Implementační dokumentace ke 2. projektu do IPP 2018/2019

Jméno a příjmení: David Valecký

Login: xvalec01

Skript interpret.py:

Jedná se o skript interpret.py, který vykoná lexikální analýzu, syntaktickou analýzu, sémantickou analýzu a pak případnou interpretaci vstupního XML (XML reprezentuje IPPcode19). Skript je zpracovaný v jazyce Python, konkrétně ve verzi 3.6.

Použití skriptu:

Skript se soustředí zajistit pro uživatele funkčnost a pohodlí při jeho obsluze. Skript lze spustit těmito způsoby: „`python3.6 interpret.py --source=vstupni_soubor --input=vstupni_soubor`“ nebo také s nápovědou „`python3.6 interpret.py --help`“. Pokud je skript spuštěn, s přepínači `--source` a `--input`, uživatel si může vybrat z jakého souboru budou brána data na `stdin` a z kterých se bude čerpat vstupní kód jazyka.
V případě, že jeden z přepínačů vynecháme je vstup do skriptu načítán ze standardního vstupu), nikdy nesmí chybět oba argumenty. Druhý přepínač je klasický `–help` a nabídne uživateli nápovědu k používání. Jiné spuštění, než uvedené skript ukončí s chybou 10.

Implementace:

Skript využívá knihovny `xml.etree.ElementTree` pro načtení a parsování přiloženého xml souboru v jazyce IPPcode19 a načítá vstupní informace. Tyto vstupní data jsou v cyklech zpracovávány lexikální a syntaktickou kontrolou. Důležité je zmínit tag atributu `program`, který se pečlivě kontroluje. Další kontrola atributů tohoto tagu, je kontrola, jestli jednotlivé instrukce mají atributy `opcode a order`. Kontrola, jestli je zadaná instrukce platná, je součástí lexikální kontroly. Jednotlivé argumenty instrukce, kontrola pomocí regulárních výrazů…

Po těchto kontrolách se provádí vkládání instrukcí do slovníků podle `opcode` (pokud instrukce mají stejné pořadové číslo instrukce, skript končí chybou 32) a na pořadí nezáleží, pomocí funkce `sorted`, lze všechny data seřadit podle klíče, už zmíněného, `opcode`. Po vložení instrukcí probíhá vkládání jednotlivých argumentů instrukcí – od 1 do 3 (skript s větší hodnotou je ukončen s chybou 32) a platí zde to stejné jako u instrukcí, všechny data jsou uloženy ve slovnících.

Jakmile je vkládání dokončeno, lexikální a syntaktická kontrola je hotová. Potřebná data (`opcode`, `order`, `type`, a `text` argumentů) pro sémantickou konrolu jsou zformátována a nejvíc vnořená data (typ a hodnota proměnných) jsou v poli v jako klíče nadřazených slovníků. Slovníky zde slouží hlavně jako prostředníky pro snadný přístup k instrukcím a žádoucím atributům, protože nemají pořadí a dá se lehce nakládat s daty žádoucími způsoby v tomto projektu.

Před interpretací `interpret.py` hledá skript ve `for cyklu` v kódu všechny `label` a kontroluje, jestli se uživatel nesnaží o redefinici návěští a uchová si všechny návěstí ve slovníku `labels`

Internet pracuje s proměnnými `localFream, temporaryFrame, globalFrame, stackFrame, callStack, valueFrame.`

Hlavní část programu je ve funkci `syntaxCheck,` který plní hlavní funkci interpretu. Primární funkci plní cyklus `while`, který kontroluje, která instrukce má být zrovna na řadě a jaká sémantická instrukce se má vykonávat. Jednotlivé části kontroly typů proměnných, vyhledávání v rámcích, načítání ze souborů a podobné náležitosti jsou rozděleny do jednotlivých funkcí, které se ve těle cyklu volají.

Seznam instrukcí, které jazyk IPPcode19 podporuje je vypsaný v polích v jedné z funkcí a těle cyklu jde poznat, co jaká instrukce dělá.