

**Fakulta riadenia a informatiky**  
Žilinská univerzita v Žiline

# Algoritmy a údajové štruktúry 2

Knižničný informačný systém

Ivana Lieskovská  
556206  
Piatok 8:00  
ZS 2016

# Problém

Knižničný informačný systém obsahuje veľké množstvo dát o knihách, výpočiskách a čitateľoch, vyžaduje rýchle vyhľadávanie v týchto dátach podľa rôznych kľúčov a zároveň relatívne nízku pamäťovú náročnosť. Vzhľadom na komplexnosť väzieb medzi jednotlivými objektmi a historické uchovávanie údajov (minulé výpožičky) je potrebné implementovať údajovú štruktúru, ktorá by vhodným spôsobom spĺňala tieto požiadavky.

## Použité údajové štruktúry a ich zložitosť

Použitá údajová štruktúra bola AVL strom, ktorý je vhodný pre časté vyhľadávanie najmä vďaka logaritmickému zložitosti a samovyvažovaniu.

Pre vyhľadávanie v štruktúrach (nezreťazene) podľa kľúča (názov knihy, isbn alebo meno čitateľa) je výpočtová zložitosť  $O(\log n)$ . Tento prípad je realizovaný napríklad pri vyhľadaní čitateľa v štruktúre obsahujúcej všetkých čitateľov knižničnej siete.

Pre viacúrovňové vyhľadávanie (strom ako atribút v uzle stromu) je náročnosť výpočtu možné zapísať ako súčet logaritmov  $O(\log n) + O(\log m)$ , kde  $n$  je počet uzlov v primárnej štruktúre a  $m$  je počet uzlov v sekundárnej štruktúre. Tento prípad znázorňuje napríklad vyhľadávanie knihy podľa názvu v konkrétnej pobočke knižničnej siete.

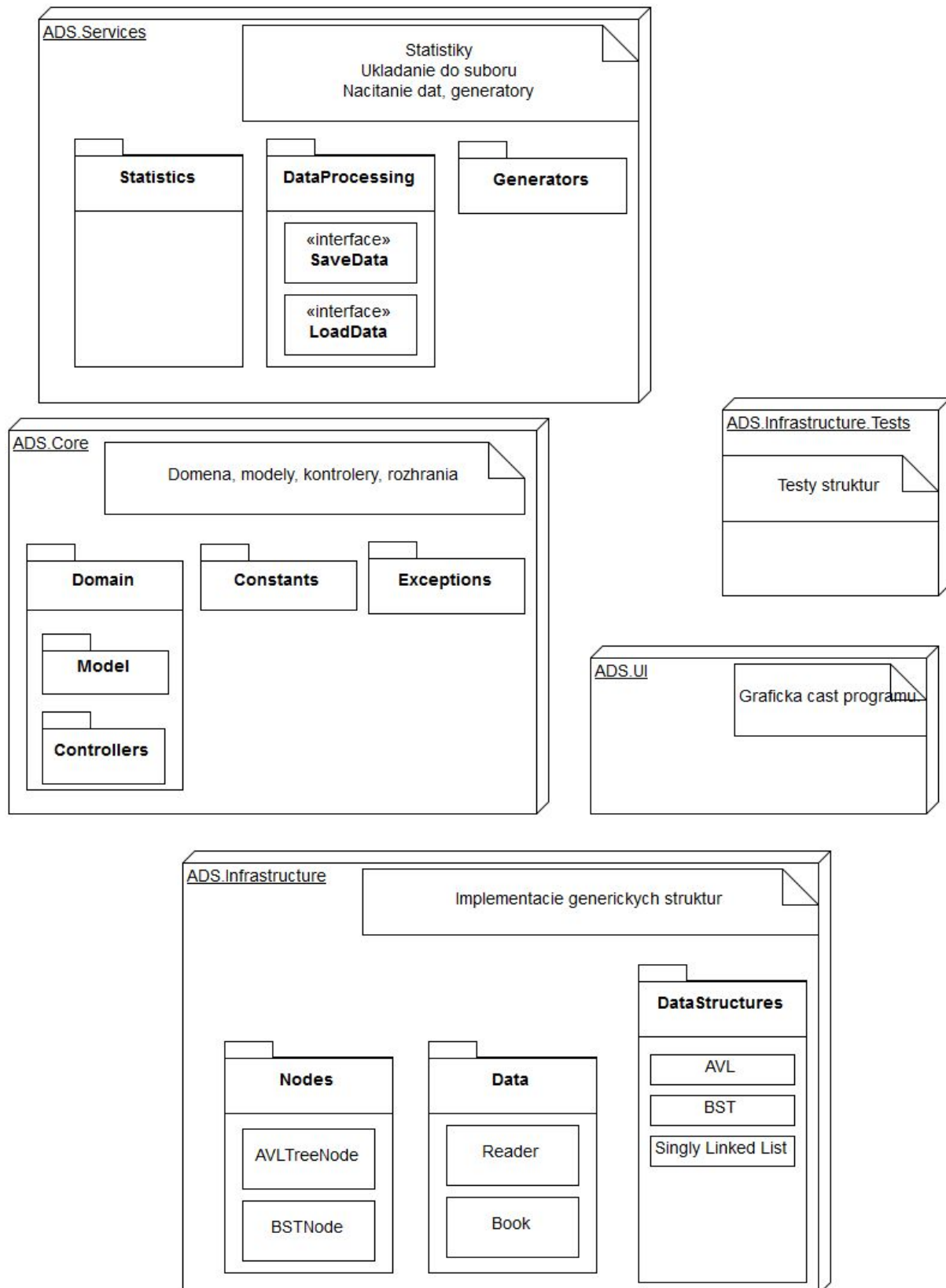
Pre vyhľadávanie a následne prejdienie celej sekundárnej štruktúry (strom ako atribút v uzle stromu) je zložitosť možné popísať ako súčet logaritmického zložitosti a konštanty, teda  $O(\log n) + k$ , kde  $n$  je počet prvkov v primárnej štruktúre a  $k$  je počet prvkov v sekundárnej štruktúre. Tento prípad je napríklad aplikovaný pri výpise všetkých výpožičiek konkrétneho čitateľa.

