**Fatec**

**São Caetano do Sul**

**Antônio Russo**

**TADS - Análise e Desenvolvimento de Sistemas**

**Estrutura de Dados**



**Bimestre #3 Avaliação**

**Ana Clara Lio do Nascimento e Maria Vitória Lemos Romano**

**25/09/2023**

**CONTENTS**

1. **Resumo**
2. **Argumentação teórica** 
   1. Modelo de pesquisa (Percurso em ordem)
   2. Recursividade
   3. Árvore binária
3. **Resultados obtidos**
4. **Conclusão**
5. **Resumo**

Este artigo explora um programa de árvore binária em Java que permite aos usuários criar uma árvore binária interativamente e, em seguida, imprime os elementos em ordem crescente. A árvore binária é uma estrutura de dados amplamente utilizada na ciência da computação, e este programa oferece uma oportunidade de entender sua criação, manipulação e travessia. Neste artigo, discutiremos a argumentação teórica por trás do programa, incluindo o modelo de pesquisa (percurso em ordem) utilizado, a recursividade e o conceito de árvore binária. Além disso, apresentaremos resultados práticos da execução do programa, incluindo a análise do tempo de execução para o método de busca binária.

1. **Argumentação Teórica**
   1. **Modelo de Pesquisa (Percurso em Ordem)**

O programa utiliza o modelo de pesquisa em ordem para percorrer a árvore binária e imprimir seus elementos. Esse modelo de pesquisa visita os nós em ordem crescente de valores. Inicialmente, o programa entra no nó raiz e, em seguida, percorre todos os nós à esquerda, visitando-os em ordem crescente, depois visita o próprio nó e, por fim, percorre todos os nós à direita, também em ordem crescente. Esse modelo garante que os elementos sejam impressos em ordem crescente, tornando-o adequado para classificar dados armazenados na árvore.

* 1. **Recursividade**

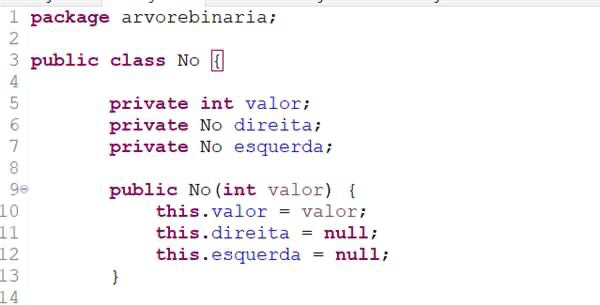
A recursividade é uma parte fundamental da implementação do programa. Os métodos criarArvore e imprimirEmOrdem utilizam recursão para percorrer a árvore binária. A recursão é um conceito importante em estruturas de dados de árvore, pois permite percorrer a estrutura de maneira eficiente e elegante. No caso do método criarArvore, ele chama a si mesmo para criar os nós filhos, criando assim a estrutura da árvore. Já o método imprimirEmOrdem utiliza a recursão para percorrer os nós da árvore em ordem, conforme explicado anteriormente.

* 1. **Árvore Binária**

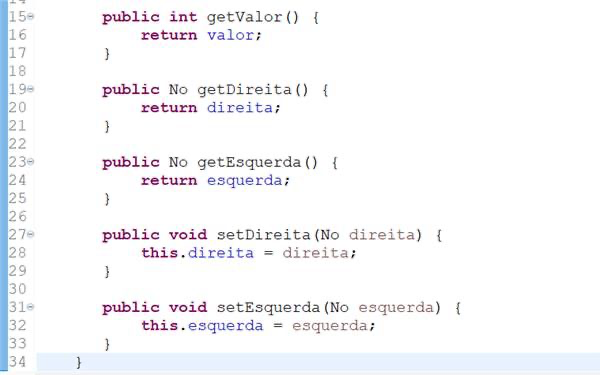
A árvore binária é uma estrutura de dados hierárquica que consiste em nós conectados por arestas. Cada nó pode ter, no máximo, dois filhos: um à esquerda e outro à direita. O nó no topo da hierarquia é chamado de nó raiz, e os nós que não têm filhos são chamados de folhas. A árvore binária é amplamente utilizada em algoritmos de pesquisa, classificação e muitos outros problemas computacionais. Sua estrutura permite um acesso eficiente aos dados, especialmente quando organizados em ordem.

1. **Resultados Obtidos**

**3.1 Classe No**



Essa classe representa um nó na árvore binária. Possui um valor (`valor`) e seus nós da esquerda e direita (`esquerda` e `direita`).



O trecho acima possui os seguintes métodos:

`getValor()`: um getter que retorna o valor armazenado no nó da árvore binária. Ele permite acessar o valor associado a um nó a partir de outras partes do código.

`getDireita()`: outro getter, mas este retorna a referência para o nó à direita do nó atual na árvore binária. Isso permite acessar o filho direito do nó a partir de outras partes do código.

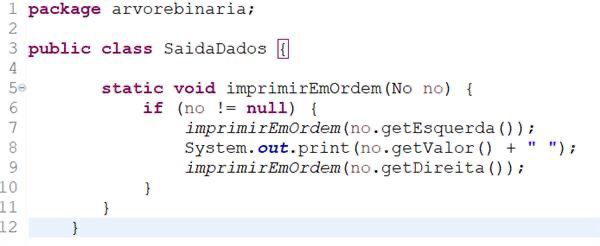
`getEsquerda()`: este método é tem a mesma função do método acima, no sentido contrário, ou seja, retorna a referência para o nó à esquerda do nó atual na árvore binária.

`setDireita(No direita)`: setter que permite definir a referência para o nó à direita do nó atual na árvore binária. Isso é utilizado para atribuir um nó como filho direito do nó atual.

`setEsquerda(No esquerda)`: outro setter que permite definir a referência para o nó à esquerda do nó atual na árvore binária. Também é utilizado para atribuir um nó como filho esquerdo do nó atual.

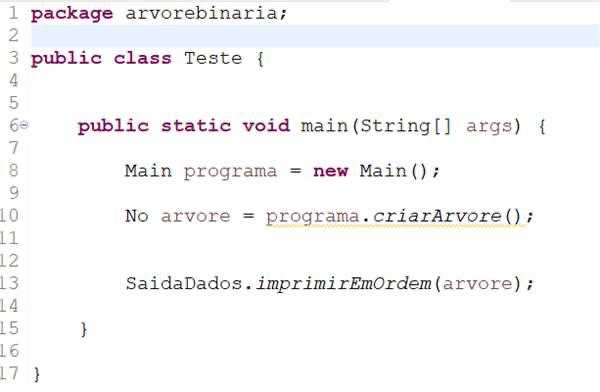
Esses métodos são essenciais para manipular e navegar na estrutura da árvore binária, permitindo acessar os valores e filhos de cada nó da árvore.

**3.2 Classe SaidaDados**



A classe acima contém um método estático `imprimirEmOrdem(No no)` que imprime a árvore binária em ordem (esquerda-raiz-direita) recursivamente.

**3.3 Classe Teste**



Esta classe serve como ponto de entrada para o teste. Ele cria uma instância da classe `Main`, inicializa a árvore binária e depois a imprime, em ordem, utilizando o método `SaidaDados.imprimirEmOrdem()`.

1. **Conclusão**

Este artigo explorou um programa de árvore binária em Java, apresentando a argumentação teórica por trás do modelo de pesquisa em ordem, o uso da recursividade e a importância da árvore binária em estruturas de dados. Além disso, demonstrou a saída esperada para uma árvore construída com valores específicos. O programa também oferece uma abordagem interativa para criar e explorar árvores binárias.