UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE CRATEÚS

Lista de Exercícios 04

Programação Orientada a Objetos • 2023.1

Breve explicação sobre uso da palavra `this`.

Em Java, a palavra-chave "`this`" é utilizada para fazer referência a um objeto da própria classe, permitindo distinguir entre variáveis locais e membros de uma classe com o mesmo nome.

O "`this`" é especialmente útil em construtores, quando é preciso diferenciar o valor passado como argumento para o construtor do valor do atributo da classe com o mesmo nome. Por exemplo:

```
public class Pessoa {
    private String nome;
    private int idade;

    public Pessoa(String nome, int idade) {
        this.nome = nome; // atribui o valor do argumento ao atributo "nome"
        this.idade = idade; // atribui o valor do argumento ao atributo
"idade"
    }
}
```

Questões sobre Escopo

01. Crie uma classe "`EscopoV`" com um atributo "`numero`" e um método "`calcular`", que receba um parâmetro "`numero`". Dentro do método, crie uma variável local "`resultado`" e atribua a ela o valor da multiplicação entre "numero" e "`this.numero`". Retorne o valor de "`resultado`". Qual é o escopo da variável "`resultado`"?

02. Modifique o método "`calcular`" da questão anterior para que ele crie uma variável local "`numero`" e atribua a ela o valor do parâmetro recebido. Qual é o escopo da variável "`numero`"? Ela tem algum conflito com o atributo "`numero`" da classe?
Responda: O método acima pode ser compilado? Justifique sua resposta.

Questões sobre Encapsulamento

03. Crie uma classe "ContaBancaria" que possua os atributos "saldo" e "limite" e os métodos "getSaldo", "getLimite", "depositar" e "sacar". Implemente o encapsulamento dos atributos e utilize-o nos métodos para garantir que o saldo não fique negativo.

04. Em uma classe "`Cliente`", crie um atributo "`nomeCompleto`" e um método "`setNomeCompleto`". Utilize o encapsulamento para garantir que o nome não ultrapasse 80 caracteres (sem espaços).

Questões sobre Construtores

05. Crie uma classe “`Retangulo`” com dois atributos privados, “`base`” e “`altura`”, ambos do tipo `double`. Implemente dois construtores para a classe: um que não recebe argumentos e inicializa

os atributos com os valores 0.0, e outro que recebe dois argumentos do tipo `double` e inicializa os tributos com estes valores. Crie a classe principal para exibir o tamanho do retângulo.

06. Crie uma classe "`Carro`" com quatro atributos privados, "`marca`", "`modelo`", "`ano`" e "`cor`", todos do tipo `String`. Implemente um construtor que receba os quatro argumentos e inicialize os atributos, e outro construtor que receba apenas a "`marca`" e o "`modelo`", e inicialize o "`ano`" como 0 e a "`cor`" como "`preta`". Crie a classe principal para exibir os valores dos atributos.

Questões sobre Sobrecarga de métodos

07. Crie uma classe "`Calculadora`" com dois métodos sobrecarregados chamados "`somar`". O primeiro método deve receber dois argumentos inteiros e retornar a soma deles, enquanto o segundo método deve receber dois argumentos `double` e retornar a soma deles. Crie a classe principal para testar a execução dos métodos.

08. Crie uma classe "`Relogio`" com três métodos sobrecarregados "`setHora`". O relógio pode ser configurado de três maneiras diferentes:

1. Informando-se hora, minuto e segundo;
2. Informando-se somente a hora e o minuto, e inicializando o segundo com 1;
3. Informando-se somente a hora, e inicializando o minuto e o segundo com 1.

Crie a classe principal para exibir a hora.

****

Orientações para a entrega

- *A entrega deverá ser feita por meio do SIGAA até 27/04/2023 às 23:59.*
- *Apenas os arquivos .java devem ser enviados.*
- *O conjunto dos arquivos .java deve ser compactado em formato .zip*

***Obs**.: Em caso de dificuldade para resolver a lista, solicitem ajuda ao monitor da disciplina.*