CENTRO PAULA SOUZA

FATEC ANTONIO RUSSO

Análise e Desenvolvimento de Sistemas - AMS

INSTRUMENTO AVALIATIVO TERCEIRO BIMESTRE

ESTRUTURA DE DADOS:

Árvore Binária

Profº Carlos Henrique Veríssimo Pereira

Guilherme Catti de Almeida Manso Santos

Kauê da Luz Catto

São Caetano do Sul

19/09/2023

SUMÁRIO

1. RESUMO

1.1. Contexto

Este programa se baseia em uma pesquisa que se concentra na implementação de um modelo de pesquisa de árvore binária na linguagem de programação Java. Uma árvore binária é uma estrutura de dados que é amplamente utilizada em algoritmos de busca e ordenação. A pesquisa envolve a criação de um método chamado "emOrdem" para percorrer uma árvore binária e imprimir os valores dos nós em ordem crescente. Além disso, a pesquisa explora o conceito de recursividade na inserção e busca de nós na árvore.

1.2. Propósito

O propósito deste programa é desenvolver um modelo de pesquisa de árvore binária que possa ser utilizado para percorrer e ordenar os valores dos nós da árvore em ordem crescente. Isso é útil em uma variedade de aplicações, incluindo algoritmos de ordenação e busca de informações hierárquicas.

1.3. Metodologia

A pesquisa utiliza um método chamado "emOrdem" para percorrer a árvore binária de forma recursiva. O método de inserção na árvore também é implementado de forma recursiva. A inserção começa pelo nó raiz e percorre a árvore até encontrar o local correto para inserir um novo nó. O método "emOrdem" percorre a árvore da esquerda para a direita, concatenando os valores dos nós em ordem crescente em um StringBuilder.

1.4. Resultados

Os resultados da pesquisa incluem a implementação bem-sucedida do modelo de pesquisa de árvore binária e a demonstração de como ele pode ser usado para percorrer e ordenar os valores dos nós em ordem crescente. Os resultados são apresentados em forma de código e explicação teórica.

1.5. Conclusão

Em conclusão, este programa demonstra a utilidade de uma árvore binária na ordenação e busca de dados de forma eficiente. A implementação bem-sucedida do método "emOrdem" permite a ordenação dos valores dos nós em ordem crescente, o que pode ser valioso em uma variedade de cenários de programação. Além disso, a pesquisa ressalta a importância da recursividade na manipulação de árvores binárias. Em suma, a pesquisa contribui para o entendimento e aplicação de estruturas de dados e algoritmos em programação.

2. ARGUMENTAÇÃO TEÓRICA,

2.1. Modelo de pesquisa

A pesquisa utilizada foi Em Ordem. Foi criado um método chamado "emOrdem" para percorrer uma árvore binária e imprimir os valores dos nós em ordem crescente. A ideia básica é visitar os nós da árvore na seguinte ordem: primeiro, visita o nó esquerdo, depois visita o próprio nó e por fim visita o nó direito.

