



Exercício: Filtrando Números Ímpares em um Conjunto (Set) em Java

Objetivo:

Criar uma expressão lambda que filtre os números ímpares de um conjunto (Set) de inteiros.

Instruções:

1. Crie um conjunto (Set) de inteiros com alguns números.
2. Implemente uma expressão lambda que filtre apenas os números ímpares.
3. Mostre o resultado após a filtragem.

Passos para fazê-lo:

1. Crie um conjunto (Set) de inteiros:

- Declare e inicialize um conjunto (Set) de inteiros (por exemplo, `Set<Integer>`).

2. Implemente a expressão lambda:

- Use o método `removeIf` do conjunto para aplicar a expressão lambda de filtragem.
- A expressão lambda deve verificar se o número é ímpar (ou seja, se o resto da divisão por 2 é diferente de zero).

Dica:

```
Set<Integer> numeros = new HashSet<>();  
// adicione elementos  
numeros.removeIf( "expressão lambda aqui!" );
```

3. Exiba os números ímpares resultantes:

- Após a filtragem, exiba os números ímpares presentes no conjunto.

Código do Exercício:

```
import java.util.HashSet;
import java.util.Set;

public class Main {
    public static void main(String[] args) {
        // 1. Crie um conjunto (Set) de inteiros
        Set<Integer> numeros = new HashSet<>();
        numeros.add(1);
        numeros.add(2);
        numeros.add(3);
        numeros.add(4);
        numeros.add(5);

        // 2. Implemente a expressão lambda para filtrar números
        // ímpares
        numeros.removeIf(n -> n % 2 == 0);

        // 3. Exiba os números ímpares resultantes
        System.out.println("Números ímpares: " + numeros);
    }
}
```

Explicação do Código:

- Criamos um conjunto de inteiros chamado `numeros`.
- Usamos a expressão lambda `n -> n % 2 == 0` para filtrar os números ímpares (removendo os pares).
- O método `removeIf` remove os elementos que atendem à condição da expressão lambda.
- Por fim, exibimos os números ímpares resultantes.



Exercício: Manipulação de Listas com Expressões Lambda

Objetivo:

O objetivo deste exercício é praticar o uso de expressões lambda em Java para manipular listas.

Passos para fazer o exercício

1. Criação da Lista:

- **Objetivo:** Inicializar uma lista de strings e adicionar elementos a ela.
- **Instruções:**
 - Use apenas o método main (se quiser pode criar mais métodos).
 - Crie uma instância de `ArrayList` do tipo `String`.
 - Utilize o método `add` para adicionar strings à lista.

- **Código:**

```
List<String> minhaLista = new ArrayList<>();  
minhaLista.add("maça");  
minhaLista.add("banana");  
minhaLista.add("cereja");  
minhaLista.add("abacaxi");
```

- **Explicação:** Aqui, estamos criando uma lista chamada `minhaLista` e adicionando quatro frutas a ela.

2. Remoção de Elementos:

- **Objetivo:** Remover strings da lista que tenham mais de 5 caracteres.
- **Instruções:**
 - Utilize o método `removeIf` com uma expressão lambda que verifica o comprimento das strings.

- **Código:**

```
minhaLista.removeIf(s -> s.length() > 5);
```

- **Explicação:** A expressão lambda `s -> s.length() > 5` verifica se o comprimento da string `s` é maior que 5. Se for, a string é removida da lista.

3. Adição de Elemento:

- **Objetivo:** Adicionar uma nova string no início da lista.
- **Instruções:**
 - Utilize o método `add` com o índice `0` para inserir a string na primeira posição.
- **Código:**

```
minhaLista.add(0, "uva");
```

- **Explicação:** A string "grape" é adicionada na posição `0` da lista, ou seja, no início da lista.

4. Impressão da Lista:

- **Objetivo:** Iterar sobre a lista e imprimir cada elemento.
- **Instruções:**
 - Utilize o método `forEach` com uma referência ao método `println` de `System.out`.
- **Código:**

```
minhaLista.forEach(System.out::println);
```

- **Explicação:** O método `forEach` itera sobre cada elemento da lista e imprime cada um deles no console.

Código Completo do Exercício

```
import java.util.ArrayList;
import java.util.List;

public class ListaManipulacaoLambda {

    public static void main(String[] args) {
        // Passo 1: Criação da Lista
        List<String> minhaLista = new ArrayList<>();
        minhaLista.add("maça");
        minhaLista.add("banana");
        minhaLista.add("cereja");
        minhaLista.add("abacaxi");

        // Passo 2: Remoção de Elementos
        minhaLista.removeIf(s -> s.length() > 5);

        // Passo 3: Adição de Elemento
        minhaLista.add(0, "uva");

        // Passo 4: Impressão da Lista
        minhaLista.forEach(System.out::println);
    }
}
```

Explicação do Código

1. **Criação da Lista:** Uma lista de strings chamada `minhaLista` é criada e inicializada com quatro frutas.
2. **Remoção Condicional:** Utilizando a expressão lambda `s -> s.length() > 5`, removemos todas as strings da lista que possuem mais de 5 caracteres.
3. **Inserção de Elemento:** Adicionamos a string "uva" no início da lista usando o método `add` com índice 0.
4. **Impressão da Lista:** Utilizamos o método `forEach` com uma referência de método (`System.out::println`) para imprimir cada elemento da lista resultante.