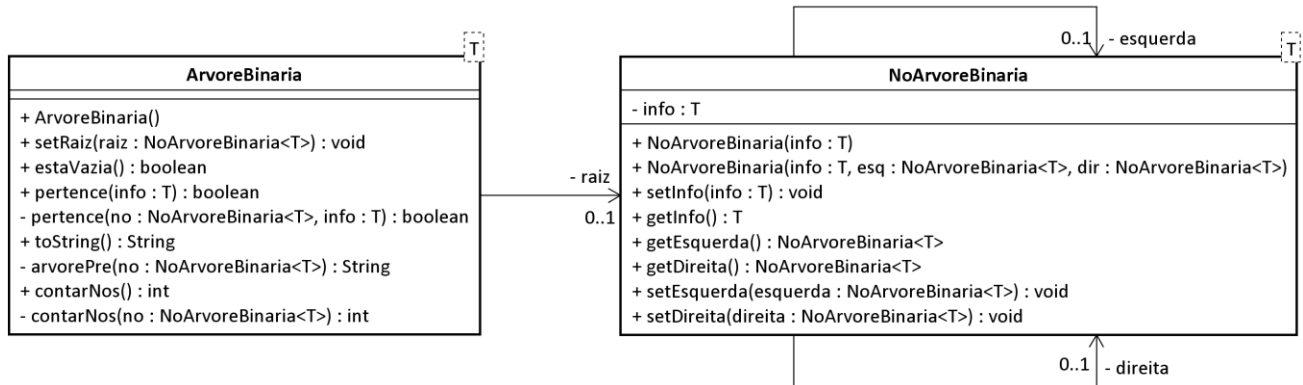


Lista de Exercício 07

Questão 1

O objetivo desta atividade prática é realizar a implementação de árvores binárias. Crie um projeto novo de acordo com o diagrama de classes da figura abaixo.

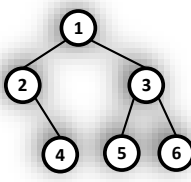


Os métodos a serem implementados na classe **ArvoreBinaria** são:

- **ArvoreBinaria()**: construtor da classe, que deve estabelecer que a árvore está vazia;
- **setRaiz()**: método *setter* do atributo *raiz* da classe **ArvoreBinaria**;
- **estaVazia()**: deve retornar **true** se a árvore estiver vazia ou **false** se não houver um ou mais nós na árvore;
- **pertence()**: este método deve procurar na árvore binária se determinado dado está armazenado na árvore. Implemente este método recursivamente, utilizando o método privado **pertence(NoArvoreBinaria, T)**;
- **toString()**: este método deve retornar o conteúdo da árvore, usando o percurso pré-ordem. Utilize o método **arvorePre(NoArvoreBinaria)** para implementar de forma recursiva este algoritmo;
- **contarNos()**: este método deve retornar a quantidade total de nós de uma árvore binária. Utilize o método privado **contarNos(NoArvoreBinaria)** para implementar de forma recursiva este algoritmo.

Questão 2

Implemente o seguinte plano de testes:

Plano de testes PL01 – Validar funcionamento da implementação estática de fila			
Caso	Descrição	Entrada	Saída esperada
1	Conferir se é reconhecida árvore vazia	Criar uma árvore binária de inteiros.	O método <code>estaVazia()</code> deve resultar em <code>true</code> .
2	Conferir se é reconhecida árvore não vazia	Criar uma árvore binária de inteiros e adicionar um nó contendo o valor 5.	O método <code>estaVazia()</code> deve resultar em <code>false</code> .
3	Conferir a representação textual da árvore usando o caminho pré-ordem	Criar uma árvore binária que represente esta árvore: 	O método <code>toString()</code> deve resultar em: <1<2<><4<><>>><3<5<><>>><6<><>>>>
4	Conferir se a árvore consegue	Dada uma árvore igual à do caso de	<code>Pertence()</code> deve resultar em <code>true</code> .



	avaliar se um valor armazenado na raiz pertence à árvore	testes 3, buscar o valor 1	
5	Conferir se a árvore consegue avaliar se um valor armazenado num nó não raiz e não folha pertence à árvore	Dada uma árvore igual à do caso de testes 3, buscar o valor 3.	Pertence() deve resultar em true.
6	Conferir se a árvore consegue avaliar se um valor armazenado num nó folha pertence à árvore	Dada uma árvore igual à do caso de testes 3, buscar o valor 6.	Pertence() deve resultar em true.
7	Conferir se a árvore conclui que determinado dado não pertence à árvore	Dada uma árvore igual à caso de testes 3, buscar o valor 10.	Pertence() deve resultar em false.
8	Verificar se o árvore consegue contar a quantidade de nós de uma árvore com altura igual à 2.	Dada uma árvore igual à caso de testes 3, contar a quantidade de nós.	contarNos() deve resultar em 6.