

Estrutura de Dados: (Programação Orientada a Objetos)

Utilizou a estrutura de dados nativa do Java, a classe 'HashSet' para implementar o tipo abstrato de conjunto

Algoritmos:

- É Vazio (isEmpty): Verifica-se o comprimento da lista é igual a zero
- União (union): Combina o elemento de dois Arrays, evitando duplicatas
- Interseção (intersect): Identifica os elementos que são comuns a ambos Arrays
- Diferença Simétrica (diffSym): Identifica os elementos exclusivos de cada Array
- Diferença (diff): Identifica os elementos presentes no primeiro conjunto mas ausentes no segundo conjunto
- Adição (add): Adiciona um elemento ao array se não houver
- Remoção (remove): Remove um elemento do Array se houver
- Tamanho (size): Retorna o comprimento do Array
- Elemento Pertencente (contains): Verifica se um elemento está presente no Array

Utilização

- Os conjuntos são representados como arrays de objetos, portanto podem ser utilizados com chamadas de função

```
17 // Função principal para testar as operações de conjunto
18 public static void main(String[] args) {
19     Scanner scanner = new Scanner(System.in);
20
21     // Entrada dos conjuntos A e B
22     System.out.print("Digite os elementos do conjunto A separados por espaço: ");
23     Object[] conjunto_A = lerArray(scanner);
24
25     System.out.print("Digite os elementos do conjunto B separados por espaço: ");
26     Object[] conjunto_B = lerArray(scanner);
27
28     // Exibição dos conjuntos A e B
29     System.out.println("A: " + arrayParaString(conjunto_A));
30     System.out.println("B: " + arrayParaString(conjunto_B));
31
32     // Realização das operações e exibição dos resultados
33     System.out.println("É vazio (A): " + ehVazio(conjunto_A));
34     System.out.println("União de A e B: " + arrayParaString(uniaoConj(conjunto_A, conjunto_B)));
35     System.out.println("Interseção de A e B: " + arrayParaString(intersConj(conjunto_A, conjunto_B)));
36     System.out.println("Diferença A - B: " + arrayParaString(diffConj(conjunto_A, conjunto_B)));
37     System.out.println("Diferença Simétrica A e B: " + arrayParaString(diffSimConj(conjunto_A, conjunto_B)));
38
39     // Adição de um elemento ao conjunto A
40     System.out.print("Digite um elemento para adicionar ao conjunto A: ");
41     Object elemento_add = lerElemento(scanner);
42     conjunto_A = addConj(conjunto_A, elemento_add);
43     System.out.println("A após adição: " + arrayParaString(conjunto_A));
44
45     // Remoção de um elemento do conjunto A
46     System.out.print("Digite um elemento para remover do conjunto A: ");
47     Object elemento_rem = lerElemento(scanner);
48     conjunto_A = remConj(conjunto_A, elemento_rem);
49     System.out.println("A após remoção: " + arrayParaString(conjunto_A));
50
51     // Exibição do tamanho do conjunto A
52     System.out.println("Tamanho de A: " + tamConj(conjunto_A));
53
54     // Verificação se um elemento está presente no conjunto A
55     System.out.print("Digite um elemento para verificar se está em A: ");
56     Object elemento_verificar = lerElemento(scanner);
57     System.out.println(elemento_verificar + " pertence a A: " + ehElemConj(conjunto_A, elemento_verificar));
58 }
```

tp_2021020317.java