

TECNOLOGIA EM SISTEMAS PARA INTERNET

Turma: **3º PERÍODO**

Unidade Curricular: **PROGRAMAÇÃO ORIENTADA A OBJETOS**

Professor: **WILL ROGER PEREIRA**

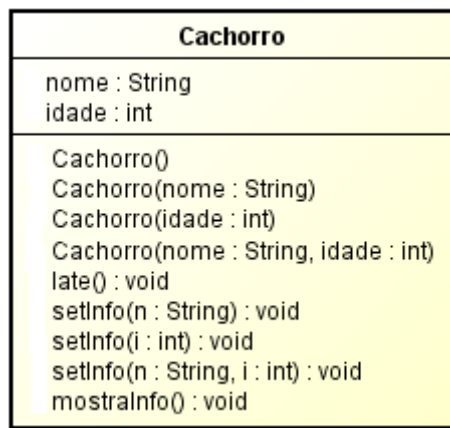
LISTA 1-8

**Obs:** Para todos os exercícios, crie pelo menos 2 (dois) objetos, inicialize os atributos utilizando os construtores (exceto quando especificado no diagrama), e execute todos os métodos para demonstrar suas funcionalidades.

**Obs2:** As especificações e/ou restrições para os valores dos atributos sempre se encontrarão neles!!! Caso este valor esteja fora das especificações, sempre mostre uma mensagem de erro.

**Obs3:** No caso dos construtores, caso aconteça algum problema com os atributos, atribua valores padrões.

**1ª Questão**



**Classe Cachorro:**

nome : String → Nome do Cachorro.

idade : int → Idade do Cachorro. Deve ser um valor Natural.

Cachorro() → Defina valores padrões para os atributos.

Cachorro(nome : String) → Defina um valor padrão para a idade.

Cachorro(idade : int) → Defina um valor padrão para o nome. Atente-se à restrição da idade.

Cachorro(nome : String, idade : int) → Atribua os valores e atente-se às restrições.

late() : void → Deve mostrar na tela que o Cachorro latiu uma vez. O latido será “au au”.

setInfo(n : String) : void → Modifica o nome de acordo com o argumento.

setInfo(i : int) : void → Modifica a idade de acordo com o argumento.

setInfo(n : String, i : int) : void → Modifica os dois atributos do objeto. Utilize os métodos responsáveis para mudar o valor dos atributos, setInfo com o tipo respectivo.

mostraInfo() : void → Mostra na tela as informações do Cachorro.

## 2ª Questão

Retangulo
lado1 : int lado2 : int
Retangulo() Retangulo(lado1 : int, lado2 : int) setLados(lado1 : int, lado2 : int) : void calcArea() : int calcPerimetro() : int ehQuadrado() : boolean mostraInfo() : void

### Classe Retangulo:

lado1 : int → Lado horizontal. Deve ser um valor positivo.

lado2 : int → Lado vertical. Deve ser um valor positivo.

Retangulo() → Construtor de aridade 0.

Retangulo(lado1 : int, lado2 : int) → Construtor completo.

setLados(lado1 : int, lado2 : int) : void → Modifica os lados do retângulo, baseado respectivamente nos argumentos do método.

calcArea() : int → Retorna a área do Retangulo.

calcPerimetro() : int → Retorna o perímetro do Retangulo.

ehQuadrado() : boolean → Retorna **true** se o Retangulo for um quadrado. Caso contrário, retorna false. Pesquise sobre o quadrado ser um tipo especial de retângulo.

mostraInfo() : void → Mostra as informações do Retangulo na tela. Além dos atributos, mostra sua área, perímetro e se este retângulo é ou não um quadrado.

---

## 3ª Questão

Triangulo
lado1 : int lado2 : int lado3 : int
Triangulo() Triangulo(lado1 : int, lado2 : int, lado3 : int) setLados(lado1 : int, lado2 : int, lado3 : int) : void getTipo() : String mostraInfo() : void

### Classe Triangulo:

lado1 : int → Lado do triângulo. Deve ser um valor positivo.

lado2 : int → Lado do triângulo. Deve ser um valor positivo.

lado3 : int → Lado do triângulo. Deve ser um valor positivo.

Triangulo() → Construtor de aridade 0.

