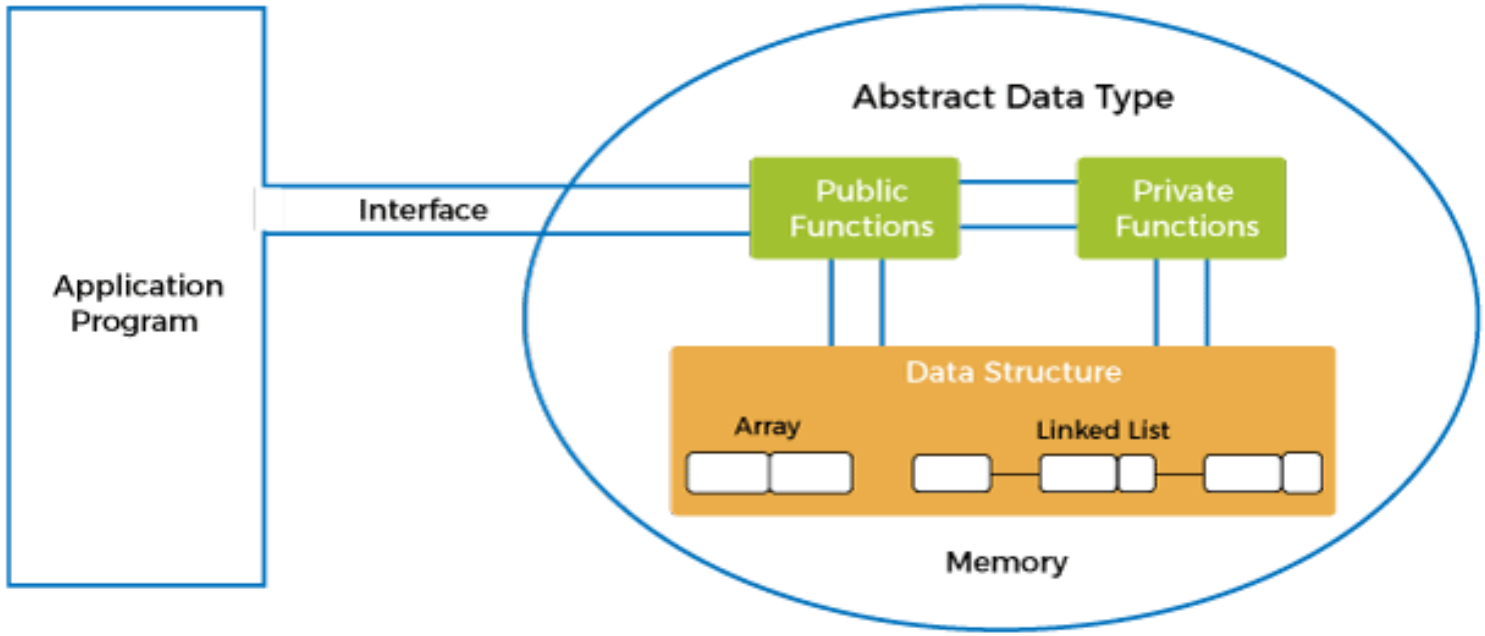
Estrutura de Dados II

Laboratório - TAD BST

Profs Jean M. Laine e Jamil K. N. Junior



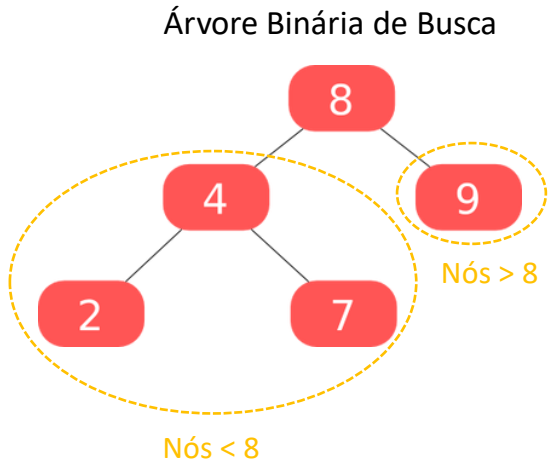
Fonte: https://www.javatpoint.com/abstract-data-type-in-data-structure

Para se considerar abstrato o tipo de dado, precisamos apenas conseguir aplicar as operações que foram definidas para ele (sem saber como foram implementadas ou como os dados foram armazenados). O importante é saber *o que* as operações fazem, quais seus parâmetros de entrada e o que geram de resultado.

