

# Dokumentace - implementace AVL stromů

Zápočtový program z cvičení ADS I.

## Podstata programu

Implementace datové struktury AVL stromu jako třídy v programu C#. Jádrem řešení jsou funkce Insert, Delete a Find, dále je obsažena metoda Vypis, která slouží pro co nejpřehlednější výpis struktury stromu včetně příznaků vyváženosti jednotlivých vrcholů.

Řešení se skládá ze souborů AVLtree.cs - vlastní implementace - a Program.cs, což je konzolová aplikace demonstrující funkce třídy AVLstrom.

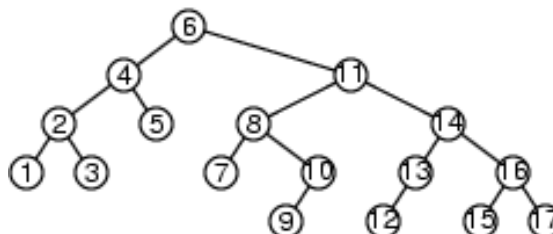
## Ovládání aplikace

Program se ovládá následujícími příkazy zadávanými do konzole:

- **i** *[x]* - vložení čísla *x* do stromu, např. "i 5". Pokud *x* již ve stromě je, nestane se nic.
- **d** *[x]* - smazání čísla *x* ze stromu, např. "d 5". Pokud *x* ve stromě není, nestane se nic.
- **f** *[x]* - nalezení čísla *x* ve stromě, vrátí "*Klic je ve strome.*" nebo "*Nenalezeno.*"
- **v** - výpis struktury stromu. Jednotlivá patra jsou za sebou ve sloupcích, synové jsou napravo od svého otce, pravý syn jako první, levý pod ním. U každého vrcholu je příznak jeho vyváženosti (+, - nebo °) Např. strom na obrázku by byl vypsán v následujícím formátu:

```

6+   11°  14°  16°  17°
      15°
      13-
      12°
      8+   10-
      7°   9°
      4-   5°
          2°  3°
          1°
  
```



Formátování výstupu funguje správně pro klíče menší než 1000000.

- **h** - nápověda
- **q** - ukončení programu

## Diskuze

Cílem bylo vytvořit datovou strukturu stromu, na které by funkce přidání i odebrání vrcholu byly instanční, tedy není nutné pracovat "zvnějšku" pomocí statické funkce. Z toho důvodu bylo třeba aplikovat několik úprav.

První vrchol stromu je vždy tzv. **hlava**. Čistě pomocná konstrukce slouží především k tomu, abychom mohli provádět rotace i na kořeni, jinými slovy umožňuje překořenění stromu. Technicky je hlava rodičem kořene, přičemž kořen je jejím pravým synem.

Každý vrchol obsahuje nejen ukazatele na své dva syny, ale i na svého předka. Tato úprava značně zjednodušuje úpravy po operacích Insert a Delete, kdy by jinak bylo nutné předávat si předka jako parametr. Cenou je nutnost několika přepojení ukazatelů navíc při odebírání vrcholů nebo při rotaci.

Vrchol naopak neobsahuje informaci o hloubkách levého a pravého podstromu, celou dobu si vystačíme pouze s příznakem vyváženosti. Jedinou komplikací z toho vzešlou je nutnost "manuálního" přepisování příznaků při rotacích (protože pro přímý výpočet bychom hloubky potřebovali). Naštěstí možností není příliš mnoho a tato drobná nepříjemnost je tedy únosná.

Funkce Insert i Delete mají návratovou hodnotu typu boolean, kterou vrací informaci o tom, zda se hloubka podstromu změnila (zvýšila u Insertu, snížila u Deletu). Toto umožňuje volání funkce rekurzivně (tedy zanořování do stromu, dokud nenajdeme hledané umístění) a následné zpětné propagování signálu o změně hloubky po přidání/odebrání vrcholu stromem nahoru ke kořeni, přičemž cestou přepisujeme příznaky a případně provádíme odpovídající rotace.

Samotné mazání vrcholu probíhá stejně, jako u obyčejného BVS - tedy pokud daný vrchol je list nebo má jen jednoho potomka, jednoduše jej smažeme (a přepojíme syna). Pokud má oba syny, najdeme vrchol s nejvyšší hodnotou v levém podstromě / s nejnižší hodnotou v pravém podstromě (volíme podle příznaku vrcholu, abychom se vyhnuli případnému vyvažování), ten smažeme a jeho klíč dosadíme do původně mazaného vrcholu.