar um número definido

Obs2: As especificações e/ou restrições para os valores dos atributos sempre se encontrarão neles!!! Caso este valor esteja fora das especificações dentro de um método, sempre mostre uma mensagem de erro. No caso dos construtores, caso aconteça algum problema com os atributos, atribua valores padrões.

Obs3: O levantamento de restrições também é de sua responsabilidade. Portanto, sempre que encontrar alguma irregularidade na execução de um método, informe este erro.

Obs4: LEIA, NA ÍNTEGRA, A DESCRIÇÃO DE TODOS OS ATRIBUTOS E MÉTODOS.

1ª Questão



Classe Chip:
- operadora : String → Operadora do Chip. Não pode ser uma String vazia.
- numero : String → Número do Chip. Deve ter 10 caracteres numéricos.

+ Chip(operadora : String, numero : String) → Construtor.

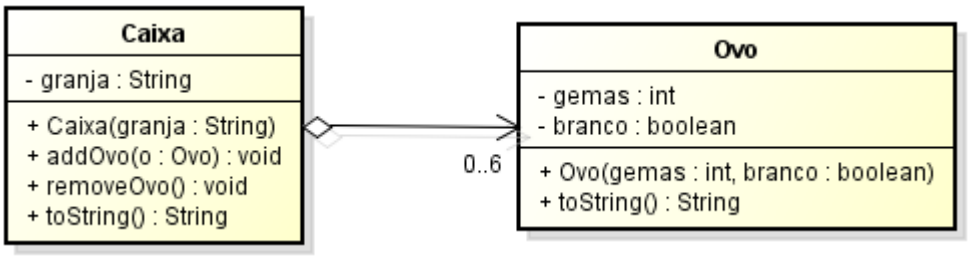
+ toString() : String → Retorna as informações do Chip, para ser mostrado na tela.

Classe Celular:
- marca : String → Marca do Celular. Não pode ser uma String vazia.
- modelo : String → Modelo do Celular. Não pode ser uma String vazia.

+ Celular(marca : String, modelo : String) → Construtor.

+ colocaChip(c : Chip) : void → Coloca um Chip no Celular. Só pode colocar Chip em um Celular sem Chip.
+ removeChip() : void → Remove um Chip do Celular. Só pode retirar Chip de Celular com Chip.
+ toString() : String → Retorna as informações do Celular e de seu Chip, para ser mostrado na tela.

2ª Questão



Classe Ovo:
- gemas : int → Quantidade de gemas no Ovo. Deve ser maior ou igual a 1.
- branco : boolean →Será **true** se o Ovo for branco e **false** se for vermelho.

+ Ovo(gemas : int, branco : boolean) → Construtor.

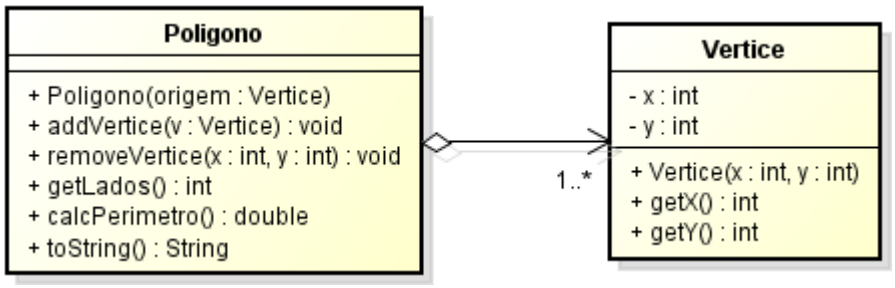
+ toString() : String → Retorna as informações do Ovo, para ser mostrado na tela.

Classe Caixa:
- granja : String → Granja que produz a Caixa. Não pode ser uma String vazia.

+ Caixa(granja : String) → Construtor.

+ addOvo(o : Ovo) : void → Adiciona um Ovo na Caixa. Controle a multiplicidade.
+ removeOvo() : void → Remove o último Ovo da Caixa. Só pode retirar Ovo de Caixa que possuir Ovos.
+ toString() : String → Retorna as informações da Caixa e dos Ovos que fazem parte dela.

3ª Questão



Classe Vertice:
- x : int → Coordenada abscissa do Vertice.
- y : int → Coordenada ordenada do Vertice.

+ Vertice(x : int, y : int) → Construtor.

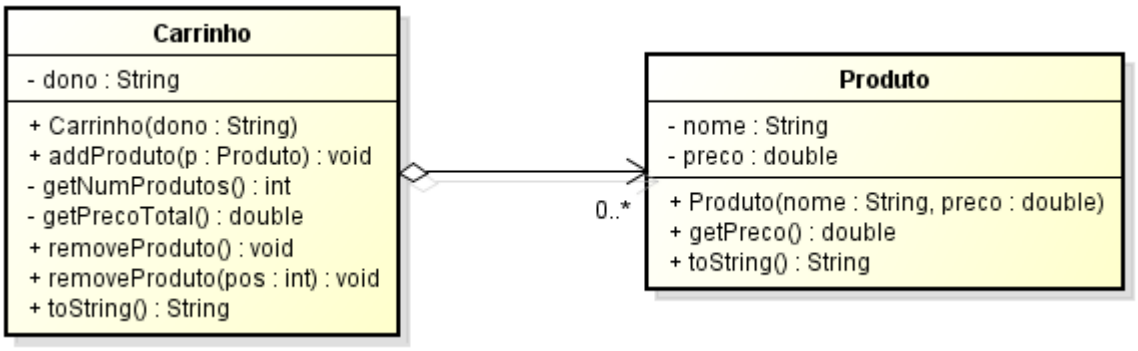
+ getX() : int → Retorna a coordenada abscissa do Vertice.
+ getY() : int → Retorna a coordenada ordenada do Vertice.

