

Dokumentácia k projektu

IPP- Interpret XML reprezentace kódu

Login: xfeile00

Meno a priezvisko: Marcel Feiler

Návrh

Riešenie druhej úlohy týkajúcej sa skriptu s názvom *interpret.py*, ktorá má za cieľ korektne interpretovať zdrojový jazyk IPPcode23, kde zdrojový kód tohto jazyka je reprezentovaný určitou XML štruktúrou, bol navrhnutý objektovo. Samotný interpret je rozdelený na niekoľko častí, z ktorých pomocné skripty sa nachádzajú v priečinku */interpreter_lib* nachádzajúci sa v root adresári. Tieto skripty ďalej používajú ďalšie skripty v tomto priečinku, čo nám umožnilo vytvoriť základnú štruktúru návrhu. Ako prvé sa teda spracovávajú užívateľom zadané argumenty, ktoré sa následne kontrolujú ako aj XML štruktúra. Po korektnej kontrole program prejde do stavu *start()*, kde sa začínajú konať jednotlivé inštrukcie.

Štruktúra adresárov

Na spustenie zdrojového programu je používaný *interpret.py*. Ten sa nachádza v koreňovom adresári spolu priečinkom */interpreter_lib*. Tento priečinok obsahuje skripty *core.py*, *components.py*, *frames.py*, *instructions.py*, *instruction_list.py* a *exit_codes.py*. V štruktúre uvedenej nižšie môžete vidieť konkrétnu adresárovú kostru

- interpret.py

- README.pdf

- interpreter_lib

 - components.py

 - core.py

 - exit_codes.py

 - frames.py

 - instruction_list.py

 - instructions.py

Spracovanie vstupných parametrov

Vstupné parametre sa spracovávali vo funkcii *__argument_parse()* definovaná v skripte *core.py*. Pri vstupných parametroch sa kontroluje prítomnosť *-input* a *-source* a existencia k nim priradených súborov.

Spracovanie, validácia XML zdrojového kódu

Validácia XML sa koná vyvolaním funkcií zo skriptu *components.py*, a to predovšetkým *check_integrity_of_tree()*, ktorá kontroluje integritu a správnu hlavičku. Funkcia *check_each_instruction()*, ktorá kontroluje správny tvar inštrukcií a následne ako posledná funkcia *instruction_importing()*, v rámci ktorej si potrebné inštrukcie importujeme a teda s nimi môžeme v ďalšom behu programu pracovať.

Inštrukcie a ich spracovanie

Po korektnom overení XML formátu pokračujeme na overenie správneho tvaru v liste inštrukcií, ktoré vyvoláme pomocou funkcie *string_checking()* definovanej v *instruction_list.py*. Ako pomocná funkcia jej na to slúži *string_fixation()*. Získanie dát a typu argumentov danej inštrukcie získame pomocou funkcie *get_data_arg_type()*, ktorú využíva každá jedna inštrukcia v hlavnom chode *core.py* kvôli umožneniu prístupu k jej hodnotám. Ako špeciálna nám slúži funkcia *get_type()* pre korektné zaobchádzanie s inštrukciou TYPE.

Rámce

Skript *frames.py* nám slúži ako pre inicializáciu, tak aj pre funkcie vytvárajúce alebo upravujúce tieto rámce. Využívajú ich funkcie v skripte *instructions.py* a *core.py*, kde pomáhajú chodu hlavného programu.