

## Cvičenia č. 2, úloha č. 2

Stiahnite si kostru k tejto úlohe, ktorá obsahuje čiastočnú implementáciu binárnych vyhľadávacích stromov zo zimného semestra pomocou objektov v jazyku Java. Pripomeňme si, že binárny vyhľadávací strom je špeciálny binárny strom, ktorého uzlom sú priradené (v našom prípade) celočíselné hodnoty, tzv. kľúče. Pre každý uzol binárneho vyhľadávacieho stromu s kľúčom `key` pritom platí, že v jeho ľavom podstrome sú uzly s kľúčmi menšími alebo rovnými `key` a v pravom podstrome uzly s kľúčmi väčšími alebo rovnými `key`. Minulý semester ste videli viacero operácií na binárnych vyhľadávacích stromoch. Z nich sú v priloženej kostre implementované metódy na pridanie uzla do binárneho vyhľadávacieho stromu a na vyhľadanie uzla binárneho vyhľadávacieho stromu podľa jeho kľúča (nie je implementované mazanie uzlov). Vašou úlohou bude doimplementovať ďalšie dve jednoduché operácie.

Kostra pozostáva z troch tried, ktoré sú súčasťou spoločného balíka `bst`:

- Trieda `Node` reprezentuje uzol binárneho vyhľadávacieho stromu. Do tejto triedy budete dopĺňať váš kód.
- Trieda `BinarySearchTree` reprezentuje samotný binárny vyhľadávací strom pomocou jeho koreňa. Táto trieda je už hotová a poskytuje niekoľko metód, ktoré sú väčšinou implementované pomocou volania metód triedy `Node` (vrátane tých, ktoré sú určené na dorobenie).
- Trieda `CommandInterpreter` obsahuje hotový kód metódy `main`, ktorá načítava z konzoly príkazy vzápätí interpretované ako volania metód pre inštanciu binárneho vyhľadávacieho stromu. Výsledky volaní jednotlivých metód sú priebežne vypisované na konzolu. Ukážkový vstup a výstup možno nájsť nižšie.

Pred riešením samotnej úlohy môže byť užitočné pochopiť implementáciu už hotových častí.

Doimplementujte do triedy `Node` telá nasledujúcich dvoch metód:

- Metódy `public Node minNode()`, ktorá pre uzol reprezentovaný danou inštanciou triedy `Node` vráti minimálny uzol podstromu zakoreneného v tomto uzle (čiže uzol s minimálnou hodnotou kľúča; ak je takých uzlov viac, môže vrátiť ľubovoľný z nich).
- Metódy `public int[] inorderKeys()`, ktorá vráti pole kľúčov jednotlivých uzlov podstromu zakoreneného v danom uzle v poradí `inorder` (poradie kľúčov tak bude neklesajúce; ide teda vlastne o implementáciu triedenia `TreeSort`).

Veľmi podobné funkcie ste už implementovali minulý semester v `C/C++`.

Na testovač odovzdávajte iba súbor `Node.java` obsahujúci zdrojový kód vami upravenej triedy `Node`. Triedy `BinarySearchTree` a `CommandInterpreter` k nej budú na testovači priložené a spúšťať sa na testovacích vstupoch bude trieda `CommandInterpreter`.

### Príklad vstupu:

```
add 2
add 4
contains 3
add 5
minKey
contains 2
inorderKeys
```

### Príklad výstupu:

```
Pridavam uzol s klucom 2.
Pridavam uzol s klucom 4.
Strom neobsahuje uzol s klucom 3.
Pridavam uzol s klucom 5.
Minimalna hodnota kluca v strome: 2.
Strom obsahuje uzol s klucom 2.
Kluce stromu v poradí inorder: 2 4 5
```