

Navrh knihovny pro parsovani ini souboru:

Cela knihovna se sestava z tri dulezitych trid a nekolika dalsich pomocnych trid.

Nejprve se zamerme na trojici nejdulezitejsich trid.

IniParser:

Jedna se o kontaktni interface mezi uzivatelem a knihovnou a jedna instance reprezentuje jeden konfiguracni soubor. Jeho ukolem je zprostredkovavat moznosti definice jednotlivych voleb a sekci a pristupu k nim. Stejne tak je jeho ukolem zajistovat jednotlivy druhy nacisti a vypisovani konfigurace.

Definice struktury konfiguracniho souboru probiha pomoci volani definicnich metod. Pro definovani jednotlivych voleb jsme se rozhodli s ohledem na pohodlnost pouziti zavest metody pro definici primo na zaklade nazvu sekce a nazvu definovane volby.

Pro praci s hodnotami voleb jsme podobne metody nakonec zavrli z toho duvodu, ze uzivatel by mel mit moznost si otestovat, zda uz volba existuje, nebo ji musi teprve definovat. Proto pristup probiha s jednim mezikrokem, behem nehoz si uzivatel pomoci najde metodou `getOption` prislusnou volbu a pak pristupuje primo k ni.

Dale je zde moznost (a vlastne nutnost) definovat vycetove typy pouzivane v konfiguracnim souboru a jejich pripustne hodnoty.

IniSection:

Jde o interface jedne sekce konfiguracniho souboru.

Definicni moznosti voleb sekce jsou analogicke k definicnim moznostem celeho parseru. Duvodem je pohodlnejsi pristup pri systematickem definovani voleb ve stejne sekci.

IniOption:

Trida reprezentujici jednu volbu. Kazda volba si drzi informaci o typu svych hodnot a zda jde o seznam, nebo jednohodnotovou volbu. Vnitřně jsou hodnoty ukládány jako řetězce (v instancích třídy `Element`). Důvodem je možnost nahradit reference uvnitř hodnot až ve chvíli přístupu k hodnotě. Jednotlivé instance jsou také zodpovědné za typovou kontrolu při přiřazování hodnot a jejich zpětném přetypování při přístupu k hodnotám.

K dispozici jsou sady metod pro přístup k jednotlivým možným typům hodnot. Existují varianty pro přístup k jednohodnotovým volbám a seznamům hodnot. Přístup správným typem metody se kontroluje. Typ volby lze zjistit metodou `getType`, zda se jedná o seznam lze zjistit pomocí metody `isList`.

Komentare:

Pokud je komentar na stejne radce jako definice nejakeho objektu (sekce nebo volba), pak prislusi k tomuto objektu a je u nej ulozen.

Komentare na samostatných radkách prísluší objektu ktorý nasleduje za nimi. Týchto komentárov môže byť niekoľko riadkov a sú uložené ako `priorComments` u prvého objektu ktorý za nimi nasleduje. Dvomi je skutočnosť, že býva zvykom najprv v komentároch popísať čo voľba znamená a potom prvú voľbu uviesť.

Komentare na konci súboru prísluší samotnému konfiguračnému objektu, sú teda uvedené ako `closingComments` u `IniParseru`.

Možnosť úprav z pohľadu užívateľa:

Rozhodli sme sa implementovať možnosť užívateľského prístupu (okrem primárneho volania prístupových metód) i možnosť prechodu pomocou návrhového vzoru `visitor`. `IniParser`, `IniSection` a `IniOption` majú preto metódu `accept`. Týmto spôsobom

probíha napríklad už zahrnutý výpis konfigurácie (použitím `StringVisitoru`). Výhodou tohto prístupu je, že nie je pre užívateľa napríklad problém sám si napísať nejakú triedu implementujúcu poskytnutý `visitor` interface. Môže tak napríklad kontrolovať ďalšie integritní omešenie, alebo definovať vlastný formát výpisu konfigurácie.

Postup prechodu `visitorom` je nasledujúci - Postupne sa prechádza sekcie, v každej sekcii sa prechádza všetkými voľbami. Nakoniec sa pristúpi k `IniParseru`.

Hlasenie chýb:

Jako spôsob hlásenia chýb sme zvolili systém výjimek. Hlavnú výhodu vidíme vo skutočnosti, že pomocou výjimek dáme užívateľovi nejen vedieť, že nastala chyba, ale aj o aký typ chyby sa jedná. Užívateľ má potom možnosť na chybu reagovať priamo z programu. Napríklad pokiaľ pristúpi k list-voľbe metódou pre prístup k jednohodnotovej voľbe, pomocou výjimky sa dozvie o svojej chybe a môže zvoliť správnu metódu prístupu.

`BadTypeException` - vyhazovaná v prípade, že sa užívateľ snažil získať typovanú hodnotu z voľby špatného typu (napr. `getBoolean` na string hodnotu)

`BadValueException` - vyhazovaná v prípade, že sa užívateľ snažil dosadiť nepripustnú hodnotu. Napríklad do enum voľby hodnotu, ktorá nie je prvkom príslušného výčtového typu.

`IniAccessException` - vyhazovaná v prípade, že sa užívateľ snažil pristúpiť k list-voľbe metódou pre jednohodnotovú voľbu, alebo opačne.

`ParserException` - vyhazovaná v prípade, že nastala chyba pri parsovaní vstupného konfiguračného súboru.

Parsovanie konfiguračného súboru:

Pro parsovanie konfiguračného súboru sme sa rozhodli využiť to, že Java umie hezky pracovať s regulárnymi výrazmi a používame ich k matchovaniu identifikátoru. Je sice pravda, že regulárne výrazy bývajú niekedy obťažné, ale podľa nášho názoru je to presne len elegantnejšie riešenie než zdĺhavé funkcie na prechádzanie po znákoch a testovanie a ošetrovanie všetkých prípadov.

Autori: Vladimír Fiklik, Michal Demin