

1. Stručný popis činnosti skriptu

Účelem skriptu je vyfiltrovat vstupní XML soubor na základě zadaného dotazu, jenž je podobný příkazu `SELECT` jazyka `SQL`. Vyfiltrovaný obsah je následně zapsán do výstupního XML souboru. Pro tyto činnosti skript využívá funkce pro zpracování regulárních výrazů a nástroje pro přístup k XML datům.

2. Popis jednotlivých částí skriptu

Skript je rozdělen na několik částí, které jsou blíže popsány níže.

2.1 Zpracování parametrů

Zpracování parametrů je realizováno pomocí funkcí pro práci s regulárními výrazy. Nejdříve jsou všechny parametry příkazové řádky načteny do jedné hlavní proměnné pomocí funkce `join`. Následně jsou všechny parametry, které má skript znát jednotlivě načteny do pomocné proměnné pomocí funkce `re.compile`, přičemž po načtení každého parametru probíhá pomocí funkce `search` test, zda se daný parametr nachází v hlavní proměnné. Součástí zpracování parametrů je i testování, zda některý z parametrů nebyl zadán vícekrát nebo zda nebyl zadán parametr, který skript nezná.

2.2 Zpracování dotazu

Zpracování dotazu probíhá pomocí funkcí pro práci s regulárními výrazy a funkcí pro práci s datovým typem `string`. Nejdříve jsou z celého dotazu odstraněny přebytečné bílé znaky pomocí funkce `strip`. Následně jsou jednotlivá slova z dotazu vložena do pole pomocí funkce `split`.

Zpracování všech slov v poli probíhá v cykle, který zkoumá syntaktickou a sémantickou správnost daných slov pomocí konečného automatu na základě pravidel bezkontextové gramatiky. V případě že byla porušena syntaktická nebo sémantická správnost některého slova, pak skript skončí s příslušným chybovým kódem. Na základě přítomných slov v poli jsou v cykle taktéž nastaveny různé pomocné přepínače, které jsou dále využity při analýze vstupního XML souboru.

Pokud je zadaný dotaz uložen v souboru, pak jeho načtení probíhá pomocí funkcí `open` a `read`.

2.3 Analýza vstupního XML souboru a vytváření výstupního XML souboru

Analýza vstupního a vytváření výstupního XML souboru využívá knižnici `xml.dom.minidom`. Nejdříve je celý vstupní XML soubor načten pomocí funkce `parse`. Následně proběhne redukce vstupního XML souboru na základě aktivních `from` přepínačů:

Pokud je aktivní přepínač `from_element`, pak redukce proběhne pouze pomocí funkce `getElementsByTagName`.

Pokud je aktivní přepínač `from_element.attribute`, pak jsou nejdříve načteny všechny dané elementy pomocí funkce `getElementsByTagName`. Následně proběhne v cyklu vyhledávání daného atributu v načtených elementech pomocí funkce `hasAttribute`. Cyklus skončí, jakmile je nalezen první element, který obsahuje daný atribut.

Pokud je aktivní přepínač `from_.attribute`, pak jsou nejdříve načteny všechny elementy daného XML dokumentu. Následně proběhne v cyklu vyhledávání daného atributu stejným způsobem, jako u přepínače `from_element.attribute`.

Zredukovaný vstupní XML soubor je pak dále redukován na základě aktivních `select` přepínačů. Tato redukce probíhá stejným způsobem jako redukce na základě aktivních `from` přepínačů.

Po redukci na základě aktivních `select` přepínačů následuje vytváření výstupního XML souboru z redukovaného XML souboru na základě aktivních `where` přepínačů a pomocných proměnných. V případě, že není aktivní přepínač `-n`, je nejdřív do výsledného XML souboru zapsána XML hlavička pomocí funkce `write`. Zbytek výstupního XML souboru se vytváří v cykle:

Na začátku každého průchodu cyklem je nastavená pomocná proměnná `zapis` na hodnotu `true`. Následně probíhá analýza jednotlivých elementů z redukovaného XML souboru. Pokud daný element nesplňuje podmínky `where` přepínačů, je pomocná proměnná `zapis` nastavená na hodnotu `false`. Po analýze daného elementu proběhne kontrola hodnoty pomocné proměnné `zapis`, na základě které se daný element zapíše nebo nezapíše do výstupního XML souboru.

