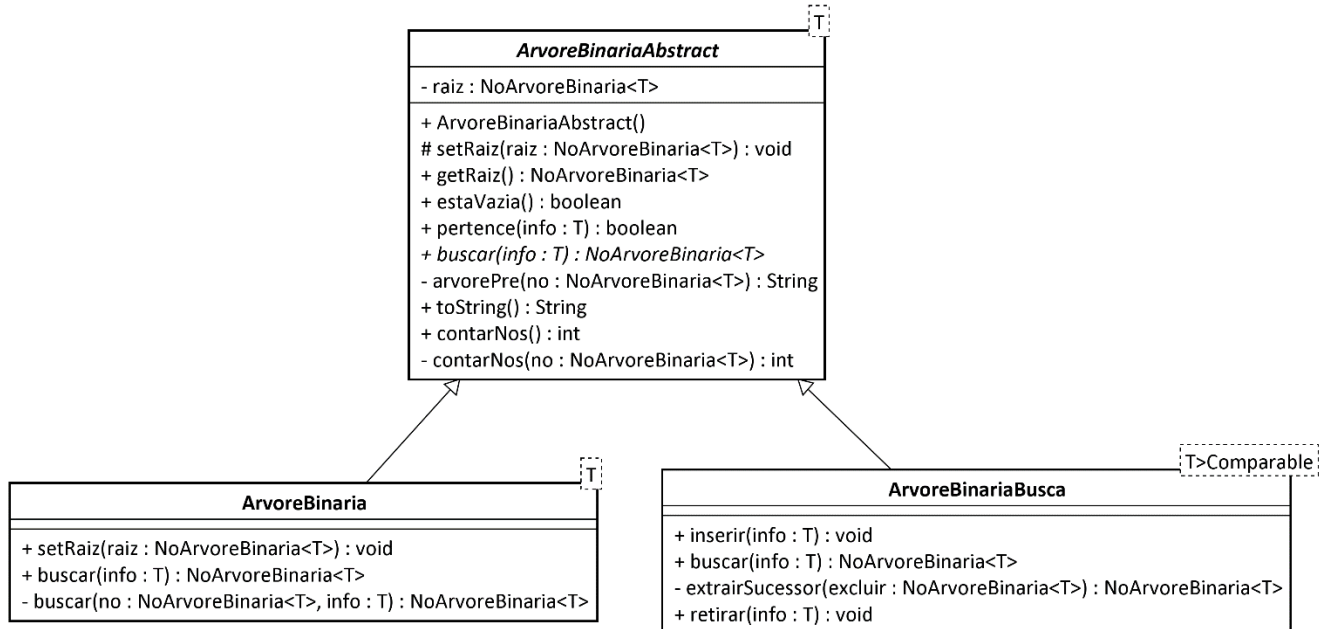


Lista de Exercício 11

Questão 1

O objetivo desta atividade prática é realizar a implementação de árvores binárias de busca, de acordo com o diagrama de classes da figura abaixo.



Para construir a solução, vamos aproveitar a implementação do exercício 8. Para isso, vamos modificar nossa implementação de árvore binária para que seja extensível. Veja as etapas:

- Copiar as classes **ArvoreBinaria** e **NoArvoreBinaria** do exercício 8 para um novo projeto;
- Renomear a classe **ArvoreBinaria** para **ArvoreBinariaAbstract**;
- Tornar a nova classe **ArvoreBinariaAbstract** uma classe abstrata;
- Criar o método abstrato **buscar()** na classe **ArvoreBinariaAbstract**;
- Implementar o método **pertence()** na classe **ArvoreBinariaAbstract** para reusar o método **buscar()**;
- Tornar o método **setRaiz()**, da classe **ArvoreBinariaAbstract**, protegido;
- Criar a classe **ArvoreBinaria** estendendo-a da classe **ArvoreBinariaAbstract**;
- Inserir o método **setRaiz()** na classe **ArvoreBinaria** e implementar seu código para reutilizar a implementação do método **setRaiz()** da superclasse.
- Implementar o método **buscar()** na classe **ArvoreBinaria**, para que localize se há um nó que armazene o dado fornecido como argumento. Em caso positivo, o método **buscar()** deve retornar este nó, caso contrário, deverá retornar **null**.

Os métodos novos para serem implementados na classe **ArvoreBinariaBusca** são:

- inserir()**: este método deve inserir o dado, fornecido como argumento, na árvore binária de busca. O método deve armazenar o dado num nó de forma que a árvore binária mantenha as características de uma “árvore binária de busca”.
- buscar()**: este método deve buscar o dado fornecido como argumento, na árvore binária, retornando o nó que o armazena. Utilizar o algoritmo de busca binária em árvore.
- retirar()**: este método deve retirar o dado fornecido como argumento, da árvore. Após a remoção, a árvore deve manter as características de uma “árvore binária de busca”.



Questão 2

Implemente o seguinte plano de testes.

Plano de testes PL01 – Validar funcionamento da implementação de árvore binária de busca			
Caso	Descrição	Entrada	Saída esperada
1	Conferir se o método inserir() mantém os dados armazenados adequadamente, mantendo a árvore com a característica de ser uma árvore binária de busca.	Criar uma árvore binária de inteiros e adicionar os seguintes dados, nesta ordem: 50,30,70,40,25,75,65,35,60	O método toString() deve retornar: <50<30<25<>>><40<35<>>>>><70<65<60<>>>>><75<>>>>>>>
2	Conferir se a árvore consegue remover um nó folha	Criar uma árvore inserindo os dados nesta ordem: 50,30,25,40 Remover o nó 40	O método toString() deve retornar: <50<30<25<>>>>>>>
3	Conferir se a árvore consegue remover nó com um filho	Criar uma árvore inserindo os dados nesta ordem: 80,52,90,48,71,63,67 Remover o nó 71	O método toString() deve retornar: <80<52<48<>>><63<><67<>>>>><90<>>>>>
4	Conferir se a árvore consegue remover um nó com dois filhos	Criar uma árvore inserindo os dados nesta ordem: 250,38,26,72,55,90,41,60,43,78,92,74 Remover o nó 38	O método toString() deve retornar: <250<41<26<>>><72<55<43<>>><60<>>>><90<78<74<>>>><92<>>>>>>>