2.2. A recursividade

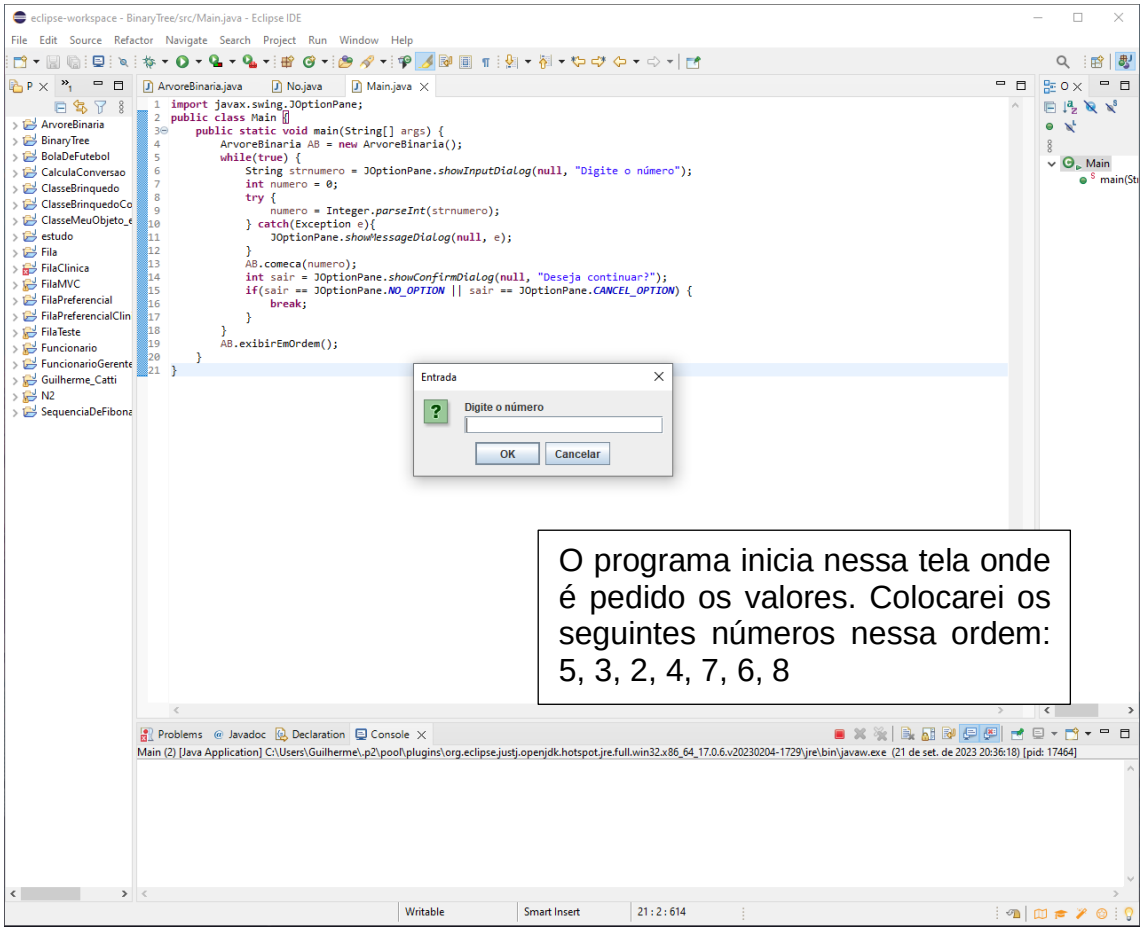
Tanto a função de incluir nós na árvore quanto a função de buscá-la em ordem estão implementadas de forma recursiva, para inserir. O método inserir recebe dois argumentos: atual (que é o nó atual da árvore) e valor (o valor que desejamos inserir na árvore). A primeira verificação compara se atual é null, o que indica que chegamos a um ponto na árvore onde podemos inserir o novo valor. Se atual for null, é criado um nó chamado novoNo com o valor fornecido e ele é adicionado a lista de nós. Em seguida é retornado esse novo nó. Se atual não for null, é comparado o valor com o valor do nó atual. Se o valor for menor, é chamado recursivamente o método inserir no nó esquerdo (atual.esquerda). Se o valor for maior, é chamado recursivamente o método inserir no nó direto (atual.direita). A recursão continua até ser encontrado um nó null, onde o novo nó é inserido na árvore, ou até encontrarmos o local correto para inserir o valor. Em segue, é retornado o nó atual.
O método emOrdem é usado para percorrer a árvore binária em ordem crescente. Ele recebe dois argumentos: atual (o nó atual da árvore) e resultado (um StringBuilder que será utilizado para concatenar os valores para mostrar futuramente em um JOptionPane). A primeira verificação é se atual é null. Se for, nada é feito. Se atual não for null, é chamado recursivamente o método emOrdem no nó esquerdo (atual.esquerda). Isso nos leva ao nó mais à esquerda da árvore, que contém o menor valor. Em seguida, anexamos o valor do nó atual ao StringBuilder chamado resultado. Depois é chamado recursivamente o método emOrdem e no nó direito (atual.direita). Isso nos leva a percorrer todos os valores maiores em ordem crescente. A recursão continua até que todos os nós tenham sido processados, e o StringBuilder terá todos os valores da árvore em ordem crescente.

