Implementačná dokumentácia k 2. úlohe do IPP 2021/2022
Meno a priezvisko: Peter Ďurica
Login: xduric05

1 Návrh riešenia – interpret.py

Interpret po zistení pozície zdrojového xml kódu a vstupných hodnôt začne pomocou knižnice xml prekladať zdrojový kód. Keď sú elementy stromu uložené v premennej s ktorou vie interpret pracovať začína kontrola správneho zapísania názvov a atribútov jednotlivých elementov v strome. Ak sú všetky elementy správne tak ich interpret zoradí podľa poradia inštrukcií a začne preklad do vlastných objektov stromu. Keď je zdrojový kód uložený v objektoch, vytvorí interpret rámce pre pamäť programu a pomocné premenné pre zabezpečenie správneho chodu programu. Cez zdrojový kód interpret prejde dvakrát. Raz nájde všetky návestia a uloží si ich polohu v kóde. Ostatné inštrukcie sa vykonávajú počas druhého behu programu. Tým zabezpečíme skoky na ľubovoľné miesto v kóde.

2 Triedy – interpret.py

2.1 Triedy na uloženie programu

Na uloženie programu sa využívajú triedy Program, Instruction a Argument. Objekt triedy Program sa skladá z poľa inštrukcií. Objekt triedy Instruction sa skladá z operačného kódu inštrukcie, jej poradia v kóde a poľa argumentov. Objekt triedy Argument je zložená z typu argumentu a jeho hodnote.

2.2 Triedy na správu pamäte

Trieda na uloženie premennej sa volá Variable. Obsahuje meno, typ a hodnotu premennej. Objekt takejto premennej môžeme uložiť do troch rôznych typov rámcov a to GF, LF a TF. GF je uložený ako klasické pole ale LF je uložený ako objekt triedy Local_Frame a obsahuje pole rámcov do ktorého môže interpret vkladať a vyberať objekty triedy Temporary_Frame a ukladať ich do premennej TF. Objekt triedy Temporary_Frame obsahuje pole premenných a boolean hodnotu svojej platnosti.

3 Pomocné funkcie – interpret.py

3.1 Práca s premennými

Na prácu s premennými využíva interpret funkcie hlavne funkcie `Get_Var()`, `Put2Var()` a `MakeVar()`. Funkcia `Get_Var()` dokáže podľa názvu premennej ju vyhľadať v ľubovoľnom rámci a vrátiť jej hodnoty. Ak takáto premenná nie je nájdená tak sa interpret ukončí na príslušnej chybe. Funkcia `Put2Var()` dokáže poslať hodnotu z premennej do druhej premennej alebo ak cieľová premenná neexistuje tak je ukončený program chybou. Funkcia `MakeVar()` dokáže podľa typu argumentu inštrukcie zistiť či sa jedná o premennú alebo nie a podľa toho vrátiť správny objekt premennej s doplnenými hodnotami pre následné operácie.

3.2 Aritmetické, relačné a booleovské operácie

Na tieto operácie interpret využíva funkcie `QuickMafs()`, `QuickRel()`, `QuickBool()` a `QuickNot()`. Tieto funkcie dokážu vykonať podľa zadaného operátora a argumentov danú operáciu, prípadne identifikovať chybné operandy a ukončiť program na príslušnej chybovom kóde. Po vykonaní operácie príslušnú hodnotu vložia do cieľovej premennej

4 Funkcia interpret – interpret.py

Funkcia interpret podľa operačného kódu vloženej inštrukcie zistí ktorá inštrukcia sa má vykonať na zadaných argumentoch. Táto funkcia je volaná na každú inštrukciu v programe. Ak operačný kód kontrolovanej inštrukcie nie je medzi podporovanými možnosťami je program zhodený s príslušnou chybou. Pri skokoch v programe sa mení hodnota globálnej premennej, ktorá znázorňuje pozíciu v programe. Táto hodnota sa po každej vykonanej inštrukcii inkrementuje pokiaľ nie je väčšia ako maximálne poradie inštrukcie v programe. Ak toto nastane tak interpret vie že prešiel celý program a môže ukončiť svoju činnosť.