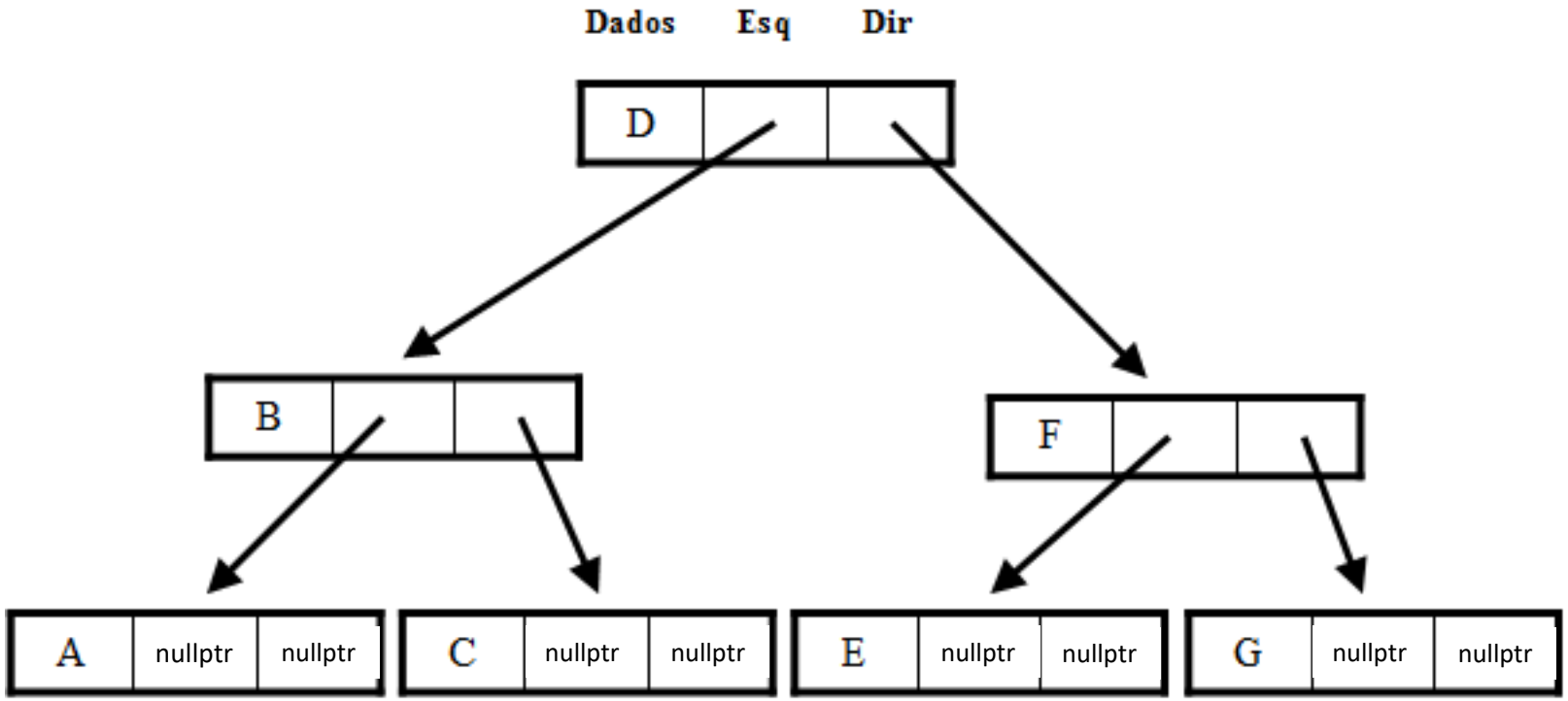
TAD (Tipo Abstrato de Dados)

Árvores Binárias de Busca (BST)

- As **Árvores Binárias de Busca** são árvores binárias com as seguintes **propriedades**:
  - Todos os nós da subárvore da *esquerda* de um nó V qualquer possuem um valor numérico *inferior* ao nó V
  - Todos os nós da subárvore da *direita* de um nó K qualquer possuem um valor *superior* ao nó K
- Essa propriedade deve ser válida para todas as subárvores da árvore!
- **Atenção**: por simplificação, não vamos considerar a possibilidade de existir elementos repetidos na árvore



Como modelar e implementar uma BST?



Exemplo

```
class node {
        char Dados;
        node *Esq;
        node *Dir;
....
....
}


public class BST {
   node *raiz;
   BST(){
     raiz = NULL;
    }
    void inserir(char key);
    void remove(char key);
    void print2D();
    node pesquisar(char key);
...
...
...
}
```

Código no Moodle

ArvoreBinBusca.cpp

Já está pronto

- ❑ classe no
- ❑ gets e sets
- ❑ Esqueleto da classe ArvoreBST
- ❑ Construtor da ArvoreBST
- ❑ Inserir(int chave)
- ❑ emOrdem()
- ❑ preOrdem()
- ❑ posOrdem()
- ❑ main para teste da BST

Atividade Entregável
(Valendo Nota de Lab)

Vocês irão utilizar o arquivo base *ArvoreBinBusca.cpp* e estendê-lo para implementar os seguintes métodos:

- ❑ pesquisarRec(int chave) //Versão Recursiva
- ❑ pesquisarIter(int chave) //Versão Iterativa
- ❑ qdeNos()
- ❑ alturaBST()
- ❑ min() //menor chave presente na BST
- ❑ max() //maior chave presente na BST
- ❑ folhas() //imprimir as chaves, em ordem crescente, dos nós folhas da BST
- ❑ removerFolha(int chave)

- Adicionar no menu de teste opções para testar e validar cada método implementado na BST.
- Entrega (código + print dos testes) e apresentação (obrigatória) na próxima aula.


Universidade Presbiteriana
Mackenzie
150 anos
1870 - 2020


Faculdade de
Computação e Informática
FCI
50
ANOS