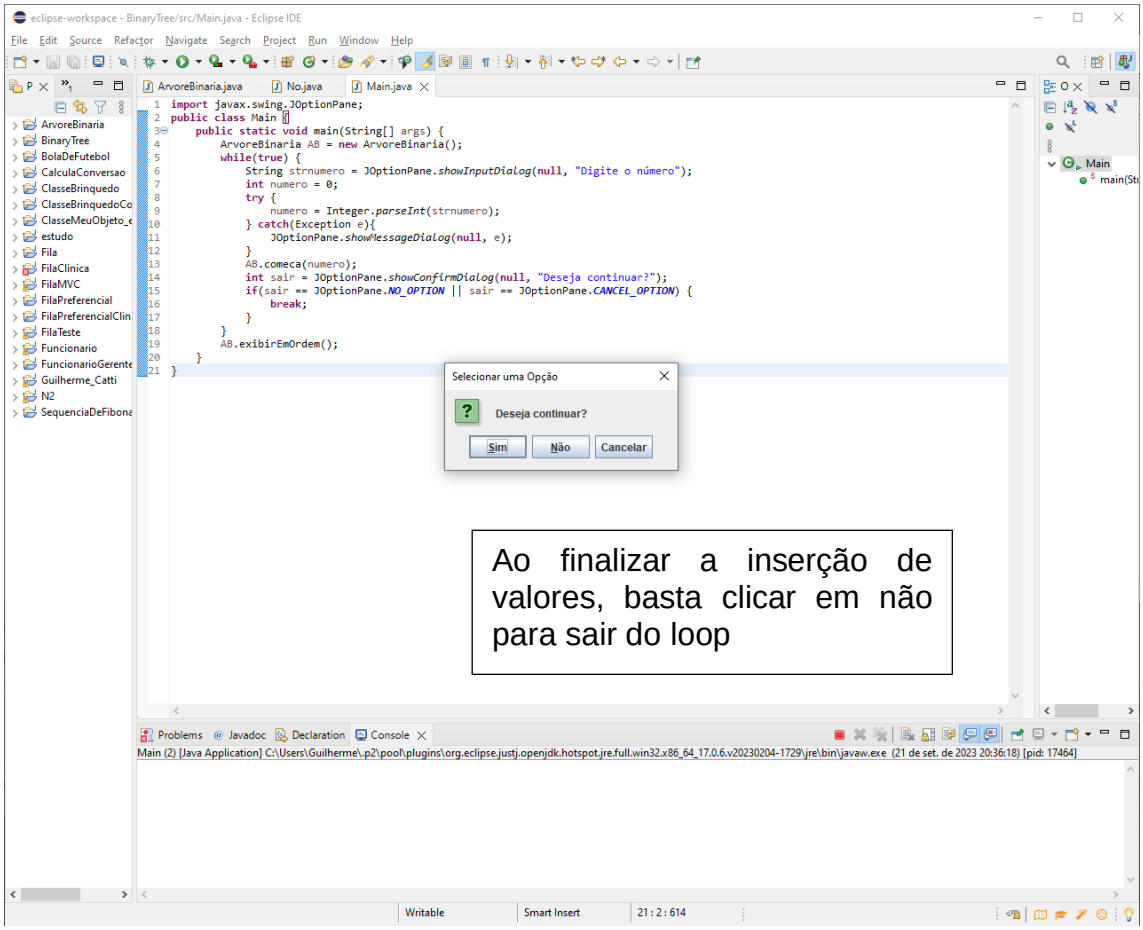
2.3. Árvore binária

Uma árvore binária é uma estrutura de dados na qual cada nó pode ter até dois filhos: um à esquerda e um à direita. Cada nó contém um valor e pode ter nenhum, um ou dois filhos. Os nós são classificados como pais, filhos à esquerda e filhos à direita, dependendo da posição na árvore. Nós sem filhos somos chamados de folhas, enquanto nós com pelo menos um filho são considerados internos. O nível de um nó é a distância até a raiz, começando com 0 para a raiz. A altura da árvore é o comprimento máximo do caminho de um nó folha até a raiz. Uma subárvore é uma árvore contida dentro de outra árvore. As árvores binárias são usadas para representar informações hierárquicas e em algoritmos de busca e ordenação.

