Dokumentace úlohy CST: C Stats v Python 3 pro předmět IPP 2014/2015
Jméno a příjmení: Tomáš Coufal
Login: xcoufa09

Úvod

Program zjišťuje statistiky zdrojových souborů jazyka C a to: počty klíčových slov, identifikátorů, operátorů, znaků komentářů či hledá uživatelem zadaný řetězec. Pracuje jak s jedním souborem tak s mnoha.

Zpracování parametrů

Pro zpracování parametrů program používá modul `argparse`. Ten definuje třídu `ArgumentParser`, která bohužel nevyhovuje zadání plně, a proto program používá její modifikaci, odvozenou třídu `MyParser`. Ta vynucuje použití znaku rovná se u parametrů, které jej vyžadují, zabraňuje nepřesnostem a používání zkrácených názvů (tato možnost, jinak implementovaná se v Pythonu objeví až od verze 3.5) . Bohužel se tento zásah neobešel bez nečistého a neestetického zásahu do privátní metody `ArgumentParser`u, metody `._get_option_tuples()`. Pro druhou úpravu byl použit kód podle rady zde: http://stackoverflow.com/questions/10750802/disable-abbreviation a kód je až na ony úpravy opsán z modulu `argparse`.

Návratové kódy

Pro ošetření návratových kódů při otevírání souborů byl vytvořen nový `@contextmanager open_w_exit()`, který je použit v rámci `with` statementu a volá funkci `open()` s tím rozdílem, že pokud byla vyvolána výjimka `OSError`, zachytí ji a pomocí modulu `traceback` vypíše hlášení a následně skončí se správným návratovým kódem.

Hledání souborů

Z parametru `--input=FILEORDIR` je třeba vyčíst, zdali se jedná o složku či soubor. K tomuto účelu se velmi hodí standardní modul pro Python, modul `os`. Soubory prohledává funkce `list_files()`. Ta, pokud se jedná o složku program použije `_list_dir()`, která z ní vybere pouze soubory s příponou `.c` a `.h`, jinak použije zadaný soubor bez ohledu na příponu. Pokud není uživatelem specifikováno jinak a právě prohledávaná složka obsahuje podsložky, pak na ně funkce rekurzivně volá sama sebe. Nalezené soubory se střádají do listu, který se předává k dalšímu zpracování. U všech souborů také modifikuje cesty a vynucuje absolutní způsob zápisu.

Zpracování souboru

Nad každým ze zpracovávaných souborů je spuštěna funkce find_in_file(), která soubor transformuje na instanci třídy SourceCode. Tato třída načte obsah souboru a uloží si jej. Třída dále nabízí metody jako `.comments_strip()`, `.identificator_match()`, `.keyword_strip()`, `.macro_strip()`, `.operator_strip()`, `.pattern_match()` a `.string_strip()`, které následně načtený text souboru upravují, počítají výskyty a odstraňují již započítané. Samotné zpracování probíhá následovně:
Pokud chce uživatel vyhledat konkrétní řetězec (parametr `-w`) je zavolána metoda `.pattern_match()`. Ta prohledá obsah souboru pomocí regulárního výrazu tvořeného hledaným řetězcem. Pak je vrácen počet výskytů.
Jinak jsou metodou `.macro_strip()` odstraněna makra a zavolána metoda `.comments_strip()`. Ta spočítá znaky komentářů a odstraní je z textu. Pokud je tedy zadám přepínač -c není důvod k dalším výpočtům a je vrácen výsledek. V opačném případě jsou metodou `.string_strip()`, odstraněny řetězcové a znakové literály. Poté jsou spočítány a odstraněny klíčová slova metodou `.keyword_strip()`, následně operátory `.operator_strip()` a nakonec jsou spočteny zbylé řetězce jakožto identifikátory metodou `.identificator_match()`.
Vrácením počtu výskytů je myšlen návrat slovníku ve tvaru `{nazev_souboru: pocet_vyskytu}`. Tímto slovníkem je pak aktualizován hlavní slovník, kde jsou kumulovány výsledky.

Výpis

Ze zadání je možné dovodit, že první akce při výpisu je úprava názvů souborů (jestli vypisovat i cestu či nikoliv). Při odstraňování cesty však může docházet ke konfliktům (název souboru je klíč ve slovníku a více souborů může mít stejný název). Proto se ke každému názvu souboru přidá mezera (nemůže být v názvu souboru) a číslo. Při výpisu se pak použije jen první část. Následně slovník seřadí abecedně podle klíčů a uloží do instance třídy `OrderedDict` z modulu `collections`.
Python poskytuje možnost formátování textu pomocí metody `.format()`, kde lze definovat zarovnání, vyhrazenou délku pro prvek a další. Program tohoto využívá a zjišťuje nejdelší název a největší (nejdelší co do zápisu) hodnotu, do svých výpočtů zahrne i poslední řádek se sumou. Mezi ně pak přidá mezeru. Následně prochází slovníkem a zapisuje takto naformátovaný řádek do výpisového bufferu. Nakonec je přidán řádek s konečným součtem. Podle toho, zda-li uživatel definoval parametr `--output=` je rozhodnuto o vypsání bufferu do souboru či na `stdout`.