Dokumentace úlohy CST: C Stats v Pythonu 3 do IPP 2014/2015

Jméno a příjmení: Jan Pawlus

Login: xpawlu00

Struktura projektu

Celý projekt je rozdělen do dvou tříd a spouštěcího souboru, který vytváří objekty. Třída `ArgParser` zpracovává argumenty a ošetřuje některé nepřípustné situace. Jelikož Python neumožňuje nastavit modifikátory přístupu u proměnných třídy a všechny proměnné jsou tedy okolí přístupné, může spouštěcí soubor tyto proměnné přímo přečíst a poslat je jako parametry do další, vyhledávací, třídy `FileExamination`.

Třída ArgParser

V této třídě se ukládají informace o argumentech. Přístup k nim je řešený funkcí `getopt`, následně se v cyklu for prochází jednotlivé volby. V této třídě jsou ošetřeny některé nepřístupné kombinace argumentů (`--help` s čímkoliv jiným, více voleb vyhledávání, …) a je zde také metoda pro výpis nápovědy.

Třída FileExamination

Tato třída obsahuje v podstatě většinu řešení tohoto projektu. Její konstruktor přijímá jako parametry informace o argumentech skriptu a na základě těchto informací zkompiluje do určitých proměnných třídy potřebné regulární výrazy. Poté je zavolána metoda `openFile`, která podle zadaných argumentů buď projde všechny soubory (a podsložky) v aktuální složce pomocí `os.walk`, nebo pouze specifický cíl. Při otevření každého souboru si skript ukládá informaci o nejdelší cestě k souboru, což je klíčové při výpisu výsledků.

Jednotlivé metody vyhledávání

Jak je již zmíněno výše, při konstrukci objektu třídy `FileExamination` se rovnou zkompilují jednotlivé regulární výrazy. Jako první poté metoda `searchString` zpracuje zdrojový text tak, aby se `\n` a `\r\n` započítaly jako jeden znak, odstraní se textové řetězce a nahradí se problematické escape znaky. Pro vyhledávání komentářů jsou to pouze regulární výrazy pro nalezení maker, jednořádkových a víceřádkových komentářů. Pro vyhledávání klíčových slov nejsou kromě odstranění komentářů (a maker) a samotného nalezení klíčových slov zapotřebí žádné další zásahy do textu. Naopak při hledání identifikátorů je třeba navíc odstranit všechna klíčová slova, při hledání operátorů je třeba rozlišit operátor a `*` při deklaraci proměnné a samotné vyhledávání operátorů je rozděleno do více fází. První je nutné najít složitější operátory (například `++`) a započítat je jako jeden výskyt, následně je odstranit a až poté najít jednodušší operátory. Při vyhledávání určitého prvku se jen neprovedou regulární výrazy pro odstranění komentářů a maker.

Zpracování výsledků

Vyhledávací metoda `searchString` se za běhu skriptu volá dvakrát, pokaždé s jiným parametrem. Poprvé se pouze počítá celkový počet výskytů a nejdelší cesta k souboru, a to proto, aby v druhém průběhu mohl skript vytvořit správný formát výsledku. Správný formát je dosažen v druhém průběhu přidáváním mezer ve for cyklu, který proběhne tolikrát, jaký je rozdíl v délce cesty aktuálního souboru a v délce nejdelší cesty. Další cyklus pak dělá to samé, jen proběhne tolikrát, jako je rozdíl délky aktuálního počtu výskytů a celkového počtu výskytů. Tímto je dosaženo zarovnání posledního sloupce doprava.

Jako poslední je třeba seřadit výsledky podle abecedy. Místo výpisu výsledku v druhém průchodu jsou jednotlivé výsledky ukládány do seznamu, který nakonec metoda `printSum` seřadí a vypíše podle argumentu buď na standardní vstup, nebo do souboru. Tímto je tak vyřešen problém seřazení výsledků na základě toho, zda si uživatel vyžádá nebo nevyžádá cestu k souboru (pořadí výpisu v těchto dvou situacích může být rozdílné).