

Este terceiro trabalho de programação em Java consiste na implementação parcial de uma árvore binária de pesquisa para armazenar números inteiros, que está descrita no livro *Estruturas de Dados e Algoritmos em Java*, de António Adrego da Rocha publicado pela editora FCA em 2011.

Pretende-se implementar o tipo de dados **ABPTree** utilizando programação defensiva com lançamento de exceções no caso de situações de dados anormais que violam as pré-condições e/ou as pós-condições dos métodos. Para esse efeito é disponibilizada duas classes de excepção designadas por *EmptyTreeException* e *RepeatedElementException* para indicar, respetivamente, quando uma árvore está vazia e quando um elemento já existe na árvore. A outra classe de excepção utilizada *NoSuchElementException* já existe na linguagem Java.

1. Comece por ler a descrição da classe **ABPTree** (documento **ABPTree.html**). Depois implemente as funções privadas auxiliares que fazem efetivamente as travessias recursivas da árvore: *recSumTotal*; *recCountEnd5*; *recSumEven*; *recCountMult3*; e *recCountLefties*. Em caso algum deve alterar os restantes métodos. **Tenha em atenção que esta classe dá erro de compilação porque a assinatura (protótipo) da última função está incompleta.**
2. Pode utilizar o programa de simulação **SimABPInteger.java** para testar os algoritmos de inserção, remoção e pesquisa de elementos numa árvore binária.
3. Utilize o programa de testes unitários **TestABPInteger.java** para verificar se os seus métodos estão a dar os resultados corretos.

O trabalho deve ser entregue até 30 de Junho de 2022. Apenas um dos alunos do grupo deve enviar por email para adrego@ua.pt um arquivo zip contendo apenas o ficheiro final da classe: **ABPTree.java**.

Para quaisquer questões de Java pode consultar o livro *Estruturas de Dados e Algoritmos em Java*. Também pode e deve contactar-me por *email* com dúvidas acerca do trabalho.