Dokumentace úlohy CST: C Stats v python do IPP 2012/2013
Jméno a příjmení: Tomáš Kubovčík
Login: xkubov02

Dokumentácia k projektu č.2 do predmetu IPP, CST v jazyku Python

Analýza zadania:

Mojou úlohou v tomto projekte bolo spracovať zdrojové súbory v jazyku C a vytvoriť štatistiky údajov ako počet znakov v komentároch, počet kľúčových slov, identifikátorov, jednoduchých operátorov a užívateľom zvoleného reťazca. Vstupný súbor môže byť zadaný konkrétne, prípadne môže byť vstupom adresár v ktorom sa zdrojové súbory môžu vyhľadávať rekurzívnym zanorením. Výstupné dáta majú pevne daný formát, ktorým je jeden riadok pre jeden zdrojový súbor, ktorý obsahuje názov súboru, prípadne absolútnu cestu k tomuto súboru a počet nájdených elementov.

Postup riešenia:

Prvým krokom po spustení skriptu je spracovanie parametrov z príkazového riadku. Na ich spracovanie som nevyužil knihovné funkcie, ktoré ich spracovanie umožňujú, ale spracovával som manuálne metódou `handle()` z triedy `handleParams` a to takým spôsobom, že pri detekcii správneho formátu prepínača som zvyšoval hodnoty pomocných premenných. Po zistení týchto hodnôt je potrebné overiť prípadné nevyhovujúce kombinácie prepínačov (`--nosubdir` zadaný súčasne s prepínačom `--input`, ktorý sa odkazuje na jeden konkrétny súbor, `--help` nie je zadaný samostatne, v mojom prípade taktiež pri implementácii rozšíreniaCOM kedy bolo potrebné implementovať prepínač `-s` , ktorý sa viaže iba na prepínač `-c`) a skutočnosť, či nebol niektorý z prepínačov zadaný viackrát.

Po spracovaní parametrov dochádza k samotnému prehľadávaniu zdrojových súborov. Jednotlivé súbory som spracovával po riadkoch v ktorých som s využitím triednych metód triedy `RemoveNotNeeded` eliminoval zvolené elementy a predišiel tým vyhľadávaniu v konštrukciách, v ktorých to nie je potrebné.

Po úspešnom vyhľadaní požadovaných elementov v zdrojovom súbore sa celý proces opakuje za predpokladu, že užívateľ zadal ako vstupný parameter priečinok, ktorý obsahuje viac valídnych zdrojových súborov alebo v prípade, že nebol vstupný parameter definovaný kedy dochádza k implicitnému prehľadávaniu aktuálneho pracovného adresáru a jeho podadresárov.

Vyhľadané dáta sa ukladajú do štruktúrovaného údajového typu vo forme [(*názov/cesta, počet*), (*názov/cesta, počet*), ...] kde *počet* je výsledný počet nájdených požadovaných elementov a *názov* je názov súboru v ktorom sa vyhľadávalo a *cesta* je absolútna cesta k nemu. Tieto dáta sa po ukončení vyhľadávania zoradia podľa ASCII hodnôt znakov s využitím vstavanej funkcie `sorted()`, kde pri radení použijeme ako kľúč *názov* alebo *cestu* k súboru.

V poslednej časti tohto programu sa nájdené a zotriedené dáta vypíšu do zadaného výstupného súboru v kódovaní ISO-8859-2 prostredníctvom funkcie `printStats()`, prípadne na štandardný textový výstup ak nebol výstupný súbor zadaný.

Vyhľadávanie v zdrojových súboroch

Jedná sa o nosnú časť tohto projektu. Ako som už naznačil, vyhľadávaniu v súboroch predchádzala eliminácia nepotrebných elementov. Túto elimináciu zabezpečujú metódy `gridElements()`, ktorá zmaže makrá, definície, atď., metóda c`omments()` vymaže komentáre, `strings()` zase reťazcové literály a nakoniec `chars()` znakové literály. Tieto metódy sú súčasťou vyššie spomínanej triedy `RemoveNotNeeded`. Týmto postupom sa predchádza niektorým kolíziám ako napríklad vyhľadávanie kľúčových slov v reťazcových alebo znakových literáloch, kedy by dochádzalo k nesprávnym výsledkom. Takýchto prípadov je však v tomto projekte potrebné ošetriť niekoľko pre každý prepínač. Napriek tomu, v niektorých prípadoch nebola táto eliminácia nutná. Konkrétne sa jedná o rozšírenie COM, ktoré v mojej implementácii pozostávalo zo zakázania eliminácie komentárov v makrách pri zadaní prepínačov `-s -c`.

Po eliminovaní nepotrebných prvkov a ošetrení možných kolízií sa vyhľadávajú požadované prvky. Samotné vyhľadávanie v jednom zdrojovom súbore sa realizuje prostredníctvom metódy `handleFile()` z triedy `find`, výber zdrojových súborov jazyka C realizuje metóda `FindElements()` taktiež z tejto triedy.

Vyhľadávanie sa realizuje použitím regulárnych výrazov, ktoré vyhľadávanie značne uľahčujú. Pri vyhľadávaní identifikátorov sa jedná napríklad o regulárny výraz `[_A-Za-z][_A-Za-z0-9]*`, pri hľadaní

kľúčových slov (*\Wkeyword\W|^keyword\W|^keyword$|\Wkeyword$)* kde *keyword* je kľúčové slovo jazyka C uložené v poli reťazcov. Na vyhľadávanie operátorov sa taktiež používajú regulárne výrazy, ktoré pre ich počet a možnú nejednoznačnosť budem uvádzať. Pre záujemcov sú dostupné v zdrojovom súbore projektu *cst.py*.

Záver:

Po spracovaní tohto projektu musím usúdiť, že sa jedná o jeden z jednoduchších projektov na vytvorenie, ale o to ťažší na odladenie. Veľkou nevýhodou je, že k projektu nebol zadaný referenčný nástroj ktorým si môžeme výsledky overiť. Preto aj testovanie na väčších zdrojových súboroch nemá zmysel, ak si človek nechce krátiť dlhé chvíle manuálnym počítaním rádovo stoviek, prípadne tisícov identifikátorov alebo operátorov.

Projekt som sa rozhodol vyvíjať priamo na referenčnom serveri merlin s operačným systémom CentOS, aby som sa vopred vyhol prípadným medziplatformovým problémom a nekompatibilitám. Na testovanie som využil verejné testy k tomuto projektu zverejnené na stránkach predmetu IPP a testovacie skripty, ktoré som si vytvoril pre vlastnú potrebu.