



TECNOLOGIA EM SISTEMAS PARA INTERNET

Turma: **3º PERÍODO**

Unidade Curricular: **PROGRAMAÇÃO ORIENTADA A OBJETOS**

Professor: **WILL ROGER PEREIRA**

LISTA 1-7

Obs: Para todos os exercícios, crie pelo menos 2 (dois) objetos, inicialize os atributos utilizando os métodos, e execute todos os métodos para demonstrar suas funcionalidades.

1ª Questão

Cachorro
nome : String = "Totó" idade : int = 5
late() : void late(latido : String) : void late(vezes : int) : void late(vezes : int, latido : String) : void setInfo(n : String) : void setInfo(i : int) : void setInfo(n : String, i : int) : void mostraInfo() : void

Classe Cachorro:

nome : String → Nome do Cachorro.

idade : int = 5 → Idade do Cachorro. Deve começar com o valor 5.

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

late(latido : String) : void → Deve mostrar na tela que o Cachorro latiu uma vez. O latido será o texto que estiver no argumento.

late(vezes : int) : void → Deve mostrar na tela que o Cachorro latiu quantas vezes estiver no argumento. Deve ser um valor Natural. Caso contrário, mostre uma mensagem de erro. O latido será "au au".

late(vezes : int, latido : String) : void → Deve mostrar na tela que o Cachorro latiu quantas vezes estiver no primeiro argumento. Deve ser um valor Natural. Caso contrário, mostre uma mensagem de erro. O latido será o texto que estiver no segundo argumento.

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

setInfo(i : int) : void → Modifica a idade de acordo com o argumento. Deve ser um valor Natural. Caso contrário, mostre uma mensagem de erro.

setInfo(n : String, i : int) : void → Modifica os dois atributos do objeto. Utilize os métodos responsáveis para mudar o valor dos atributos, setNome e setIdade respectivamente.

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

2ª Questão

Calculadora
<pre>soma(op1 : int, op2 : int) : int soma(op1 : int, op2 : int, op3 : int) : int soma(op1 : double, op2 : double) : double subtrai(op1 : int, op2 : int) : int subtrai(op1 : int, op2 : int, op3 : int) : int subtrai(op1 : double, op2 : double) : double multiplica(op1 : int, op2 : int) : int multiplica(op1 : double, op2 : double) : double divide(dividendo : int, divisor : int) : double divide(dividendo : double, divisor : double) : double potencia(base : int, exp : int) : int potencia(base : double, exp : int) : double raizQuadrada(op : int) : double raizQuadrada(op : double) : double</pre>

Classe Calculadora:

soma(op1 : int, op2 : int) : int → Retorna a soma dos argumentos.

soma(op1 : int, op2 : int, op3 : int) : int → Retorna a soma dos argumentos.

soma(op1 : double, op2 : double) : double → Retorna a soma dos argumentos.

subtrai(op1 : int, op2 : int) : int → Retorna a diferença dos argumentos.

subtrai(op1 : int, op2 : int, op3 : int) : int → Retorna a diferença dos argumentos.

subtrai(op1 : double, op2 : double, op3 : double) : double → Retorna a diferença dos argumentos.

multiplica(op1 : int, op2 : int) : int → Retorna o produto dos argumentos.

multiplica(op1 : double, op2 : double) : double → Retorna o produto dos argumentos.

divide(dividendo : int, divisor : int) : double → Retorna o quociente da operação. Existe uma exceção para esta operação. Descubra e evite que ela aconteça.

divide(dividendo : double, divisor : double) : double → Retorna o quociente da operação. Existe uma exceção para esta operação. Descubra e evite que ela aconteça.

potencia(base : int, exp : int) : int → Retorna a operação base^{exp} . Faça utilizando laços de repetição.

potencia(base : double, exp : int) : double → Retorna a operação base^{exp} . Faça utilizando laços de repetição.

raizQuadrada(op : int) : double → Retorna a raiz quadrada do argumento. Descubra como fazer esta operação utilizando potenciação. Descubra como fazer a potenciação. O argumento não pode ser negativo, pois o resultado não dará um número complexo. Caso seja, mostre uma mensagem de erro.

raizQuadrada(op : double) : double → Retorna a raiz quadrada do argumento. Descubra como fazer esta operação utilizando potenciação. Descubra como fazer a potenciação. O argumento não pode ser negativo, pois o resultado não dará um número complexo. Caso seja, mostre uma mensagem de erro.
