

# SBVESDD: Estruturas de Dados

## Aula 06: Estruturas de Dados Não-Lineares - Tabelas de Símbolos

Bacharelado em Ciência da Computação  
Prof. Dr. David Buzatto

# Tabela de Símbolos

## Contextualização

- ▶ A partir de agora começaremos a lidar com estruturas de dados mais sofisticadas e que possuem diversos tipos de aplicações importantes. Essas estruturas não mais serão organizadas geometricamente em uma dimensão, mas sim em várias. A tabela de símbolos é a primeira estrutura de dados que iremos tratar, sendo que sua implementação poderá ser feita de diversas formas, abordando muitos conceitos importantes e essenciais para a formação de um cientista da computação. A característica mais importante de uma tabela de símbolos é que ela consegue associar chaves com valores. Provavelmente vocês já tenham usado uma tabela de símbolos 😊

# Tabela de Símbolos

## Aplicações

- ▶ Uma tabela de símbolos é utilizada sempre em que há a necessidade de se associar uma chave a um valor. Por exemplo, eu quero associar um nome (chave) a um peso (valor), ou então um estado de um autômato (chave) a um apelido (valor), ou mesmo uma letra (chave) à quantidade de vezes (valor) que ela aparece em um texto. As tabelas de símbolos possuem papel fundamental também no processo de compilação e/ou interpretação de código, pois podem armazenar o estado das variáveis do programa.
- ▶ As tabelas de símbolos nos acompanharão por algumas aulas a partir de agora!

# Tabela de Símbolos

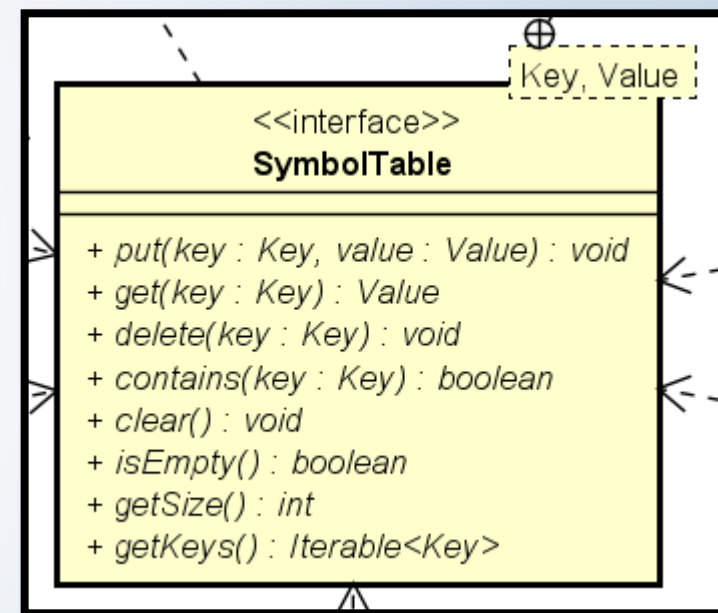
## Implementação

- Iremos estudar quatro abordagens para implementar as Tabelas de Símbolos:
  1. Busca Sequencial em Lista Encadeada;
  2. Busca Binária em Array Ordenado;
  3. Árvores Binárias de Busca;
  4. Tabelas de Dispersão.
- Tanto as árvores binárias de busca, quanto as tabelas de dispersão, também chamadas de tabelas de espalhamento ou tabelas *hash*, podem ser implementadas de diversas formas. Nós veremos:
  - Árvore Binária de Busca;
  - Árvore AVL (Adelson-Velsky e Landis);
  - Árvore Vermelho-Preto;
  - Tabela de Dispersão com Encadeamento;
  - Tabela de Dispersão com Sondagem Linear.

