

## Atividade de Árvores Binárias de Busca

**Questão única:** Implemente o cadastro de produtos de uma empresa utilizando uma **árvore binária de busca**. Cada nó da árvore deve armazenar um objeto da classe Produto. Um objeto da classe Produto deverá ter os seguintes atributos: código (String), descrição (String), fornecedor (String), preço (double) e quantidade em estoque (int). O campo de ordenação da árvore é o código do produto. O programa deverá ter os seguintes sub-programas:

- (a) Procedimento para cadastrar um novo produto. O procedimento deverá solicitar ao usuário as informações do produto a ser cadastrado e inserir o produto no cadastro. Não serão aceitos mais de um produto com mesmo código.
- (b) Procedimento para exibir as informações (código, descrição, fornecedor, preço e quantidade em estoque) de todos os produtos cadastrados por ordem crescente de código.
- (c) Função para procurar um produto no cadastro. A função deverá receber como parâmetro o código do produto procurado e deverá retornar uma referência para o produto procurado ou null, caso não encontre. Esta função deverá ser utilizada pelos procedimentos dos itens (d), (e) e (f).
- (d) Procedimento para alterar o preço de um produto. O procedimento deverá receber como parâmetro o código do produto cujo preço será alterado. Este procedimento deverá utilizar a função definida no item (c).
- (e) Procedimento para alterar a quantidade em estoque de um produto. O procedimento deverá receber como parâmetro o código do produto cuja quantidade será alterada. Este procedimento deverá utilizar a função definida no item (c).
- (f) Procedimento para exibir as informações (código, descrição, fornecedor, preço e quantidade em estoque) de um produto. O procedimento deverá receber como parâmetro o código do produto cujas informações serão exibidas. Este procedimento deverá utilizar a função definida no item (c).

### OBSERVAÇÕES:

1. Este programa deverá ter, **no mínimo**, cinco classes: a classe Produto, a classe ABB, a classe ABBNode, a classe Cadastro e a classe da Aplicação.
2. A classe **Produto** é a classe da informação. Um código identifica um produto unicamente. Não podem existir dois produtos com um mesmo código.
3. A classe **Cadastro** tem um atributo que é uma ABB onde são “armazenados” os produtos. Isso significa, por exemplo, que cadastrar um produto é inserir o produto na árvore.
4. A classe **ABB** deve ter apenas os métodos que implementam as operações de árvore.
5. A classe **Aplicação** terá um método main a partir do qual a aplicação será executada.

### DICAS:

1. Comece criando a classe **Produto**. Ela deverá ter um método construtor, métodos gets e sets, um método toString (para permitir dar um print em um objeto desta classe) e um método compareTo (para permitir comparar dois objetos desta classe).
2. Depois crie as classes **ABBNode** e **ABB**. A classe ABB somente deverá ter os métodos de árvore (inserir, buscar, isEmpty, passeios).
3. Então crie a classe **Cadastro** que terá como atributo uma ABB de produtos. A classe Cadastro terá os métodos solicitados nos itens de (a) a (f).
4. Por último, crie a classe que será sua aplicação. No main, declare e crie o objeto da classe Cadastro e implemente o loop principal de manipulação do cadastro.