

## Bonusová domáca úloha č. 2

(Termín odovzdania úlohy: **do utorka 16. apríla 2024, 9:50**, t. j. do začiatku deviatych cvičení.)

Priložený ZIP archív obsahuje balík `graphs` s triedami pre grafy z prednášky a s hotovou generickou triedou `Node<T>` reprezentujúcou uzol zakoreneného stromu uchováajúci hodnotu typu `T`. Trieda `Node<T>` poskytuje konštruktory umožňujúce vytvoriť uzol s danou hodnotou a prípadne aj s danými synmi. Ďalej poskytuje metódu `getValue`, ktorá vráti hodnotu typu `T` uloženú v danom uzle, metódu `getChildren`, ktorá vráti nemodifikovateľný pohľad na zoznam synov daného uzla a metódu `addChild` umožňujúcu pridať nového syna daného uzla. V triede je tiež prekrytá metóda `toString` tak, aby vracala vhodnú textovú reprezentáciu stromu zakoreneného v danom uzle.

Doprogramujte do balíka `graphs` triedu `GraphSearchTrees`, ktorá bude pre daný orientovaný graf konštruovať stromy prehľadávania do hĺbky resp. do šírky z daného počiatočného vrcholu. Táto trieda by mala poskytovať:

- Konštruktor `public GraphSearchTrees(DirectedGraph g)`. Jeho argumentom je orientovaný graf, ktorého prehľadávania sa budú realizovať.
- Metódu

```
public Node<Integer> getDepthFirstSearchTree(int startingVertex),
```

ktorá pre graf `g` z argumentu konštruktora vráti koreň *stromu* jeho *prehľadávania do hĺbky* s počiatočným vrcholom `startingVertex`. Strom prehľadávania do hĺbky je definovaný ako na prednáške, pričom hodnotami uloženými v jeho uzloch sú čísla vrcholov navštívených prehľadávaním. Prehľadávanie grafu `g` do hĺbky by následníkov každého vrcholu `v` malo spracúvať v rovnakom poradí, v akom ich prechádza iterátor cez `g.outgoingEdgesDestinations(v)`.

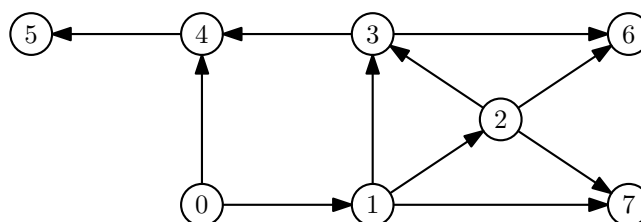
- Metódu

```
public Node<Integer> getBreadthFirstSearchTree(int startingVertex),
```

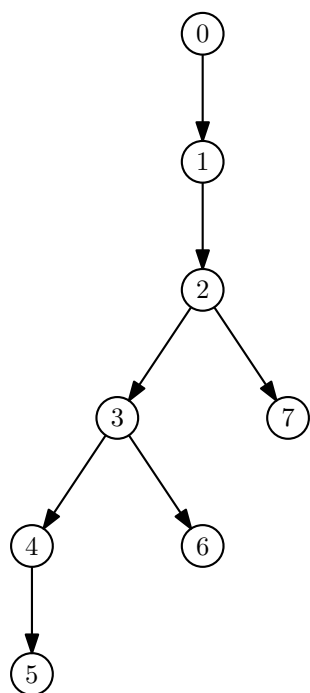
ktorá pre graf `g` z argumentu konštruktora vráti koreň *stromu* jeho *prehľadávania do šírky* s počiatočným vrcholom `startingVertex`. Strom prehľadávania do šírky je definovaný ako na prednáške a následníci každého vrcholu `v` grafu `g` by sa mali spracúvať v rovnakom poradí, v akom ich prechádza iterátor cez `g.outgoingEdgesDestinations(v)`.

Na testovač odovzdávajte iba súbor `GraphSearchTrees.java` obsahujúci zdrojový kód vašej triedy balíka `graphs`. Pri tvorbe kódu rešpektujte konvencie jazyka Java a elementárne zásady objektovo orientovaného programovania.

**Príklad.** Uvažujme graf na nasledujúcom obrázku a inštanciu `gst` triedy `GraphSearchTrees` vytvorenú pre tento graf.



Výstupom metódy `gst.getDepthFirstSearchTree(0)` je potom koreň stromu na nasledujúcom obrázku.



Podobne výstupom metódy `gst.getBreadthFirstSearchTree(0)` je koreň stromu na nasledujúcom obrázku.

