

Fundamentos de Java (Java Fundamentals)

Seção 6: Criando um Projeto de Inventário

Projeto

Visão Geral

Este projeto vai conduzi-lo ao longo das Seções 4, 5, 6 e 7 do curso. Após cada seção, outras etapas deverão ser concluídas até ser criado um aplicativo Java completo para manutenção de Inventário. Para cada parte, tome como base a parte anterior de modo que requisitos antigos e novos sejam atendidos. Inclua todas as partes em um pacote chamado Inventário.

Crie um programa de inventário que possa ser usado para uma ampla gama de produtos (cds, dvds, software etc.).

Tópico(s):

- Usando loops (Seções 5.2 e 6.1)
- Como tratar exceções (Seções 6.2)
- Usando instruções if (Seção 5.1)
- Arrays de objetos (Seção 6.1)

Instruções:

1. Abra o programa de inventário atualizado na **Seção 5: Criando um Projeto de inventário**
2. Peça ao usuário para inserir o número de produtos que deseja adicionar. Aceite um número inteiro positivo para o número de produtos e trate o valor zero.
 - a. Crie uma variável com o nome `maxSize` que pode armazenar inteiros.
 - b. Crie um prompt no início do seu método principal que instruirá o usuário a inserir o valor necessário para o número de produtos que ele deseja armazenar:

```
Insira o número de produtos que gostaria de adicionar
Insira 0 (zero) se não quiser adicionar produtos:
```
 - c. Utilize um loop do `while` para que o programa continue até um valor positivo ser inserido. Se um valor menor que zero for inserido, uma mensagem de erro informando “`Valor incorreto inserido`” deverá ser exibida antes de ser solicitado novamente que o usuário insira um novo valor. Você não deverá sair do loop até que um valor igual ou maior que zero seja inserido.
3. Agora você vai adicionar um tipo de tratamento de erros para lidar com erros de runtime no seu código. Atualmente, o seu programa lida com números inseridos fora do intervalo fornecido, mas não consegue lidar com entradas incorretas de tipo de dados.
 - a. Adicione um bloco de tentativa (`try`) envolvendo todo o código dentro do loop do `while`.
 - b. Adicione uma instrução de captura (`catch`) acima da cláusula `while` que utilizará um parâmetro **Exception** `e`. O programa deverá usar uma instrução de saída na console para exibir o valor de `e` na tela.

- c. Como agora você designará um valor a `maxSize` dentro de uma instrução de tentativa (`try`), há uma possibilidade de que `maxSize` não receba um valor quando você chegar à cláusula `while`. Para garantir que isso não aconteça, designe um valor inicial de -1 a `maxSize` ao declará-lo.
DICA: Sempre designe um valor que falhe no loop para que o seu código seja forçado a designar um valor correto antes de continuar.
 - d. Execute e teste o seu código inserindo um caractere em vez de um número.
 - e. Adicione uma linha de código na sua instrução de captura (`catch`) para esvaziar o buffer de entrada, de modo que um prompt seja exibido e que o sistema aguarde uma entrada do usuário.
 - f. Tome nota do tipo específico de Exceção produzida quando você inserir um caractere e criar uma instrução de captura (`catch`) para essa exceção. Esse erro deverá exibir uma mensagem **Tipo incorreto de dados inserido!** na console e também deverá esvaziar o buffer de entrada.
 - g. Execute e teste o seu código inserindo diversos valores de entrada distintos.
4. Modifique a classe `ProductTester` para tratar diversos produtos usando um único array dimensional se um valor maior que zero for inserido.
- a. Crie uma instrução `if` que exibirá a mensagem “Não há produtos!” na console se o valor de `maxSize` for zero.
 - b. Adicione uma instrução `Else` para lidar com qualquer valor diferente de zero. Crie um array de uma dimensão com o nome `Produtos` com base na classe `Produto` que terá o número de elementos especificados pelo usuário na variável `maxSize`.
5. Agora você vai preencher o array, obtendo os valores do usuário para cada campo em um objeto `Produto`.
- a. Dentro da instrução `else`, embaixo de onde você criou o array, escreva um loop `for` que fará uma iteração pelo array de zero a 1 menor que `maxSize`.
 - b. Como a última entrada recebida do usuário era numérica, você precisará adicionar uma instrução que esvazie o buffer de entrada na primeira linha do seu loop `for`.
 - c. Copie para o loop `for` o código utilizado para obter a entrada do usuário em relação a todos os campos de um produto. Isso inclui o nome, a quantidade, o preço e o número do item.
 - d. Adicione um novo objeto `Produto` ao array usando o valor de índice para a posição e o construtor que utiliza 4 parâmetros.
6. Use um loop `for each` para exibir as informações para cada produto individual no array de produtos.
7. Remova códigos desnecessários que não sejam usados nesse exercício.
8. Salve o seu projeto.