

Prova 1

Classes, Objetos, associação entre Objetos e e Uso de Coleções

Valor: 10 pontos

Para o sistema descrito abaixo, faça:

1. (2 pontos) Elabore o diagrama de classes utilizando UML;
2. (5 pontos) Implemente em Java a solução modelada para o sistema
3. (3 pontos) Implemente uma classe aplicativo que demonstra a instanciação dos objetos e o uso de todos os métodos implementados.

Sistema de Gerenciamento de Drones de Entrega

Uma empresa de logística utiliza drones autônomos para realizar entregas em diferentes regiões. Cada drone possui características específicas — como peso máximo suportado, autonomia de voo, consumo de bateria e velocidade média — que variam conforme o tipo do drone.

O sistema precisa permitir o registro, monitoramento e cálculo de desempenho de cada drone com base em suas características e condições de uso. O objetivo é modelar a hierarquia de classes e implementar métodos com cálculos e regras específicas que representem o comportamento real de cada tipo de drone.

O sistema deve possuir uma classe que representa qualquer drone da frota e que possua os seguintes atributos:

- **codigo:** Identificador único do drone.
- **modelo:** Nome ou código do modelo do drone.
- **pesoMaximo:** Peso máximo de carga em quilogramas.
- **autonomiaKm:** Distância máxima que pode percorrer com carga total.
- **velocidadeMedia:** Velocidade média em km/h.

- **bateriaAtual**: Percentual atual da bateria (0 a 100).

e o seguintes métodos:

- **double calcularTempoEstimado(double distanciaKm)**: Retorna o tempo estimado (em horas) para percorrer determinada distância, considerando a velocidade média.
 - Fórmula: $\text{tempo} = \text{distanciaKm} / \text{velocidadeMedia}$.
 - Se a distância for maior que a autonomia, retornar -1 indicando que o trajeto é inviável.
- **void exibirTempoEstimado(double distanciaKm)**: Exibe o tempo estimado ou a inviabilidade do trajeto.
- **void exibirResumo()**: Exibe informações básicas sobre o drone (estado do objeto).

Entretanto, existem tipos específicos de drone, sendo drone leve utilizado para pequenas entregas urbanas, com foco em agilidade e drone pesado projetado para cargas maiores, com menor eficiência energética e controle de temperatura.

Os drones leves possuem um atributo que define sua eficiência energética, isto é, número km percorridos por cada 1% de bateria e deve calcular o percentual de bateria consumido em uma entrega a partir da distância informada em km. A fórmula pode ser expressada por $\text{consumo} = \text{distanciaKm} / \text{eficienciaEnergetica}$. Por exemplo: Um drone leve com eficiência de 2.5 km/% e 60% de bateria tentará entregar em 100 km, logo, $100 / 2.5 = 40\%$, portanto é possível realizar a entrega e restará 20% de bateria.

Já os drones pesados possuem um atributo que define o peso da carga atual em kg e deve calcular o consumo de bateria considerando o peso da carga, quanto mais pesada, maior o gasto energético. A fórmula pode ser expressada por:

$$\text{fatorPeso} = 1 + (\text{pesoCargaAtual} / \text{pesoMaximo})$$

$$\text{consumo} = (\text{distanciaKm} / \text{autonomiaKm}) * 100 * \text{fatorPeso}$$

Se $\text{consumo} > \text{bateriaAtual}$, retornar -1 indicando que a bateria é insuficiente para realizar a entrega da carga. É importante exibir o resultado do cálculo informando a possibilidade ou impossibilidade da entrega. No caso da possibilidade de entrega, informar a quantidade de bateria restante. Exemplo de cálculo: Drone pesado com autonomia 200 km, carga de 10 kg em um limite de 20 kg, bateria 80% e distância 50 km, a resposta esperada é **“Entrega possível (restará 42,5% de bateria).”**

Implemente uma classe de teste (Aplicativo) para demonstrar o funcionamento completo de todas as classes implementadas.

Bom trabalho.