## Architektúra systému

Ako štandardný architektonický model bol použitý model MVC architektúry. Okrem balíkov tried rozdelených na časti Core (Model, Controllers, Constants, Exceptions) obsahuje aplikácia časť Infrastructure, ktorá obsahuje konkrétne implementácie údajových štruktúr a triedy objektov zo zadanej domény (Book, Reader, Library). Nemenej podstatná časť je balíček Services, ktorá implementuje rozhrania na spracovanie dát (načítanie, zápis a generovanie dát).

Balíček UI implementuje používateľské rozhranie tak, aby bolo nezávislé od implementácie časti modelu. Využíva jednoduché rozhranie ILibraryController.

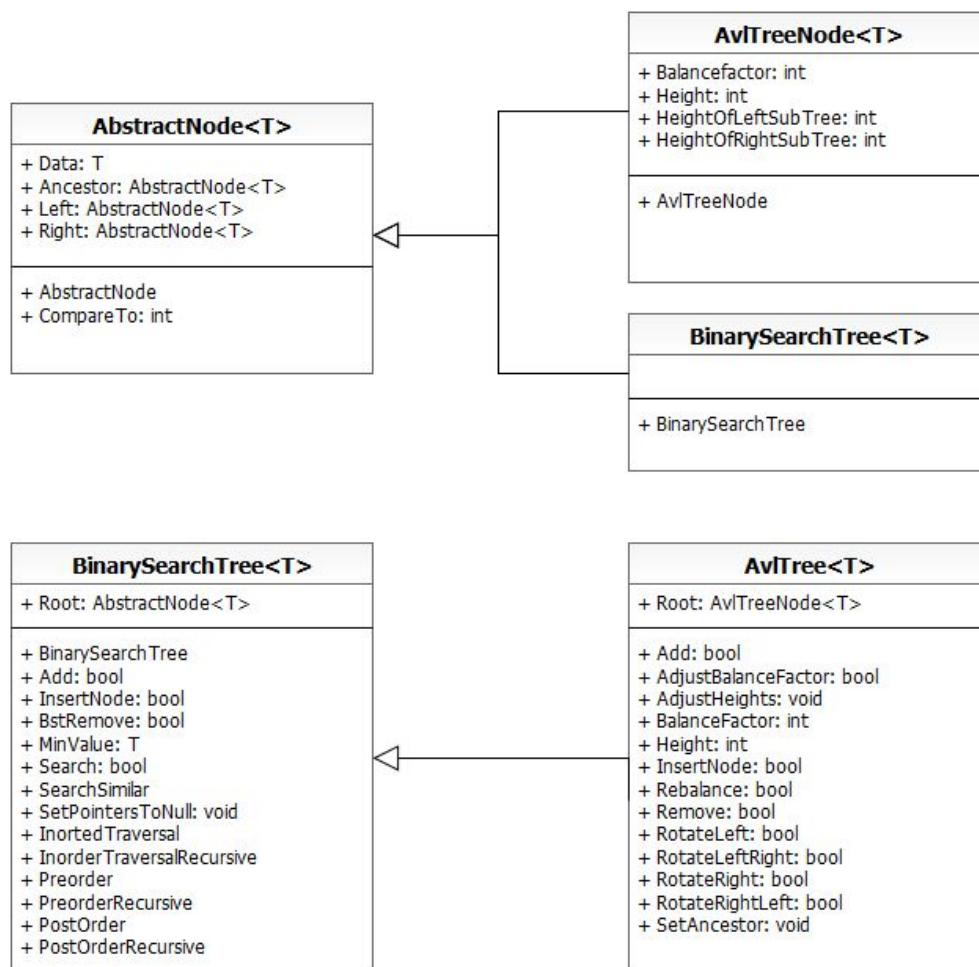
V poslednom balíku tried je obsiahnuté veľké množstvo Unit Testov jednotlivých údajových štruktúr.