Triangulo(lado1 : int, lado2 : int, lado3 : int) → Construtor completo.

setLados(lado1 : int, l2 : int, l3 : int) : void → Modifica os lados do Triangulo, baseado respectivamente nos argumentos do método.

getTipo() : String → Retorna se o Triangulo é equilátero, isósceles ou escaleno. Pesquise sobre os tipos para confeccionar o método.

mostraInfo() : void → Mostra as informações do Triangulo na tela. Além dos atributos, mostra sua área e seu tipo.

---

#### 4ª Questão

Circulo
raio : double
Circulo() Circulo(raio : double) setRaio(raio : double) : void calcPerimetro() : double calcArea() : double mostraInfo() : void

##### Classe Circulo:

raio : double → Raio do Circulo. Deve ser um valor positivo.

Circulo() → Construtor de aridade 0.

Circulo(raio : double) → Construtor completo.

setRaio(raio : double) : void → Modifica o raio do Circulo, baseado no argumento do método. Deve ser um valor natural. Caso contrário, mostre uma mensagem de erro.

calcArea() : double → Retorna a área do Circulo.

calcPerimetro() : double → Retorna o perímetro do Circulo.

mostraInfo() : void → Mostra as informações do Circulo na tela. Além dos atributos, mostra sua área e perímetro.

---

#### 5ª Questão

CarrinhoDeSupermercado
dono : String produtos : int = 0 lista : String = "" valorcomprado : double = 0.0
CarrinhoDeSupermercado() CarrinhoDeSupermercado(dono : String) adicionaProduto(nome : String, valor : double) : void limpa() : void mostraInfo() : void

##### Classe CarrinhoDeSupermercado:

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

produtos : int = 0 → Quantidade de produtos no carrinho. O carrinho começa vazio.

lista : String = "" → Lista com o nome e preço dos produtos presentes no carrinho.

valorcomprado : double = 0.0 → Valor dos produtos presentes no carrinho.

CarrinhoDeSupermercado() → Construtor de aridade 0.

CarrinhoDeSupermercado(dono : String) → Construtor completo.

adicionaProduto(nome : String, valor : double) : void → Adiciona o produto de acordo com os argumentos. Modifique os atributos corretos, de forma a comportar este produto no CarrinhoDeSupermercado. O nome não pode ser uma String vazia e o valor deve ser positivo. Não se esqueça de adicionar o produto à lista, adicionando um “\n” entre cada um, para aprimorar a impressão na tela.

limpa() : void → Limpa o CarrinhoDeSupermercado, retirando todos os produtos e colocando-o em seu estado inicial.

mostraInfo() : void → Mostra as informações do CarrinhoDeSupermercado na tela.

---

## 6ª Questão

RadioRelogio
horas : int minutos : int estacao : int ligado : boolean = false
RadioRelogio() RadioRelogio(horas : int, minutos : int, estacao : int) avancaHora() : void avancaMinuto() : void avancaMinuto(minutos : int) : void liga() : void desliga() : void trocaEstacao(estacao : int) : void mostraInfo() : void

### Classe RadioRelogio:

horas : int → Horas atuais. Deve ser um valor entre 0 e 24.

minutos : int → Minutos atuais. Deve ser um valor entre 0 e 59.

estacao : int → Estação atual. Deve ser um valor entre 87 e 108.

ligado : boolean = false → Define se está ligado ou desligado. O RadioRelogio começa desligado.

RadioRelogio() → Construtor de aridade 0.

RadioRelogio(horas : int, minutos : int, estação : int) → Construtor completo.

avancaHora() : void → Incrementa em 1 a hora do relógio. Faça uma reflexão sobre o comportamento de um Relógio.

avancaMinuto() : void → Incrementa em 1 o minuto do relógio. Faça uma reflexão sobre o comportamento de um Relógio.

avancaMinuto(minutos : int) : void → Incrementa em minutos tempo do relógio. Utilize o método análogo de aridade 0. O avanço deve ser sempre positivo.

liga() : void → Liga o RadioRelogio. Não poderá ligar se já estiver ligado. Mude o atributo correspondente.

desliga() : void → Desliga o RadioRelogio. Não poderá desligar se já estiver desligado. Mude o atributo correspondente.

trocaEstacao(estacao : int) : void → Sintoniza o RadioRelogio em uma estação conforme o argumento. Respeite os limites da estação.

mostraInfo() : void → Mostra as informações do RadioRelogio. Suas horas no formato hora:minuto, com preenchimento de zeros a esquerda (opcional), além de sua estação atual e se ele está ligado.