



Programação Java



Classes e interfaces abstratas

Prática integradora

Objetivo

O objetivo deste guia prático é estudar em detalhes o uso de classes abstratas e interfaces.

Are you ready? 😎👍



Exercício 1

Um banco possui diferentes tipos de transações que pode realizar, entre elas: Depósito, Transferência, Retirada de Dinheiro, Consulta de Saldo, Pagamento de Serviços. Todas as transações têm dois métodos em comum, que são `transactionOk ()` e `transactionNoOk ()`.

Com base nisso, existem diferentes tipos de clientes que podem interagir com o banco:

- **Executivos:** Fazem depósitos e transferências.
- **Básico:** realizam consultas de saldo, pagamentos de serviços e saques em dinheiro.
- **Coletores:** Realizam saques em dinheiro e verificam saldos.

Implementar o cenário conforme apropriado para permitir que os clientes mencionados acima tenham acesso aos diferentes métodos de acordo com a operação que desejam realizar.



Exercício 2

Propõe-se o desenvolvimento de um programa que permita, através de uma interface, imprimir diversos tipos de documentos.

Entre os tipos de documentos estão:

- **Currículos:** inclui uma pessoa com todos os seus atributos e uma lista de suas habilidades.
- **Livros em PDF:** Inclui atributos como número de páginas, nome do autor, título e gênero.



Relatórios: incluem um texto de comprimento n, número de páginas, autor e revisor. Representam um cenário onde cada um desses objetos é criado e que, por meio de um método estático de uma interface imprimível, qualquer tipo de documento pode ser passado e o conteúdo impresso.



Exercício 3

É solicitada a criação de uma classe abstrata denominada `Animal`, da qual derivam 3 animais: `Cachorro`, `Gato` e `Vaca`. Os 3 animais são capazes de “emitir sons”, para isso será necessário implementar um método que permita que o som emitido por cada animal seja mostrado na tela. Por exemplo, no caso do cachorro "uau", o gato "miau" e a vaca "muuu".

Com base nisso, levando em consideração os gostos alimentares de cada animal (gato e cachorro são considerados carnívoros e a vaca herbívora), inclua as interfaces necessárias para contemplar os métodos `comerMeat` ou `comerHerb`.

Feito o acima mencionado, realize no `Main` a criação de diferentes animais e a invocação de suas respectivas implementações de métodos.

Como proposta extra, sugere-se chamar um método `comerAnimal` onde, a partir da passagem de um objeto de qualquer tipo de animal como parâmetro, invoque o método de comer de acordo com aquele animal.