Implementační dokumentace k 2. úloze do IPP 2020/2021

Jméno a příjmení: Patrik Slovák

Login: xslova21

1 Popis zdrojového kódu interpret.py

1.1 Argumenty programu

V projekte je implementovaný volitelný argument --help. Pri jeho použití sa vypíše na štandardný výstup návod na používanie programu. Ďalej existuje argument --source, ktorý načíta súbor so zdrojový kódom vo formáte XML, a argument –input, ktorý načíta vstupný súbor. Ak jeden z argumentov source a input chýba, je nahradený štandardným vstupom STDIN.

1.2 XML analýza

Hneď po kontrole argumentov nasleduje analýza zdrojového XML kódu. V projekte je na analýzu použitá python knižnica xml.etree, ktorá zoberie XML reprezentáciu a preloží ju do stromu objektov. Z tohto stromu sú následne vytiahnuté všetky dôležité informácie, ktoré sa uložia do tried Instruction a Argument. Popri celom procese sa zároveň kontroluje správnosť daného XML kódu.

1.3 Rámce a premenné

Pre správnu funkčnosť premenných je potrebné používať rámce. Rámce sú v projekte implementované v triedach Framestack a Frame. Na začiatku interpretácie sa vytvorý nový objekt triedy Framestack, ktorý v sebe bude uchovávať všetky inicializované rámce. Ďalej sa vytvorí nultý rámec s typom „null“, ktorý sa vloží na začiatok zásobníka rámcov. Aktuálny rámec je uložený v premennej frame, ktorá sa inicializuje ako globálny rámec. Pri inštrukcii PUSHFRAME sa rámec uloží na vrchol zásobníka a jeho typ sa zmení na „LF“, s výnimkou globálneho rámca, ktorý si svoj typ zachová. Pri inštrukcii POPFRAME sa vyjme rámec z vrcholu zásobníka, uloží sa do premennej frame a jeho typ sa zmení na „TF“. V triedach rámcov sú ukladané aj všetky interpretované premenné. Pri každej deklarácii premennej inštrukciou DEFVAR sa v aktuálnom rámci vytvorí objekt triedy Variable, cez ktorú sa bude do premennej neskôr pristupovať s pomocou funkcie updatevar().

1.4 Náveštia

Náveštia sú v projekte implementované v triedach LabelList a Label. Na začiatku intepretácie sa vytovrí objekt triedy LabelList, ktorý v sebe bude uchovávať všetky náveštia. Pri inicializácii triedy LabelList prebehne funkcia load_labels(), ktorá načíta všetky náveštia a uloží ich vo forme objektov triedy Label a skontroluje či viacero náveští nemá rovnaký názov. Ku náveštiam sa neskôr pristupuje cez funkciu get_label(), ktorá vráti objekt triedy Label s požadovaným názvom.

1.5 Zásobník

Zásobník je v projekte implenetovaný ako trieda Stack. Do objektu triedy Stack sa pri inštrukcii PUSHS vloží na vrchol zásobníku nový objekt triedy Symbol, ktorý v sebe uchováva hodnotu a typ daného symbolu. Pri inštrukcii POPS sa symbol z vrcholu zásobníku vyjme.

1.6 Interpretácia

Interpretácia prebieha vo funkcii interpret(), kde sa prejde zoradený list inštrukcií cyklusom for a každá inštrukcia sa za pomocou implementovaných tried a funckií interpretuje. Ku inštrukciám sa pristupuje cez index, ktorý sa každou iteráciou inkrementuje. Pri skákacích inštrukciách sa zmení hodnota tohto indexu na opcode daného náveštia. Chyby pri interpretácii sú rozdelené podľa návratových hodnôt a pri hociakej chybe sa program ukončí s danou hodnotou.

2 Popis zdrojového kódu test.php

1.1 Argumenty

Tak ako pri interprete, existuje v projekte argument --help, ktorý vypíše na štandardný výstup návod na používanie programu. Na začiatku programu sa vytvorí list longParams, do ktorého sa uložia všetky požadované argumenty programu. Tento list je následne použitý vo vstavanej funkcii getopt(). Následne sa výskyt argumentov skontroluje a nastavia sa premenné.