

java3

Sistema de Gerenciamento de Transportes

Interface: Transporte

Métodos:

- `iniciar()`: inicia o transporte.
- `parar()`: para o transporte.
- `calcularCusto(double distancia)`: calcula o custo do transporte com base na distância percorrida.

Interface: Compartilhavel

Método:

- `calcularLugaresDisponiveis()`: retorna o número de lugares disponíveis no transporte compartilhado.

Classe: Carro

Implementa:

- Transporte
- Compartilhavel

Atributos:

- `consumoCombustivel (double)`: quantidade de quilômetros percorridos por litro de combustível.
- `precoCombustivel (double)`: preço por litro de combustível.
- `lugaresTotais (int)`: total de lugares disponíveis no carro.
- `lugaresOcupados (int)`: número de lugares ocupados.

Métodos:

- Construtor: inicializa todos os atributos.
- `getLugaresTotais()`: retorna o total de lugares do carro.
- `iniciar()`: exibe mensagem indicando que o carro foi iniciado.
- `parar()`: exibe mensagem indicando que o carro foi parado.
- `calcularCusto(double distancia)`: calcula o custo total do combustível com base na distância percorrida.
- `calcularLugaresDisponiveis()`: calcula a quantidade de lugares restantes no carro.
- `toString(double distancia, double custo)`: retorna uma string com os detalhes do transporte, incluindo distância, custo e lugares disponíveis.

Classe: Bicicleta

Implementa:

- Transporte

Atributos:

- tempoDeUso (int): tempo total de uso da bicicleta, em minutos.

Métodos:

- Construtor: inicializa o tempo de uso como zero.
- iniciar(): exibe mensagem indicando que a bicicleta foi iniciada.
- parar(): exibe mensagem indicando que a bicicleta foi parada.
- calcularCusto(double distancia): retorna sempre custo zero, já que não há custos associados à bicicleta.
- registrarTempoDeUso(int minutos): adiciona o tempo de uso informado ao total.
- toString(double distancia, double custo): retorna uma string com os detalhes do transporte, incluindo distância, tempo de uso e custo.

Classe: Ônibus

Implementa:

- Transporte
- Compartilhavel

Atributos:

- custoPorPassageiro (double): custo individual por passageiro.
- lugaresTotais (int): total de lugares disponíveis no ônibus.
- lugaresOcupados (int): número de lugares ocupados.
- consumoCombustivel (double): quantidade de quilômetros percorridos por litro de combustível.
- precoCombustivel (double): preço por litro de combustível.

Métodos:

- Construtor: inicializa todos os atributos.
- getLugaresTotais(): retorna o total de lugares do ônibus.
- iniciar(): exibe mensagem indicando que o ônibus foi iniciado.
- parar(): exibe mensagem indicando que o ônibus foi parado.
- calcularCusto(double distancia): calcula o saldo final (lucro ou prejuízo) com base nos custos do combustível e na receita gerada pelas passagens.
- calcularLugaresDisponiveis(): calcula a quantidade de lugares restantes no ônibus.

- `toString(double distancia, double custo)`: retorna uma string com os detalhes do transporte, incluindo distância, custo total, status financeiro (lucro ou prejuízo) e lugares disponíveis.

Aplicativo: Gerenciador de Transportes

O aplicativo apresenta um menu com as seguintes opções:

1. Usar Carro

- Solicita o consumo do carro (km/l), o preço do combustível (R\$/l), o número de lugares ocupados e a distância percorrida.
- Verifica se o número de lugares ocupados não excede o limite total de lugares.
- Calcula o custo do transporte e exibe os detalhes do carro, como distância percorrida, custo total e lugares disponíveis.

2. Usar Bicicleta

- Solicita a distância percorrida e o tempo de uso da bicicleta.
- Calcula os dados relevantes e exibe os detalhes da bicicleta, como distância percorrida, tempo de uso e custo (sempre zero).

3. Usar Ônibus

- Solicita o preço da passagem por passageiro, o consumo do ônibus (km/l), o preço do combustível, o número de lugares ocupados e a distância percorrida.
- Verifica se o número de lugares ocupados não excede o limite total de lugares.
- Calcula o saldo final do transporte e exibe os detalhes do ônibus, como distância percorrida, custo total e lugares disponíveis.

4. Sair

- Exibe uma mensagem de saída e encerra o programa.