**Lab05 - Binární strom**

Implementujte dodané [interface](https://cw.fel.cvut.cz/wiki/_media/courses/b0b36pjv/hw/pjv-lab05.zip) Tree a Node třídami TreeImpl a NodeImpl. Třída TreeImpl musí obsahovat defaultní konstruktor (bez parametrů). Metody a proměnné pojmenovávejte anglicky. Nepoužívejte javovské kolekce; potřebujete pouze pole, které dostanete jako parametr setTree.

Implementované třídy TreeImpl a NodeImpl umístěte do stejného balíčku jako jsou dodané interfacy.

Tree reprezentuje [binární strom](http://en.wikipedia.org/wiki/Binary_tree), který ve všech uzlech obsahuje celočíselná data. Každý uzel stromu je reprezentován třídou implementující interface Node. Tree obsahuje následující metody:

* void setTree(int[] values)
  + nastaví strom, tak aby obsahoval hodnoty z pole values
  + pokud je délka pole lichá, kořen obsahuje prostřední číslo, jinak obsahuje první číslo za polovinou posloupnosti (na jejich hodnotách tedy nezáleží, pouze na jejich pozici v poli)
  + levá část podstromu pak obsahuje prvky pole před tím prostředním prvkem a pravé prvky za ním
  + obdobně to platí i pro podstromy
* Node getRoot()
  + vrátí kořen stromu
* String toString()
  + vrátí řetězcovou reprezentaci stromu vhodnou k výpisu v následujícím formátu
    - každá hodnota je na jednom řádku, předchází ji počet mezer odpovídající hloubce uzlu (0 pro kořen) a '- '
    - na prvním řádku je hodnota kořenu
    - hodnotu uzlu následuje výpis levého podstromu a pak pravého podstromu
    - každý řádek (vč. posledního) je ukončen novým řádkem ('\n')
  + Příklad pro strom vytvořený pro pole [1, 2, 3, 4, 5, 6, 7]:

- 4

- 2

- 1

- 3

- 6

- 5

- 7

Cílem není implementovat binární *vyhledávací* strom; na samotných hodnotách tedy **nezáleží**, pouze na jejich pořadí ve vstupním poli!

Ukázka výstupu metody toString pro stromy vytvořené z posloupností [1], [1, 2], … , [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

- 1

- 2

- 1

- 2

- 1

- 3

- 3

- 2

- 1

- 4

- 3

- 2

- 1

- 5

- 4

- 4

- 2

- 1

- 3

- 6

- 5

- 4

- 2

- 1

- 3

- 6

- 5

- 7

- 5

- 3

- 2

- 1

- 4

- 7

- 6

- 8

- 5

- 3

- 2

- 1

- 4

- 8

- 7

- 6

- 9

- 6

- 3

- 2

- 1

- 5

- 4

- 9

- 8

- 7

- 10

Odevzdávejte následující soubory: NodeImpl.java, TreeImpl.java