Obrázok 1: Návrh architektúry knižničného informačného systému

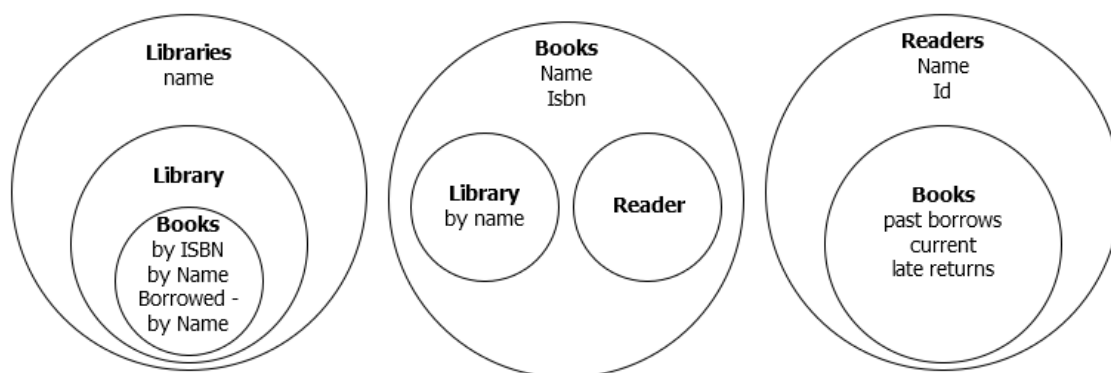
## Podrobnejší popis implementácie

AVL strom je binárny strom rozšírený o samovyvažovacie metódy. Pre väčšiu modularitu kódu bol preto binárny vyhľadávací strom implementovaný samostatne a strom AVL je potomkom BSV. Okrem toho sú aj uzly BVS a AVL stromov navrhnuté tak, aby bol systém čo najviac a najjednoduchšie rozšíriteľný bez zásahov na mnohé miesta kódu. podrobnejší náčrt poskytuje Obrázok 2.



Obrázok 2: AVL strom a BST

Ďalší obrázok znázorňuje praktické využitie AVL stromu v tejto práci. Každý kruh predstavuje jeden strom. menšie kruhy vo väčších kruhoch znamenajú atribúty stromov, sú to ďalšie stromy uložené v listoch nadradenej štruktúry.



Obrázok 3: Les AVL stromov

## Záver

Cieľom práce bolo naimplementovať knižničný informačný systém s využitím zložitejšej údajovej štruktúry. Zaujímavým riešením mohlo byť aj dodatočné implementovanie znakového stromu (trie) pre vyhľadávanie podľa reťazcov či iných údajových štruktúr.

## Použitá literatúra pri štúdiu

BARNETT, G., DEL TONGO, L.: Data Structures and Algorithms (DSA): Annotated Reference with Examples, NETSlackers (2008)

(Internetové stránky)

[www.tutorialspoint.com](http://www.tutorialspoint.com)

[www.geeksforgeeks.org](http://www.geeksforgeeks.org)

Poznámka:

Použité test cases pre vkladanie a mazanie AVL dát sú priložené k dokumentácii.