

## Teste I de Programação II Enunciado

Curso: LECC  
Turma: 2º ano  
Docente: Mujahid Karim

Data: 14-Abril-2023  
Duração: 90 Minutos  
Pontuação: 200 pontos

1. Assinale com V as respostas correctas e com F as respostas falsas. [25 Pontos]
- A Programação Orientada a Objectos pode ser entendida como objectos comunicando entre si.
  - Se deseja usar o método da classe situada no pacote **java.io.StringBufferInputStream**, deve-se usar a palavra reservada **package** para ter o acesso aos métodos.
  - Os modificadores de acesso existentes em java são: métodos **gets e sets**.
  - Herança em JAVA é um conceito importante porque permiti-nos herdar todos os comportamentos da super-classe. Juntado a Herança e o Encapsulamento, a Herança torna-se mais potente porque passa também a herdar não só os comportamentos, mas também os atributos.
  - A **sobre-escrita** é um conceito de JAVA que consiste em manter a assinatura do método, mas acrescentando um atributo novo como parâmetro.  
sobrecarga

2. Associe os conceitos abaixo: [25 Pontos]

1	Herança		Não permite que seja realizada a implementação dos metodos, as classes que implementam devem obrigatoriamente implementar seus metodos.
2	Classes Abstractas		E a capacidade de uma classe ter todos os metodos e variaveis de instancia de outra classe sem precisar reescrever o codigo.
3	Interface		E a capacidade de um objecto ser referenciado de varias formas.
4	Polimorfismo		Garante a manutenção do codigo, pois cada alteração deveras ser feita em um local.
5	Encapsulamento		Permite a declaração de metodos e atributos concretos e abstractos, porem não pode ser instanciada directamente.

3. Crie uma classe `Veiculo` com os seguintes atributos: `marca`, `modelo`, `ano` e `preco`. Crie também um método `toString` que retorna uma `String` com as informações do veículo.

Crie uma classe `Carro` que herda de `Veiculo` e acrescente um atributo `qtdPortas`. Crie também um construtor que inicializa os atributos de `Veiculo` e `qtdPortas` e um método `toString` que sobrescreve o método `toString` de `Veiculo` adicionando a quantidade de portas.

Crie uma classe `Locadora` que tenha um método `alugar` que recebe um objeto `Veiculo` e um valor `diaria`. O método deve calcular o valor total do aluguel multiplicando o valor da diária pelo número de dias. Deve ser adicionado um valor de MZN500,00 se o veículo for um carro, e tiver mais de 2 portas.

Na classe principal, crie alguns objetos `Veiculo` e `Carro`, e teste o método `alugar` da classe `Locadora`, passando alguns veículos como parâmetro e exibindo o valor total do aluguel. Certifique-se de que a sobrecarga de métodos é utilizada na classe `Carro` para exibir corretamente o número de portas. **[75 Pontos]**

4. Desenvolva um sistema de gerenciamento de uma livraria e precisa implementar uma classe `Livro` com um método `calcularPreco()` que calcula o preço de venda do livro com base no seu preço de custo.

Além disso, existem diferentes tipos de livros que têm margens de lucro diferentes. Por exemplo, livros de ficção têm uma margem de lucro de 30%, livros de não ficção têm uma margem de lucro de 20%

Aqui estão as especificações da classe `Livro`:

A classe deve ter três atributos protegidos: `titulo`, `precoDeCusto` e `margemDeLucro`.

A classe deve ter um construtor que recebe o título e o preço de custo do livro como argumentos e inicializa os atributos correspondentes. O atributo `margemDeLucro` deve ser inicializado como zero.

A classe deve ter duas subclasses: `LivroDeFiccao`, `LivroDeNaoFiccao`. Cada uma dessas subclasses deve implementar o método `calcularPreco()` de acordo com a sua margem de lucro correspondente.

Crie diferentes instâncias de livros e chame o método `calcularPreco()` **[75 Pontos]**

BOA SORTE