# Tabela de Símbolos

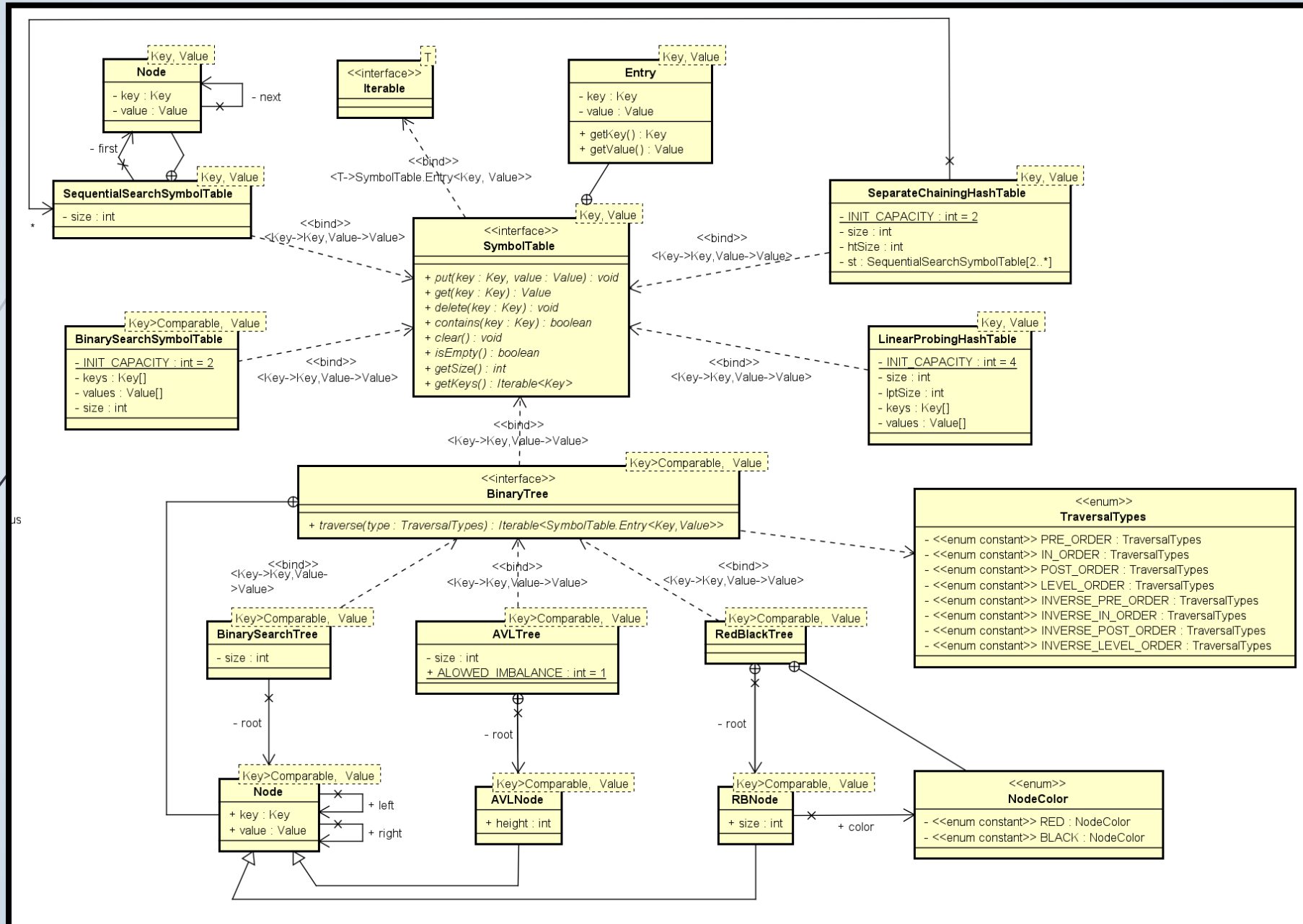
## Implementação

- Seguiremos uma API padrão (em inglês):
  - **put**: insere um par chave/valor na tabela de símbolos. Caso a chave já exista, sobrescreve o valor antigo. Caso o valor seja **null**, remove a chave da tabela;
  - **get**: obtém o valor associado à uma chave;
  - **delete**: remove uma chave e seu valor associado da tabela de símbolos;
  - **contains**: verifica se uma chave está contida na tabela de símbolos;
  - **clear**: limpa a tabela de símbolos, removendo todos os pares chave/valor;
  - **isEmpty**: verifica se a tabela de símbolos está vazia;
  - **getSize**: obtém a quantidade de itens/elementos contidos na tabela de símbolos;
  - **getKeys**: retorna todas as chaves contidas na tabela de símbolos.





