Dokumentace úlohy XQR: XML Query v Python 3 do IPP 2015/2016
Jméno a příjmení: František Šumšal
Login: xsumsa01

1 Úvod

Cílem tohoto projektu byla implementace skriptu v jazyce Python 3, který zpracovává dotazy specifikované jazykem s gramatikou podobnou SQL. Tento dotaz je poté aplikován na vstupní dokument. Výsledkem je množina XML elementů odpovídající zadanému dotazu.

2 Implementace

Vlastní implementaci obstarávají z největší části dvě hlavní třídy - QueryParser a XMLParser.

2.1 QueryParser

Tato třída má na starost zpracování vstupního dotazu. Konstruktor třídy načítá dotaz ze souboru nebo případně ze zdrojového řetězce. Tento dotaz je poté zpracován metodou process, která na dotaz aplikuje regulární výraz, jehož výsledkem je slovník (dictionary) s jednotlivými prvky dotazu.

```
"^SELECT\s+(?P<select>[^ ]+?)"
"(\s+LIMIT\s+(?P<limit>[0-9]+?))?"
"\s+FROM\s*(?P<from>((?P<elem>[^ .]+?)?"
        "(\.(?P<att>[^ ]+?))?))?"
"(\s+WHERE\s+(?P<not>NOT\s+)?"
"(?P<where>((?P<welem>[^ .]+?)?"
       "(\.(?P<watt>[^ ]+?))?))\s+"
    "(?P<wop>(=|<|>|CONTAINS))\s+"
    "(?P<wliteral>(\"[^\"]*\"|\d+(\.\d+)?)))?"
"(\s+ORDER BY\s+(?P<orderby>((?P<oelem>[^ .]+?)?"
        "(\.(?P<oatt>[^ ]+?))?)(?<!\s))"
"\s+(?P<order>ASC|DESC))?$")
```

2.2 XMLParser

Pro zpracování XML dokumentu slouží třída XMLParser. Načítání XML ze souboru či vstupu probíhá v konstruktoru třídy. Hlavní zpracování dokumentu zařizuje metoda query, která jako argument přijímá data vygenerovaná třídou QueryParser. Pomocí těchto informací je poté z XML dokumentu vybrána jen množina elementů odpovídající danému dotazu. K samotnému zpracovávání slouží také pomocná metoda compare, která porovnává data dvou elementů pomocí jazykem specifikovaných operátorů, privátní metoda _generate sloužící pro vygenerování výsledného XML stromu a funkce cast, která přetypovává danou proměnnou na daný datový typ.

Výslednou množinu elementů je poté možno zapsat do souboru/na standardní výstup pomocí metody output. Samotný výstup se dá ovlivnit nastavením patřičných argumentů, kde se poté k výstupu přidává XML hlavička nebo se výstup obalí kořenovým XML elementem.

Jedním z větších problémů bylo zpracování množiny elementů bez kořenového elementu. Jelikož to porušuje standard, tak tohle chování XML knihovna Pythonu ani neumožňuje. Tento problém jsem obešel vytvořením virtuálního kořenového XML elementu, který před vlastním výpisem dokumentu odstraním.

2.3 Výjimky

Pro jednodušší zpracování chyb a jejich svázání s patřičnými chybovými kódy je ve skriptu vytvořeno několik výjimek. Většina obsluhy těchto výjimek probíhá ve funkci 'main', kde se jednotlivým výjimkám přiřazuje vhodná chybová zpráva a korektní chybový kód.

2.4 Zpracování vstupních argumentů

Zpracování vstupních argumentů zajišťuje knihovna argparse. Pomocí této knihovny se dá jednoduše specifikovat množina validních argumentů, společně s dalším potřebným nastavením, jako jsou defaultní hodnoty, cílové umístění hodnot, konfliktní argumenty a další.

Jediným problémem s touto knihovnou byla změna defaultního chybového kódu při ukončení. Řešením je odchycení výjimky SystemExit a následné vyhodnocení chtěného koncového stavu podle člena třídy code.