Programação Orientada a Objetos

Prova P1 – 23 de setembro de 2024

NOME DO ALUNO :
RA :

INSTRUÇÕES

1. Preencha o cabeçalho acima.
2. A prova deve ser feita com consulta a uma folha de papel a4 com o conteúdo livre.
3. O fonte desenvolvido deverá ser apenas na linguagem Java.
4. Responda cada questão no espaço correspondente (mesma folha).

DURAÇÃO DA PROVA: 3 horas

	Nota
Questão 1	
Questão 2	
Questão 3	
Questão 4	
TOTAL	

B O A P R O V A

1. (2.5 pontos) Crie uma enumeração chamada `EstadoDoCarro` com os seguintes estados: `LIGADO`, `DESLIGADO`, `MOVENDO`, e `PARADO`. Essa enumeração será usada para definir o estado atual do carro.

   Implemente uma classe `Carro` que será responsável por manipular o estado de um carro em um simulador de direção. A classe deve conter:

   - (0.25) Dois atributos privados:
     - `estado` (`EstadoDoCarro`): Define o estado atual do carro.
     - `velocidade` (double): Representa a velocidade atual do carro.
   - (0.25) Um construtor que inicializa os atributos `estado` com `EstadoDoCarro.DESLIGADO` e `velocidade` com 0.
   - (0.25) Métodos getters para ambos os atributos.
   - (1.0) Um método `alterarEstado()` que permite mudar o estado do carro:
     - Se o carro estiver `DESLIGADO`, ele só pode mudar para o estado `LIGADO`.
     - Se o carro estiver `LIGADO`, ele pode mudar para `MOVENDO`.
     - Se o carro estiver `MOVENDO`, ele pode mudar para `PARADO` e a velocidade deve ser ajustada para 0.
     - Se o carro estiver `PARADO` ele vai para `DESLIGADO`.
   - (0.5) Um método `acelerar(double incremento)` que aumenta a velocidade do carro se ele estiver no estado `MOVENDO`.
   - (0.25) Um método `mostrarEstado()` para imprimir o estado atual do carro e a velocidade na tela.

2. (2.5 pontos) Uma classe chamada Radar possuirá vários carros. Implemente métodos para inserir um carro no radar, para exibir todos os carros em movimento, a quantidade e a velocidade média. Faça, também, um método para listar os carros desligados.

3. Implemente a classe `Conta` que possua:

   - Dois atributos privados:
     - `nome` (`String`): Representa o nome do titular da conta.
     - `saldo` (`double`): Representa o saldo atual da conta.
   - Um construtor que inicializa os atributos `nome` e `saldo`.
   - Métodos getters para ambos os atributos.
   - Um método `depositar(double valor)` que adiciona o valor ao saldo da conta.
   - Um método `retirar(double valor)` que subtrai o valor do saldo da conta, desde que haja saldo suficiente. Se não houver saldo suficiente, o método deve imprimir uma mensagem de erro.
   - Um método para verificar se o saldo é maior que zero.

   Além disso, implemente a classe `Transferencia` com um método estático `transferir(Conta de, Conta para, double valor)`, que transfere um valor de uma conta para outra:

   - O método `transferir` deve subtrair o valor da conta de origem (`de`) e adicionar esse valor à conta de destino (`para`), desde que haja saldo suficiente na conta de origem.
   - Se o saldo da conta de origem for insuficiente, o método deve imprimir uma mensagem de erro indicando que a transferência não pôde ser realizada.

4. (2.5 pontos) Decida se as assertivas abaixo são verdadeiras ou falsas. Justifique brevemente as falsas.

   (a) Um atributo declarado como `final` pode ter seu valor alterado após a inicialização

   (b) Uma classe B possui dois atributos de tipo B. Esta classe cria uma estrutura de uma árvore binária

   (c) O modificador static impede o acesso de um atributo de fora de uma classe.

   (d) Um método setter dá acesso de escrita a um método

   (e) É uma má prática ter métodos públicos

   (f) O valor de uma enum é sempre público

   (g) Um inteiro é um tipo primitivo

   (h) Não é possível existir um método que possua um parâmetro de tipo `String[]`.

   (i) O atributo out da classe System é estático e público.

   (j) Um atributo String sempre terá o valor padrão ¨¨ (String vazia) na ausência de construtores.