Implementační dokumentace k 2. úloze do IPP 2020/2021

Jméno a příjmení: Vojtěch Vašíček
Login: xvasic31

Interpret

Hlavní soubor interpretu

Při spuštění interpretu dochází nejprve ke zpracování argumentů. Následně se volá funkce *parse_xml*, kde se nejprve zpracuje vstupní xml pomocí knihovny *xml.etree.ElementTree.* Dále poté probíhá zpracování samotných příkazů a argumentů vstupního xml, a to včetně jejich validace. Pro samotnou interpretaci byl zvolen dvoufázový průchod kódem, přičemž první průchod zajišťuje syntaktickou a lexikální analýzu a definici návěští. Druhý průchod pak zajišťuje sémantickou analýzu a samotnou interpretaci.

V tomto souboru se také vyskytují dvě pomocné funkce *check_order* a *order_args*. První funkce kontroluje pořadí instrukcí a případně instrukce seřadí podle jejich pořadí. Druhá funkce dělá to stejné, akorát pro argumenty jednotlivých instrukcí.

Třída Instruction

V této třídě probíhá analýza sebe sama, tedy instrukce, pomocí funkce *parse.* Tato funkce zároveň volá funkci ve třídě *argument*, která slouží ke kontrole argumentů. Dále se zde nachází funkce *interpret*, která už podle jména intepretuje tuto třídu instrukce. V této třídě také můžeme najít několik pomocných funkcí. Funkce *count_check* kontroluje očekávaný počet argumentů s reálným počtem, funkce *argument_parse* která pro všechny argumenty volá jejich sebe analýzu a funkce *jump*, která vrací pozici hledaného návěští.

Zároveň tato třída ukládá operační kód instrukce, její pořadí v kódu a všechny argumenty této instrukce.

Ostatní třídy

Třída Argument obsahuje pouze jednu funkci *parse*, která kontroluje hodnotu argumentu podle jeho typu. Tato třída si ukládá typ argumentu, jeho hodnotu a jeho značku, která značí jeho pořadí v instrukci.

Třída FrameStack slouží zejména k ukládání jednotlivých rámců, ale také ukládá řetězec návěští, zásobník a řetězec instrukcí. Jeho funkce *find_retype_var* a *find_var* slouží k nalezení proměnné v zadaném rámci a kontrola jejího typu. V první jmenované funkce dochází k případné redefinici proměnné na správný typ.

Třída Frame ukládá pouze řetězec proměnných, které jsou v daném rámci uloženy. Třída Variable ukládá jméno, hodnotu a datový typ proměnné a třída Label ukládá jméno a pozici návěští.

Testovací soubor

Hlavní tělo skriptu

Nejprve se opět začíná zpracováním argumentů, přičemž při absenci některých argumentů jsou použity definované hodnoty těchto argumentů.

Dále se iteruje skrze všechny soubory v zadaném adresáři. Případně i v dalších složkách zadaného adresáře, podle toho, zda byl zadán argument *--recursive.* Zároveň se při těchto iteracích volá funkce *run_tests*, která spouští zadaný test. Podle vrácené hodnoty této funkce určujeme zda byl test úspěšný nebo ne. V případě neúspěchu ukládáme adresu testu v adresáři do řetězce *fail*.

Testovací funkce

Funkce *run_tests* nejprve dogeneruje chybějící soubory s příponou *.in* nebo *.out*. Poté volá funkci buď na test skriptu *parse.php*, *interpret.py* nebo test obou skriptů a to podle zadaných argumentů.

Testování pouze analyzátoru probíhá ve funkci *run_p_tests*, která nejprve provede daný test a poté porovnává výstupní kód a výstupní xml s jejich zadanými protějšky. Výstupní xml se kontroluje pomocí nástroje *A7Soft JExamXML*.

Testování interpretu probíhá ve funkci *run_i_tests*, kde proběhne samotný test a opět se testuje jejich výstup se zadanými protějšky.

V případě testování obou skriptů, se nejprve *run_p_tests* a poté se volá *run_i_tests* s výsledky prvního testu.

Výpis výsledků

Výsledky testů se vypisují ve formě HTML a to na standardní výstup. Toto HTML reprezentuje stránku která vyobrazuje počet správných a špatných testů. V případě špatného testu se také ukazuje jeho adresa v adresáři.