

Módulo 5 – Exceções e Pacotes

Lista de Exercícios

Exceções e Pacotes

Passo 1: Construir as classes *Conta* e *BancoAppException*

Classe *Conta*:

Atributos:

- numero (string)
- saldo (double)

Operações:

- Construtor com os parâmetros numero da conta e saldo inicial.
- Operação: debitar (parâmetro: valor) -> acumula valor no saldo da conta.
- Operação: creditar (parâmetro: valor) -> debita valor do saldo da conta.

Considere as seguintes regras:

a) debitar:

- valor a debitar tem que ser um número positivo maior que zero.
- conta tem que ter saldo suficiente para o débito, isto é, a conta tem que ter um valor maior ou igual a zero após o débito.
- Existe um limite para o valor a debitar que é de 2.000,00. O limite é por operação de débito (suponha que não exista limite diário).

b) creditar:

- valor a creditar tem que ser um número positivo maior que zero.

Em caso de falha nessas regras, a implementação da operação (debitar ou creditar) deve arremessar uma exceção do tipo *BancoAppException* (que deve ser criada por você, estendendo a classe *Exception* já existente no Java) com uma mensagem de erro correspondente ao problema.

ATENÇÃO: Defina a classe *Conta* no pacote *banco.negocio*

Passo 2: Construir a classe *BancoUI* (interface com o usuário)

Faça uma classe *BancoUI* que corresponderá a uma interface com usuário rudimentar via console que deverá se comportar da seguinte forma:

A classe *BancoUI* deverá instanciar o objeto *Conta* que será manipulado durante a execução do programa. Antes de instanciar o objeto *Conta*, porém, *BancoUI* deverá perguntar para o usuário o número e o saldo inicial que serão passados para o construtor do objeto *Conta*.

A operação do programa será baseada em um menu textual em console (não se preocupem com coisas do tipo limpar a tela, e etc), com as seguintes opções:

- 1 – creditar na conta
- 2 – debitar da conta
- 3 – consultar saldo

4 – finalizar o programa

Se o usuário entrar a opção 1, o programa deverá perguntar o valor a ser creditado. O usuário entrará o valor e a classe BancoUI chamará diretamente a operação creditar com o valor passado. Caso haja uma exceção, o programa deverá exibir a mensagem de erro correspondente. Caso contrário o programa deverá apresentar a mensagem “Operação bem sucedida”.

Se o usuário entrar a opção 2, o programa deverá perguntar o valor a ser creditado. O usuário entrará o valor e a classe BancoUI chamará diretamente a operação creditar com o valor passado. Caso haja uma exceção, o programa deverá exibir a mensagem de erro correspondente. Caso contrário, o programa deverá apresentar a mensagem “Operação bem sucedida”.

Se o usuário entrar a opção 3, o programa deverá apresentar o valor atual do saldo da conta.

Após executar qualquer destas três opções, o programa deverá apresentar novamente o menu de opções.

ATENÇÃO: Defina a classe BancoUI no pacote banco.ui.

Passo 3: Construir a classe BancoApp (classe com main do programa)

Crie uma terceira classe BancoApp que conterá apenas a operação main e a instanciação e execução de BancoUI.

ATENÇÃO: Defina a classe BancoApp no pacote banco.app.

NOTAS:

1) Para fazer uma entrada de dados via console, utilize as seguintes linhas:

```
BufferedReader r = new BufferedReader(new InputStreamReader(System.in));  
r.readLine();
```

O resultado de `r.readLine` é o `String` entrado pelo usuário.

Atenção que o trecho acima pode gerar uma exceção do tipo *IOException*.

Para usar as classes acima, é necessário fazer o *import* de suas definições.

2) O valor a debitar ou a creditar serão capturados da forma acima como um `string` e deverão ser convertidos para `Double` através da chamada:

```
Double.parseDouble(stringEntradoPeloUsuario);
```

Atenção que se o `string` entrado pelo usuário não for um número, a operação `parseDouble` arremessará a exceção *NumberFormatException*.