

Gabriel Felipe da Silva de Braga
Arthur Felipe Flamino de Oliveira
Tagore Nataniel de Lara
Eduardo Zaduski Mazzotti

TreeSet

DESCRIÇÃO

- **TreeSet** é uma estrutura de classe de Java na qual representa uma árvore, classificando itens em classificações, subclassificações e mais subclassificações.
- Os pontos nos quais “os galhos” terminam, são nódulos pretos e os lugares onde se dividem, são vermelhos.
- **Os nódulos pretos:** são nódulos normais que tendem a crescer.
- **Os nódulos vermelhos:** são organizadores evitam que fiquem muito alta.

PRINCIPAIS CARACTERISTICAS

- Elementos classificados automaticamente.
- Fácil de pesquisar, inserir e excluir elementos, com complexidade de tempo rápida.
- Útil ao trabalhar com coleções que requerem elementos distintos.
- É um pouco mais lento que HashSet devido a manutenção de ordem de classificação
- Não permite elementos nulos.

COMO IMPLEMENTAR

- A melhor forma de implementar é quando já se tem uma lista com dados pré-definidos.
- Garante que não haja elementos duplicados, para que o armazenamento seja único.
- Boa eficiência para a navegação de grandes conjuntos de dados.

OPERAÇÕES COMUNS E COMPLEXIDADE TEMPORAL

- `treeSet.add()`
- `treeSet.remove()`
- `treeSet.contains()`
- `treeSet.first()/last()`
- `treeSet.forEach()`

Complexidade: $O(\log n)$

Quanto maior o valor dos dados, maior a demora de execução

EXEMPLOS

```
import java.util.TreeSet;

public class ExemploTreeSet {
    public static void main(String[] args) {
        TreeSet<Integer> treeSet = new TreeSet<>();

        treeSet.add(5);
        treeSet.add(2);
        treeSet.add(8);
        treeSet.add(1);

        System.out.println("TreeSet de números inteiros:");
        System.out.println(treeSet);
    }
}
```

```
import java.util.TreeSet;
```

```
public class TreeSet {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        TreeSet<String> members = new TreeSet<>();

        members.add("João");
        members.add("Maria");
        members.add("Pedro");

        boolean continuar = true;

        while (continuar) {
            System.out.println("Membros atuais: " + members);
            System.out.println("Escolha uma opção:");
            System.out.println("1. Adicionar membro");
            System.out.println("2. Remover membro");
            System.out.println("3. Sair");

            int opcao = scanner.nextInt();
            scanner.nextLine();

            switch (opcao) {
                case 1:
                    System.out.println("nome do novo membro:");
                    String novoMembro = scanner.nextLine();
                    members.add(novoMembro);
                    System.out.println("Membro adicionado!");
                    break;
                case 2:
                    System.out.println("nome do membro que deseja remover:");
                    String membroRemover = scanner.nextLine();
                    if (members.contains(membroRemover)) {
                        members.remove(membroRemover);
                        System.out.println("Membro removido!");
                    } else {
                        System.out.println("Membro não encontrado!");
                    }
                    break;
                case 3:
                    continuar = false;
                    break;
                default:
                    System.out.println("Opção inválida!");
            }
        }

        System.out.println("Programa encerrado.");
        scanner.close();
    }
}
```

DICAS RELEVANTES

- Não modificar os elementos dentro do TreeSet.
- Fazer uso de comparador dentro do java, caso o tipo de dados não tenha uma forma de classificação padrão.