Dokumentace úlohy SYN: Zvýraznění syntaxe v Python 3 do IPP 2016/2017
Jméno a příjmení: Jan Grossmann
Login: xgross09

Úvod:

Můj způsob řešení úlohy SYN: zvýraznění syntaxe se skládá z několika částí. Jádro celého programu je funkce `execute()`, která spouští další funkce ve správném pořadí, a do jisté míry zajišťuje i přehlednost zdrojového kódu. Důležité hodnoty pro správnou činnost programu se ukládají do slovníku `files`, formátovacího pole `format` a do proměnné `br` se uloží `Boolean` hodnota True, jestli byl zadán argument -b. Do slovníku `files` se ukládají jména vstupního, vystupního i formátovacího souboru, načtené z argumentů použitých při spuštění programu. Do pole format uložím formátovací pravidla načtená z formátovacího souboru.

Zpracování vstupních argumentů:

Prvně je potřeba zpracovat vstupní argumenty. Funkce `basicArgsCheck()` zjistí, zda zadané argumenty splňují požadavky validity, tedy jestli mají všechny argumenty, které předávají hodnotu hodnotu a také jestli nejsou některé argumenty zadány vícekrát. V případě, že argumenty tyto podmínky nesplňují se volá funkce `throwErr()` s chybovým hlášením. Další funkce zabývající se argumenty je funkce `parseArgs()`, ta se stará o naplnění slovníku `format`, flagu `br` a vypsání nápovědy. Mé řešení úlohy SYN podporuje zkrácené argumenty, konkrétně `-i` místo `--input`, `-o` místo `--output`, `-f` místo `--format`, `-b` místo `--br` a `-h` místo `--help`. Pokud je zadaný pouze argument `--help`, program vypíše nápovědu a ukončí se.

Ověření I/O souborů:

Jednoduchá funkce, která přichází na řadu po kontrole validity argumentů, má na starosti ověřit, jestli existuje vstupní a výstupní soubor. Dále také, jestli je výstupní soubor `writeable`, tedy že není pouze ke čtení, nebo jestli existuje složka, ve které se má output soubor nacházet. V případě, že tomu tak není, se volá funkce `throwErr()`.

Načtení a zpracování formátovacího souboru:

Funkce si nejdříve rozparsuje vstup po jednotlivých řádcích, nerozlišuje zda se jedná o `STDIN` nebo textový soubor. Vynechává prázdné řádky. Zbytek funkce je jeden velký cyklus, který prochází načtený vstup řádek po řádku a ověří, jestli formát formátovacího souboru odpovídá požadavkům na formátovací soubor. Pokud ano, rozdělí řádek na 2 části. Na zapsané regulární výrazy IPP a na formátovací tagy. Regulární výrazy IPP je potřeba převést na regulární výrazy, kterým rozumí Python. Toho dosáhnu pomocí série `x.replace()` volání, vhodně umístěných za sebe. Pokud je potřeba zachovat strukturu regulárního výrazu, tedy pokud se nejedná o obyčejnou záměnu znaku za znak, použije se funkce `re.sub()`. Důležité pro validitu regulárního výrazu je spočítat počet pravých i levých závorek, případně vyhodit chybu u kombinací znaků, které netvoří žádný regulární výraz. Formátovací tagy se převedou na vhodný tvar a spolu s uzavíracím tagem se přidají do pole, ze kterého se budou později vytahovat. Je ale zapotřebí zachovat pořadí, v jakém byly tagy zapsány do formátovacího souboru. Výsledný regulární výraz a pole tagů se uloží do hlavního pole `format`.

Aplikace formátování:

Otevře se vstupní a výstupní soubor, pokud není vstupní soubor zadán, považuje se za vstupní soubor `STDIN`. Ten se přečte řádek po řádku. Z formátovacího pole se načte regulární výraz a vytvoří se z něj `pattern`. Poté se prochází celý vstupní soubor a pozice v textu, které odpovídají regexu se poznamenávají do slovníku `matches`. Klíč představuje pozici v textu, na místo hodnoty se přidají otevírací nebo zavírací tagy, které jsou spojené s vyhledávaným regexem. Pokud již pozice v textu tag obsahuje, další otvírací tag se přidá za něj, ale zavírací tag se přidá na začátek fronty tagů. Po průchodu všemi prvky z pole `format` se spustí finální cyklus,

který na číselně označené pozice přidá frontu tagů. V případě argumentu `--br` se před konce řádků přidají tagy `<br >`. Výsledný text se buď zapíše do output souboru nebo vypíše na `STDOUT`.

Rozšíření NQS

Rozšíření NQS je realizováno pomocí 4 cyklů `while`, které se orientují podle výsledku `re.search()`. Pokud se ve formátovacím textu nachází jedna ze 4 kombinací znaků, je nahrazena buď + nebo `*`.