| Uma imagem contendo placar, desenho, relógio  Descrição gerada automaticamente | **ATIVIDADE ADICIONAL PRÁTICA – JAVA** |
| --- | --- |
| JAVA-08 – Collection Java |

**Instruções gerais:**

| 1. Utilize o Eclipse ou o STS para desenvolver os algoritmos. |
| --- |
| **Mantenha as entregas das Atividades em dia na Plataforma Canvas** |

**EXERCÍCIOS**

**Boas práticas:**

1. Leia os enunciados do exercício com atenção
2. Observe as indicações de Entrada e Saída esperadas em cada exercício
3. Observe com atenção os desenhos e diagramas inseridos nos exercícios para facilitar a compreensão
4. Utilize o Cookbook, os Vídeos da Plataforma e os Códigos guia como referências para a resolução dos exercícios
5. Caso ainda fique alguma dúvida, consulte os instrutores da sua turma pelo Discord

**Lista de exercícios**

### **Exercícios com List:**

1. **Armazenamento e Exibição de Nomes**:
   * Crie um programa que leia 5 nomes do usuário, armazene-os em uma ArrayList, e depois exiba todos os nomes.
2. **Busca de Elemento**:
   * Escreva um programa que armazene uma lista de 10 números inteiros em uma ArrayList. Peça ao usuário um número e verifique se esse número está na lista.
3. **Remoção de Elemento por Índice**:
   * Crie um programa que armazene uma lista de 5 cores em uma ArrayList. Remova a cor na terceira posição e exiba a lista atualizada.
4. **Atualização de Elemento**:
   * Escreva um programa que armazene 5 cidades em uma LinkedList. Substitua a cidade na segunda posição por outra cidade e exiba a lista atualizada.
5. **Ordenação de Elementos**:
   * Crie um programa que armazene 7 números em uma ArrayList. Ordene os números em ordem crescente e exiba a lista ordenada.

### **Exercícios com Set:**

1. **Armazenamento de Palavras Únicas**:
   * Escreva um programa que leia uma frase do usuário, armazene cada palavra em um HashSet, e depois exiba as palavras únicas (sem repetições).
2. **Remoção de Elementos Repetidos**:
   * Crie um programa que armazene uma lista de 10 números inteiros, alguns deles repetidos, em uma ArrayList. Converta essa lista em um HashSet para remover os elementos duplicados e exiba o conjunto resultante.
3. **Comparação de Conjuntos**:
   * Crie dois TreeSet e adicione alguns elementos em cada um. Em seguida, determine e exiba a interseção, união e a diferença entre os dois conjuntos.
4. **Manutenção da Ordem de Inserção**:
   * Crie um LinkedHashSet que armazene uma sequência de nomes inseridos pelo usuário e exiba os nomes na ordem em que foram inseridos.
5. **Verificação de Conjunto Vazio**:
   * Escreva um programa que crie um HashSet para armazenar nomes de animais. Adicione alguns nomes ao conjunto, remova todos os elementos e, em seguida, verifique se o conjunto está vazio.