Classe Poligono:
- nome : String → Nome do Poligono aberto.

+ Poligono(nome : String, origem : Vertice) → Construtor.

+ addVertice(v : Vertice) : void → Adiciona um Vertice ao conjunto de Vertices do Poligono.
+ removeVertice(x : int, y : int) : void → Remove um Vertice, que possuir os determinados atributos, do conjunto de Vertices do Poligono, respeitando o mínimo da multiplicidade.
+ getLados() : int → Retorna a quantidade de lados do Poligono.
+ calcPerimetro() : double → Retorna o perímetro do Poligono.
+ toString() : String → Retorna as informações do Polígono. Além dos atributos, retorne a quantidade de lados e seu perímetro.

4ª Questão



Classe Produto:
- nome : String → Nome do Produto. Não pode ser uma String vazia.
- preco : double → Custo para adquirir o Produto. Deve ser um valor positivo.

+ Produto(nome : String, preco : double) → Construtor.

+ toString() : String → Retorna as informações do Produto: nome e preço.
+ getPreco() : Double → Retorna o preço do Produto.

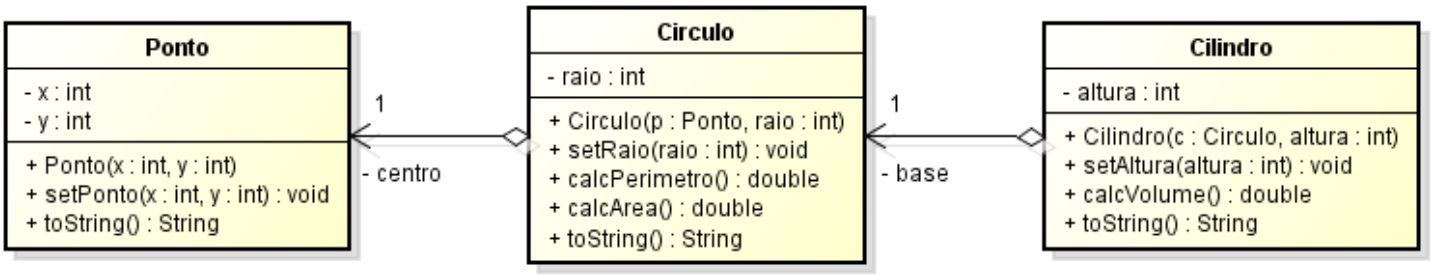
Classe Carrinho:
- dono : String → Dono do Carrinho. Não pode ser uma String vazia.

+ Carrinho(dono : String) → Construtor.

+ addProduto (p : Produto) : void → Adiciona um Produto ao carrinho.
- getNumProdutos () : int → Retorna o número de produtos dentro do Carrinho.
- getPrecoTotal () : double → Retorna a soma dos preços dos produtos dentro do Carrinho.
+ removeProduto() : void → Remove o último Produto adicionado.
+ removeProduto(pos : int) : void → Remove o Produto no índice do argumento.
+ toString () : String → Retorna as informações do Carrinho: O dono, o preço total, além do número e da informação de todos os Produtos dentro dele

---

5ª Questão



Classe Ponto:
- x : int → Coordenada abscissa do ponto.
- y : int → Coordenada ordenada do ponto.

+ Ponto(x : int, y : int) → Construtor.

+ setPonto(x : int, y : int) : void → Muda as coordenadas x e y baseadas nos respectivos argumentos.
+ toString() : String → Retorna as informações do ponto, coordenadas x e y.

Classe Circulo:
- raio : int → Raio do circulo. Deve ser um valor positivo.

+ Circulo(p : Ponto, raio : int) → Construtor.

+ setRaio(raio : int) : void → Muda o raio do circulo, baseado no argumento.
+ calcArea() : double → Retorna a área do circulo.
+ calcPerimetro() : double → Retorna o perímetro do círculo.
+ toString() : String → Mostra as informações do círculo, o raio, a área, o perímetro, e as informações do Ponto agregado.

Classe Cilindro:
- altura : int → Altura do cilindro. Deve ser um valor positivo.

+ Cilindro(c : Circulo, altura : int) → Construtor.

+ setAltura(altura : int) : void → Muda a altura do Cilindro, baseado no argumento.
+ calcVolume() : double → Calcula o volume do Cilindro. Qual será a área da base?
+ toString() : String → Mostra as informações do Cilindro, a altura, o volume, e as informações do Circulo agregado.

Complemento:
Crie um Cilindro, com um Ponto e um Circulo. Mostre suas informações.
Modifique as coordenadas do Ponto e o raio do Circulo. Mostre novamente as informações.
Alguma coisa mudou? Por que?
