

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

Exercício 1:

Crie uma interface ContaBancaria.

Inclua os seguintes métodos:

- sacar;
- depositar;

Crie duas classes que implementam ContaBancaria: ContaCorrente e ContaPoupanca.

Na classe ContaPoupanca, inclua os seguintes atributos:

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

Na classe ContaPoupanca, crie os seguintes métodos:

- sacar (implementação do método previsto na interface ContaBancaria): só deve permitir o saque caso o saldo resultante seja maior ou igual a zero;
- depositar (implementação do método previsto na interface ContaBancaria)
- calcularNovoSaldo: recebe como parâmetro a taxa de rendimento e atualiza o saldo;

Na classe ContaCorrente, inclua os seguintes atributos:

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

Na classe ContaCorrente, crie os seguintes métodos:

- sacar (implementação do método previsto na interface ContaBancaria): o valor do saque pode ser maior que o saldo disponível, mas deve respeitar o limite previsto para a conta;
- depositar (implementação do método previsto na interface ContaBancaria)

Crie uma interface Tributavel. Esta interface deve ter o seguinte método: calcularTributo().

Crie uma classe denominada `ContaInvestimento`. Ela deve implementar as interfaces `ContaBancaria` e `Tributavel`. Inclua os seguintes atributos nesta classe:

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

Na classe `ContaInvestimento`, crie os seguintes métodos:

- `sacar` (implementação do método previsto na interface `ContaBancaria`): só deve permitir o saque caso o saldo resultante seja maior ou igual a zero;
- `depositar` (implementação do método previsto na interface `ContaBancaria`)
- `calcularNovoSaldo`: recebe como parâmetro a taxa de rendimento e atualiza o saldo;
- `calcularTaxaAdministração`: recebe como parâmetro a taxa de rendimento e deve retornar um valor correspondente a 1% do lucro obtido com o rendimento;
- `calcularTributo` (implementação do método previsto na interface `Tributavel`): recebe como parâmetro a taxa de rendimento e deve retornar um valor correspondente a 0,5% do lucro obtido com o rendimento;

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

- Cadastrar a conta de um cliente;
- Sacar um valor da sua conta;
- Atualizar uma conta poupança com o seu rendimento;
- Depositar um determinado valor na conta;
- Mostrar o saldo de uma conta;
- Calcular os tributos de uma conta;
- Calcular a taxa de administração de uma conta investimento;

Exercício 2

Desenvolva:

a) Uma interface `Pessoa` com os seguintes métodos: `getNomeCompleto(String nome, String sobrenome)`, `getSalarioPrimeiraParcela(BigDecimal salarioTotal)` e `getSalarioSegundaParcela(BigDecimal salarioTotal)`.

b) Uma classe que implemente a interface `Pessoa`, chamada `Funcionario`. A classe `Funcionario` deve ter o atributo `matricula` (tipo `String`), 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 classe que implemente a interface `Pessoa`, chamada `Professor`. Todo professor recebe seu salário em uma única parcela. 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:

peessoa1 (Funcionario)

nome: Lucas

sobrenome: Mendes

salario: 2000.00

peessoa2 (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 objetos peessoa1 e peessoa2.

e) Exibir a saída dos métodos getSalarioPrimeiraParcela e getSalarioSegundaParcela para os objetos peessoa1 e peessoa2.

Exercício 3

Crie uma interface denominada FiguraGeometrica com um método calcularArea().

Crie as seguintes classes que implementam a interface FiguraGeometrica: Quadrado, Triangulo, Circulo e Losango. Para cada uma destas classes, implemente o método calcularArea().

Crie uma classe principal que permita ao usuário:

- Escolher entre as figuras disponíveis;
- Informar os dados da figura;
- Calcular a área da figura;