

---

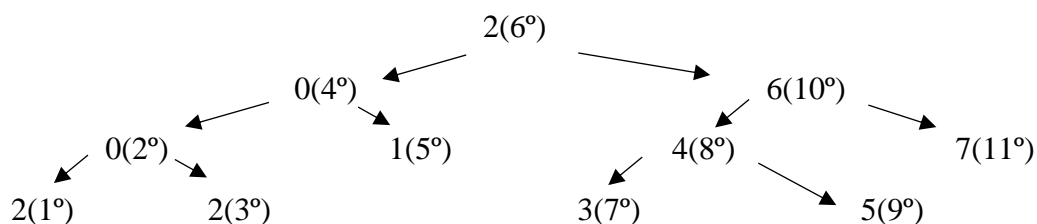
### Instruções:

- O trabalho é individual.
  - Deve ser entregue até o final do dia 05/02/2020.
  - Deve ser submetido o código fonte de cada questão.
  - Cada questão deve vir acompanhada por um vídeo explicando o que foi desenvolvido.
  - O código fonte e os vídeos devem ter no máximo 50mb.
  - Cada questão vale 2 pontos de um total de 10. A nota desse trabalho corresponde a 50% da nota da primeira unidade.
- 

### Questão 1:

Modifique o construtor da classe Tree para representar uma árvore armazenando os dígitos de sua matrícula no seguinte formato: o 6º dígito deve ser a raiz tendo como filhos o 4º e 10º dígitos. O 4º deve ter como filhos o 2º e 5º dígitos e o 10º dígito deve ter como filhos o 8º e 11º dígitos. o 1º dígito deve ter como filhos o 1º e 3º dígitos e o 8º dígito deve ter como filhos o 7º e 9º dígitos.

Suponha a matrícula 20201234567, a árvore correspondente seria:



Modifique o método toString dos nós para retornar uma String representando o valor armazenado, por exemplo, para 5 deve retornar "5" e depois efetue a o percurso em ordem (inOrder).

Restrição: cada dígito deve ser representando por um objeto diferente.

Vídeo: deve explicar o código e exibir o resultado da execução do percurso em ordem.

## Questão 2:

Modifique o método `toString` da classe `Tree` para imprimir os valores armazenados na árvore considerando o percurso em ordem (`inOrder`) exibindo o conteúdo de cada nó precedido por uma quantidade de tabulações igual ao nível em que ele se encontra na árvore. Também deve ser exibido o nível associado a cada valor. Os valores devem ser impressos em linhas sequenciais diferentes. Para a árvore do exercício anterior, obtemos a seguinte impressão (observe que o primeiro valor impresso é o do nó mais à direita):

```
    7 [Level:3]
  6 [Level:2]
    5 [Level:4]
    4 [Level:3]
    3 [Level:4]
2 [Level:1]
  1 [Level:3]
  0 [Level:2]
    2 [Level:4]
    0 [Level:3]
    2 [Level:4]
```

Restrição: a implementação deve ser recursiva sem uso de estruturas de repetição como `for`, `while`, `do`, etc.

Vídeo: deve explicar o código e exibir o resultado da impressão.

## Questão 3:

Desenvolva uma aplicação que recebe como entrada um diretório de saída e um diretório de percurso. A aplicação deve efetuar um percurso na árvore de diretórios iniciando pelo diretório de percurso e deve mover todos os arquivos de extensão `.png` para o diretório de saída. Utilize o método `renameTo` da classe `File` para efetuar a movimentação dos arquivos.

Vídeo: deve explicar o código, exibir a estrutura de diretórios antes e depois da execução.

## Questão 4:

A classe NodeN representa uma implementação de árvores n-ária, isto é, representa uma árvore em que cada nó possui uma quantidade qualquer de filhos. Implemente classe Tree como uma composição da classe NodeN e adicione um método para efetuar o percurso em pré-ordem de forma iterativa conforme o seguinte algoritmo:

```
Seja P uma pilha de nós
Insira a raiz da árvore em P
Enquanto houver elementos em P faça:
    Seja Node o elemento no topo de P
    Remove Node da pilha
    Visite o conteúdo do nó Node
    Insira cada um dos filhos de Node em P iniciando pelo
último
```

Vídeo: devem explicar o código e apresentar um exemplo de execução. As partes dos algoritmos devem ser destacadas na explicação do código.

## Questão 5:

Na classe da árvore n-ária do item anterior, adicione um método que retorna uma árvore binária correspondente aos nós armazenados. A implementando deve considerar o seguinte algoritmo que converte um nó da árvore n-ária em um nó da árvore binária:

```
Seja Tb um nó de uma árvore binária
Seja Tn um nó de uma árvore n-ária com n filhos
Tb armazena o mesmo conteúdo de Tn
O filho a esquerda Fe de Tb é o primeiro filho de Tn
Seja Fr o filho a direita de Fe
Para i de 2 até n faça
    Fr armazena o mesmo conteúdo que Tni
    Fr aponta agora para seu filho a direita
Retorne Tb
```

Restrição: a árvore binária deve conter somente nós binários e o algoritmo deve ter complexidade  $O(n)$  ou melhor.

Vídeo: deve explicar o código e apresentar a execução de um exemplo. As partes dos algoritmos devem ser destacadas na explicação do código.