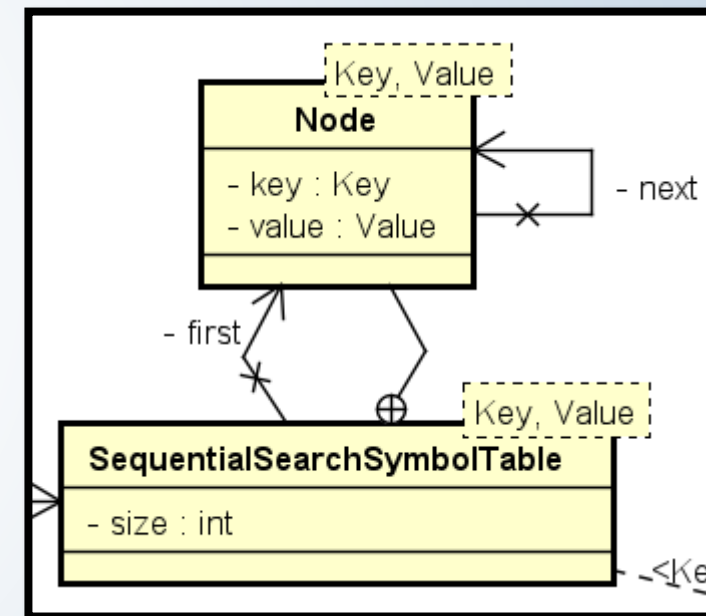
# Tabela de Símbolos - Implementação



# Tabela de Símbolos

Implementação: **SequentialSearchSymbolTable**

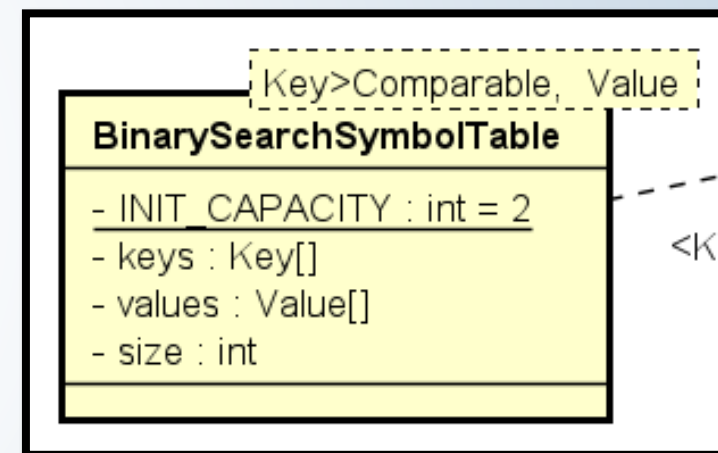
- Implementação de uma tabela de símbolos usando busca sequencial em uma lista de pares chave/valor não ordenada.



# Tabela de Símbolos

Implementação: **BinarySearchSymbolTable**

- Implementação de uma tabela de símbolo usando busca binária em um array ordenado.