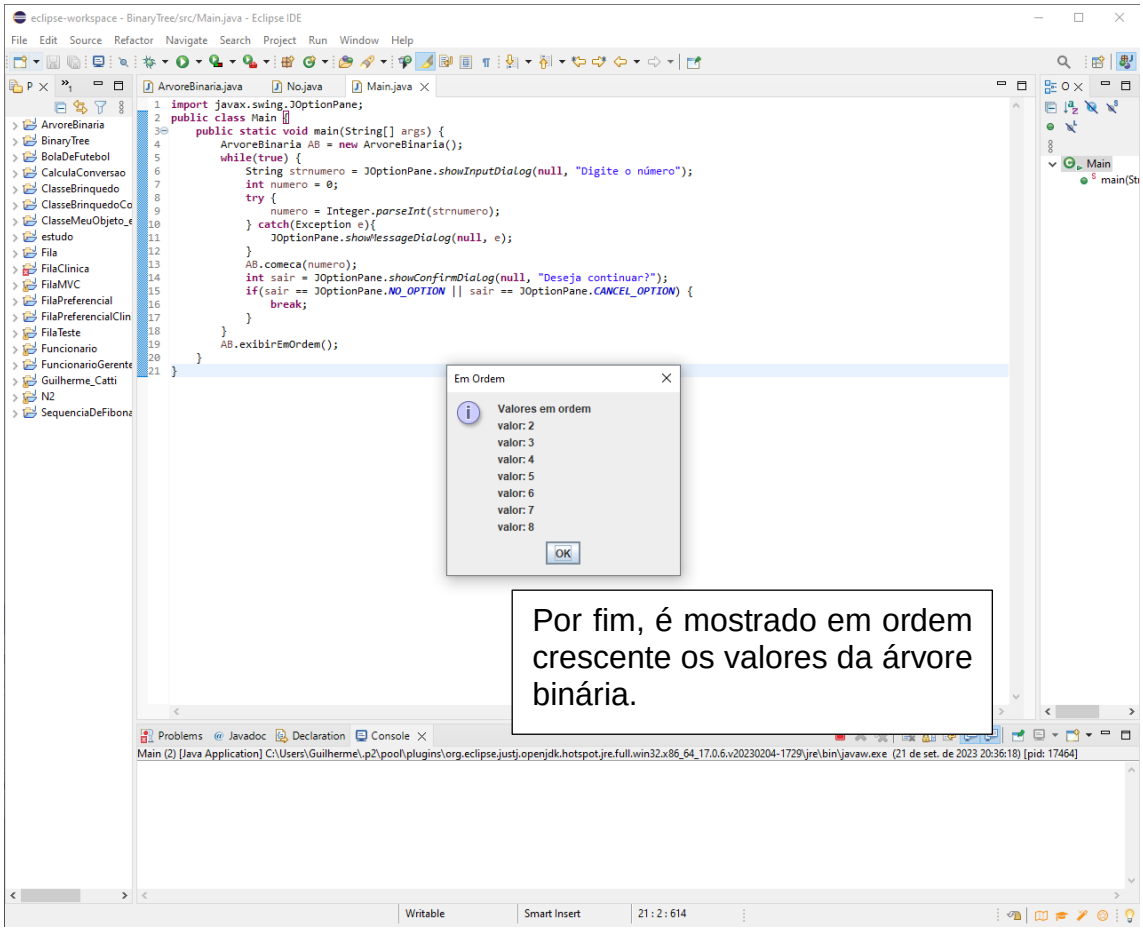
3. RESULTADOS OBTIDOS

3.1. Execução do programa





Ao finalizar a inserção de valores, basta clicar em não para sair do loop



3.2. Cálculo do tempo

```
    public void emOrdem(No atual, StringBuilder resultado) {

1.  if (atual != null) {
2.  emOrdem(atual.esquerda, resultado); // Vai para o nó esquerdo
3.  resultado.append("valor: ").append(atual.valor).append("\n");
4.  emOrdem(atual.direita, resultado); // Vai para o nó direito

    }

    }
```

L1 = t
L2 = n
L3 = 3t
L4 = n
Total = t+n(t+n+3t+n) + 3t + n(t+n+3t+n)
t+n(4t + 2n) + 3t + n(4t + 2n)
$t + 4tn + 2n^2 + 3t + 4tn + 2n^2$
$4n^2 + 8tn + 4t$

Onde n = quantidade de nós.

4. LINKS

Links para o projeto no GitHub:

https://github.com/GuilhermeCatti/FATEC-ED-1681432312037-Guilherme/tree/b8f758d875f40d0dc26359de781f87ca1602b82a/Aula26

https://github.com/KaueDaLuzCatto/FATEC_ED_1681432312014_KAUE.git