

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DEPARTAMENTO DE COMPUTAÇÃO - UNIVERSIDADE ESTADUAL DE LONDRINA
LABORATÓRIO DE PROGRAMAÇÃO – PROF. BRUNO B. ZARPELÃO
LISTA DE EXERCÍCIOS – HERANÇA

Exercício 1:

Crie uma classe ContaBancaria.

Inclua os seguintes atributos:

- cliente: String;
- numeroConta: String;
- saldo: BigDecimal;

Inclua os seguintes métodos na classe ContaBancaria:

- sacar: não permitir que o saldo fique menor que zero;
- depositar;

Crie duas subclasses de ContaBancaria: ContaCorrente e ContaPoupanca.

Na classe ContaPoupanca, inclua o seguinte atributo:

- diaRendimento: int;

Na classe ContaPoupanca, inclua o seguinte método:

- calcularNovoSaldo: recebe como parâmetro a taxa de rendimento e atualiza o saldo;

Na classe ContaCorrente, inclua o seguinte atributo:

- limite: BigDecimal;

Na classe ContaCorrente, redefina o método sacar, permitindo saldos negativos desde que não superem o limite;

Faça uma classe Principal que permita ao usuário realizar as seguintes tarefas:

- Cadastrar a conta (poupança ou corrente) de um cliente;
- Sacar um valor da sua conta (poupança ou corrente);
- Atualizar uma conta poupança com o seu rendimento;
- Depositar um determinado valor na conta (poupança ou corrente);
- Mostrar o saldo de uma conta (poupança ou corrente);

- **Desafio:** implementar busca de uma conta pelo seu número usando o método `indexOf(Object o)` do `LinkedList` (Dica: olhar na documentação do Java como o `indexOf(Object o)` é implementado).

Exercício 2

Escreva as seguintes classes:

a) Uma classe `Pessoa` com os atributos `nome` (tipo `String`) e `sobrenome` (tipo `String`). Cada um desses atributos deve ter métodos para lê-los e alterá-los (getters e setters). A classe `Pessoa` ainda deve ter um método chamado `getNomeCompleto` que não possui parâmetros de entrada e que retorna a concatenação do atributo `nome` com o atributo `sobrenome`. Além disso, a classe deve possuir um construtor sem parâmetros e um outro construtor que recebe como parâmetros o `nome` e o `sobrenome` da pessoa e altera respectivamente os atributos `nome` e `sobrenome`.

b) Uma subclasse de `Pessoa`, chamada `Funcionario`. A classe `Funcionario` deve ter os atributos `matricula` (tipo `String`) e `salario` (tipo `BigDecimal`), com seus respectivos métodos para leitura e alteração (getters e setters). O salário de um funcionário jamais poderá ser negativo. Todo funcionário recebe seu salário em duas parcelas, sendo 60% na primeira parcela e 40% na segunda parcela. Assim, escreva os métodos `getSalarioPrimeiraParcela` que retorna o valor da primeira parcela do salário (60%) e `getSalarioSegundaParcela` que retorna o valor da segunda parcela do salário (40%).

c) Uma subclasse de `Funcionario`, chamada `Professor`. Todo professor recebe seu salário em uma única parcela. Assim, deve-se sobrescrever os métodos `getSalarioPrimeiraParcela` e `getSalarioSegundaParcela`. O método `getSalarioPrimeiraParcela` da classe `Professor` deve retornar o valor integral do salário do professor e o método `getSalarioSegundaParcela` do professor deve retornar o valor zero.

d) Uma classe `Principal` que instancia os seguintes objetos:

`pessoa1` (`Pessoa`)

`nome`: Mario

`sobrenome`: Lopes

`pessoa2` (`Funcionario`)

`nome`: Lucas

`sobrenome`: Mendes

`salario`: 2000.00

`pessoa3` (`Professor`)

`nome`: Rafael

`sobrenome`: Lira

`salario`: 500.00

Depois disso, execute as seguintes operações na seguinte ordem:

- d) Exibir a saída do método `getNomeCompleto` para os 3 objetos.
- e) Exibir a saída dos métodos `getSalarioPrimeiraParcela` e `getSalarioSegundaParcela` para os objetos `pessoa2` e `pessoa3`.

Exercício 3

Implemente as seguintes classes:

- a) Implemente uma classe `Equipamento` com o atributo `ligado` (tipo `boolean`) e com os métodos `liga` e `desliga`. O método `liga` torna o atributo `ligado` `true` e o método `desliga` torna o atributo `ligado` `false`.
- b) Implemente uma classe `EquipamentoSonoro` que herda as características de `Equipamento` e que possui os atributos `volume` (tipo `short`) que varia de 0 a 10 e `stereo` (tipo `boolean`). A classe ainda deve possuir métodos para ler e alterar o volume (getter e setter), além dos métodos `mono` e `stereo`. O método `mono` torna o atributo `stereo` falso e o método `stereo` torna o atributo `stereo` verdadeiro. Ao ligar o `EquipamentoSonoro` através do método `liga`, seu volume é automaticamente ajustado para 5.
- c) Faça uma classe principal que permita ao usuário operar um equipamento genérico e um equipamento sonoro.

Exercício 4

- a) Uma classe `MeioTransporte` com atributos `ligado` (tipo `boolean`) e `velocidade` (tipo `int`) e métodos `ligar()` e `desligar()`. O método `ligar()` torna o atributo `ligado` `true` e o método `desligar()` torna o atributo `ligado` `false`, além de tornar a velocidade zero. Crie também métodos `get/set` para modificar o atributo `velocidade`, sendo que a velocidade não pode ser negativa.
- b) Uma subclasse de `MeioTransporte` chamada `Carro`. `Carro` deve ter o atributo `quilometragem` (tipo `int`) e os métodos necessários para lê-lo e alterá-lo (`get/set`). A quilometragem não pode ser negativa, nem ultrapassar o valor 999999. A velocidade do `Carro` não pode ser negativa, nem ultrapassar 200.
- c) Crie uma classe principal que permita ao usuário operar um meio de transporte genérico e um carro;