Prova técnica JAVA

> O objetivo dessa prova é validar os conhecimentos que os(as) instruores(as) possuem com relação a programação JAVA.

1. Critérios de avaliação

• Código Limpo.

• Automação de testes de unidade.

• Clean Architecture.

2. Entregáveis

• Código Fonte em repositório Git (GitHub ou qualquer outro que possa compartilhar o projeto.)

• Você terá 3 dias para finalizar essa aplicação a partir do momento que você receber esse documento;

• Qualquer dúvida, estaremos a disposição.

Estrutura da aplicação

*Antes de introduzir a estrutura dessa aplicação é importante dizer que **é permitido** realizar pesquisas para atender aos requisitos destes projetos.

Estrutura

Projeto 1

1- Implemente uma classe **Proprietário**
2- Declare os seguintes atributos na classe:

- Nome
- CPF
- RG
- Data de Nascimento
- Rua
- Bairro
- Cidade
- Estado
- Cep
- Complemento

3- Faça o encapsulamento dos atributos da classe **Proprietário**

4- Os atributos *nome*, *cpf e rg* são obrigatórios (crie um construtor com esses parâmetros)

5- Implemente uma classe **Carro**

6- Declare os seguintes atributos na classe:

- Modelo
- Cor
- Ano
- Marca
- Chassi
- Proprietário
- Velocidade máxima
- Velocidade atual
- Número de portas
- Tem teto solar?
- Número de Marchas
- Tem cambio automatico?
- Volume de combustível

7- Faça o encapsulamento da classe **Carro** e seus atributos;
8- Implemente o método *acelera* que aumenta a velocidade de 1 em 1 km/h;
9- Implemente o método *freia* que para o carro – Velocidade = 0 km/h;
10- Implemete o método *troca marcha;*
11- Implemente o método _reduz a marcha;
12- Altere a classe Proprietário para que o atributo **Endereço** vire uma classe;
13- Encapsule os atributos da classe Endereço;

14- O endereço do proprietário não pode ser vazio (altere no construtor para receber o endereço);
15- Todo veículo tem um proprietário obrigatoriamente (implemente um construtor de veículo passando o proprietário como parâmetro);
16- A marcha ré nao pode ser engatada se o a velocidade for superior a 0 KM/h;
17- Implemente um método que calcule a autonomia de viagem do veículo com base no consumo médio passado como parâmetro;
18- Implemente um método para exibir o volume de combustível;
19- Transforme o atributo Marca de um carro em uma classe Marca com nome, nrDeModelos, ano de lançamento e código identificador;
20- Instancie um objeto da classe Carro, Pessoa, Endereço, Marca e relacione os objetos utilizando os métodos ou construtores quando necessário.

Projeto 2

Crie uma classe em Java chamada fatura para uma loja de suprimentos de informática. A classe deve conter quatro variáveis:

- o número (String),
- a descrição (String),
- a quantidade comprada de um item (int)
- e o preço por item (double).

A classe deve ter um construtor e um método get e set para cada variável de instância. Além disso, forneça um método chamado getTotalFatura que calcula o valor da fatura e depois retorna o valor como um double. Se o valor não for positivo, ele deve ser configurado como 0. Se o preço por item não for positivo, ele deve ser configurado como 0.0. Escreva um aplicativo de teste chamado FaturaTeste (em outro arquivo) que demonstra as capacidades da classe Fatura.

Projeto 3

Crie uma classe chamada Empregado que inclui três partes de informações como variáveis de instância:

- nome (String),
- sobrenome (String)
- e um salário mensal (double).

A classe deve ter um construtor, métodos get e set para cada variável de instância.

Escreva um aplicativo de teste chamado EmpregadoTeste que cria dois objetos Empregado e exibe o salário de cada objeto. Então dê a cada Empregado um aumento de 10% e exiba novamente o salário anual de cada Empregado. Introduza na classe Empregado uma variável de classe capaz de contabilizar o número de empregados que passaram pela empresa até a data.

Projeto 4

Crie uma classe em Java chamada InteiroSet. Cada objeto InteiroSet pode armazenar inteiros no intervalo de 0 a 100. O conjunto é representado por um array de booleans. O elemento do array a[i] é true se o inteiro i estiver no conjunto. O elemento do array a[j] é false se o inteiro não estiver no conjunto. O construtor sem argumento inicializa o array Java como 'conjunto vazio' (todos os valores false). Forneça os seguintes métodos:
• Método union cria um terceiro conjunto que é a união teórica de dois conjuntos existentes (isto é, aplicação da função lógica OU sobre os conjuntos e retorna o valor lógico true ou false).

• Método intersecção cria um terceiro conjunto que é a intersecção teórica de dois conjuntos existentes (isto é, aplicação da função lógica AND sobre os conjuntos e retorna o valor lógico true ou false).

• Método insereElemento insere um novo elemento inteiro k em um conjunto (configurando a[k] como true).

• Método deleteElemento exclui o inteiro m (configurando a[m] como false).

• Método toSetString retorna uma string contendo um conjunto como uma lista de números separados por espaço. Inclua somente os elementos que estão presentes no conjunto. Utilize -"para representar um conjunto vazio.

• Método ehIgualTo determina se dois conjuntos são iguais. strutura

Considerações finais

Lembre-se que é permitido pesquisar para realizar qualquer requisito destes projetos, queremos que você fique a vontade para criar essa aplicação da melhor forma possível, mesmo que não consiga fazer todo o projeto, nos envie na data prevista o que você já implementou.

Qualquer dúvida ou questionamento estamos à disposição, seja criativo(a) e boa sorte!