

Padrões

Lista 07

5 de maio de 2024

Instruções de Entrega

Nome do projeto java: ListaX-NUSP1-NUSP2. Sendo X o número da lista para ser realizado em duplas; NUSP1 e NUSP2 são os números USP dos participantes.

Entregar o projeto zipado, **apenas um projeto por dupla.**

Exercícios

1. (5,0 pontos) Considere a seguinte situação. Em um determinado banco, existem vários tipos de conta. Por exemplo, *conta corrente*, que permite que o correntista possua talão de cheque, transfira dinheiro por TED, TEF ou PIX; *conta poupança*, que paga uma taxa de juros fixa e que deve estar atrelada a uma conta corrente, sendo a única conta para a qual se pode transferir valores; *conta de investimento*, que também está atrelada a uma conta corrente para transferências, e que possui uma carteira de investimentos nos quais o correntista pode investir e resgatar. Proponha uma forma de identificar unicamente uma conta, utilizada para identificar origem e destino das transferências.

Além disso, as contas deste banco podem ter dois tipos de clientes: *peçoas físicas*, identificadas por um CPF; e *peçoas jurídicas*, identificadas por um CNPJ. As contas de pessoas físicas possuem um nome do titular da conta; já as pessoas jurídicas possuem um nome de um representante legal.

Implementar a situação descrita acima utilizando o padrão Fábrica Abstrata. Faça um programa em que uma pessoa física possui uma conta corrente e uma conta poupança associada a esta conta corrente. Esta pessoa também é a representante legal de uma empresa que possui uma conta corrente e uma conta de investimentos. Mostre uma situação em que a representante legal resgata uma aplicação de investimento da empresa, a transfere para a conta corrente da empresa e da conta corrente da empresa o valor vai para a conta corrente de sua pessoa física e daí para a sua conta poupança.

Crie uma classe de exceção que detecta transferência indevida, por exemplo, quando a conta destino não existe.

2. (5.0 pontos) Procure informações na internet sobre o padrão *Singleton*, por exemplo, buscando por *Singleton Design Pattern*.

Utilizando o padrão *Singleton*, implementar um tabuleiro de jogo da velha como uma matriz 3 por 3 em que cada casa pode receber no máximo uma peça. A peça pode ser do tipo X ou do tipo O. Faça um programa que joga uma partida de jogo da velha, recebendo um tabuleiro vazio no início do jogo. Permite jogadas alternadas iniciadas por X. Jogadas inválidas devem jogar uma Exceção. Detecta após cada jogada se o jogador venceu. Detecta antes de cada jogada se o tabuleiro esta lotado, em cujo caso anuncia VELHA. Implemente uma função que mostra o estado atual do tabuleiro. Esta função deve ser chamada após a verificação de tabuleiro lotado, ou quando a vitória é detectada.

Para a leitura do teclado usar a classe Scanner, via

```
import java.util.Scanner;
```