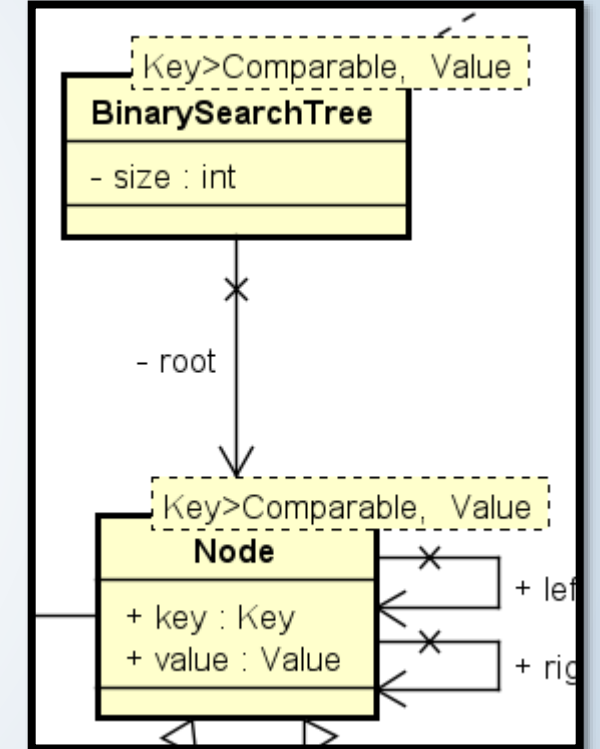




# Tabela de Símbolos

## Implementação: **BinarySearchTree**

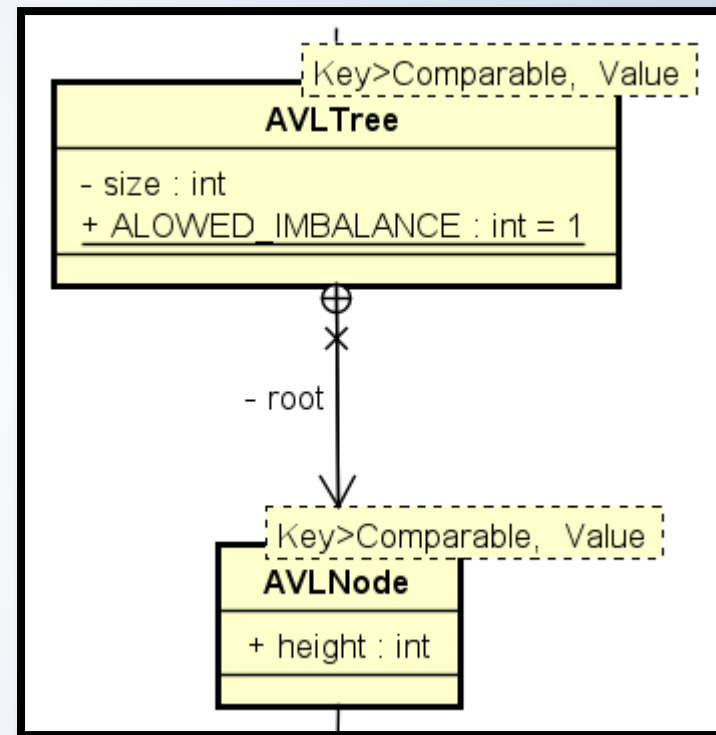
- Implementação de uma árvore binária de busca fundamental.



# Tabela de Símbolos

## Implementação: **AVLTree**

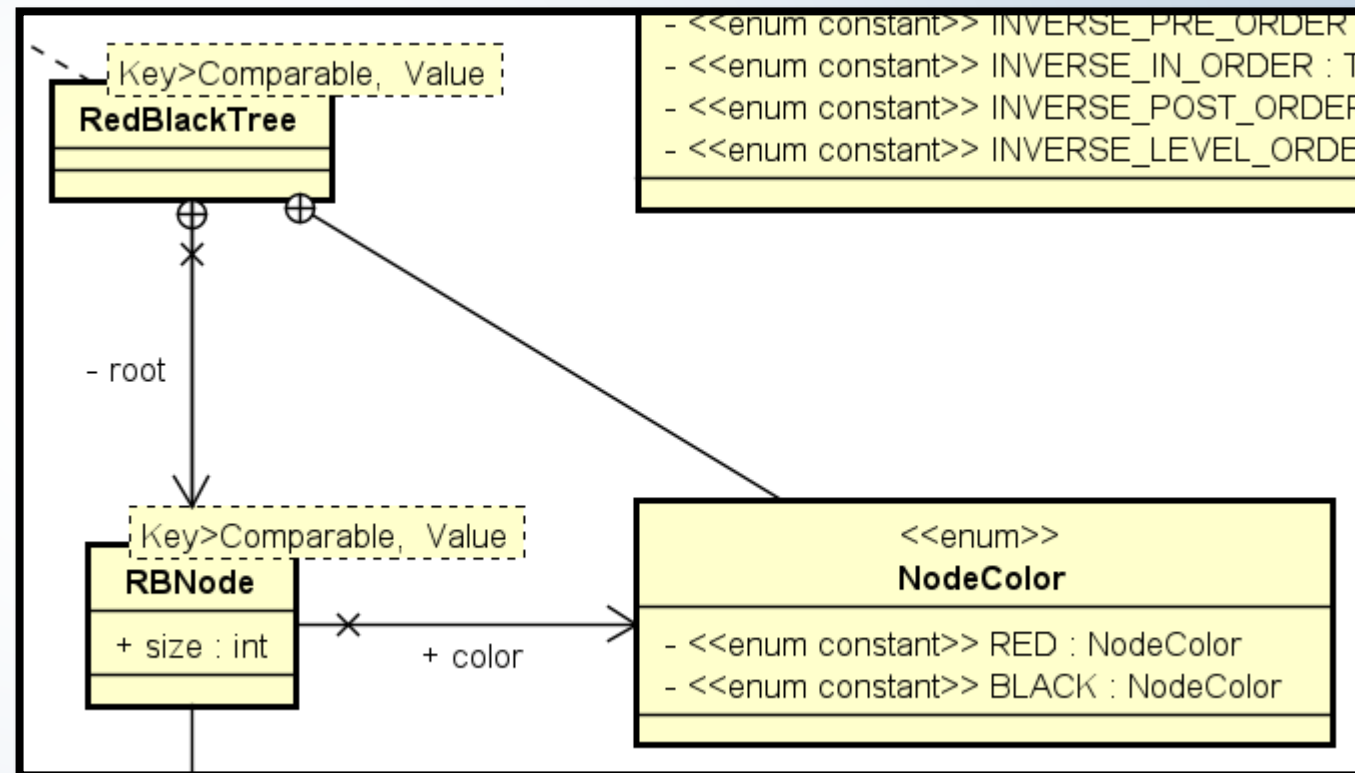
- Implementação de uma árvore AVL (Adelson-Velsky e Landis)



# Tabela de Símbolos

Implementação: **RedBlackTree**

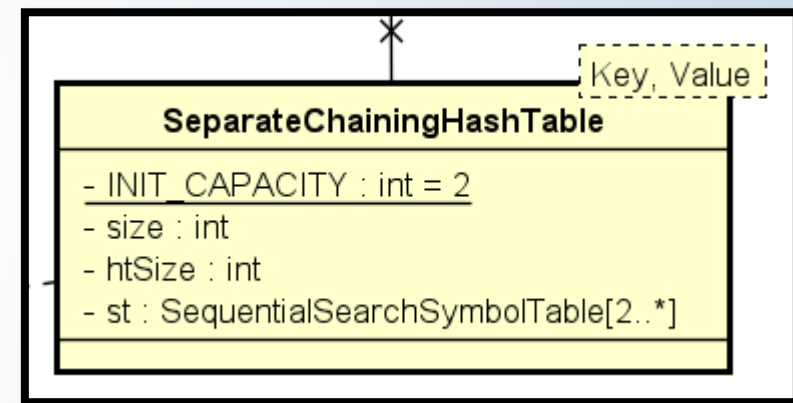
- Implementação de uma árvore vermelho-preto.



# Tabela de Símbolos

Implementação: **SeparateChainingHashTable**

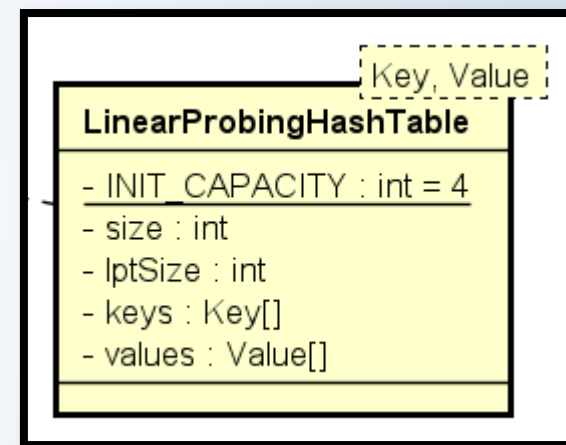
- Implementação de uma tabela de dispersão com encadeamento.



# Tabela de Símbolos

Implementação: **LinearProbingHashTable**

- Implementação de uma tabela de dispersão usando endereçamento aberto com sondagem linear (*linear probing*).





# Bibliografia

SEDGEWICK, R.; WAYNE, K. **Algorithms**. 4. ed. Boston: Pearson Education, 2011. 955 p.

GOODRICH M. T.; TAMASSIA, R. **Estruturas de Dados & Algoritmos em Java**. Porto Alegre: Bookman, 2013. 700 p.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Algoritmos – Teoria e Prática**. 3. ed. São Paulo: GEN LTC, 2012. 1292 p.