

Implementační dokumentace k 2. úloze do IPP 2018/2019

Jméno a příjmení: Radek Duchoň

Login: xducho07

Interpret.py

Započato bylo načítáním argumentů příkazové řádky. Pro to byla využita funkce `getopt`, která v jistém smyslu i příjemně testování, jelikož lze zadat libovolný jasný podřetězec, který není společný pro žádný jiný argument. Načítání je prováděno ve funkci `argParse`, kde se zároveň kontrolují povolené a nepovolené kombinace argumentů a případně vypisuje nápověda při zadání přepínače `-help`.

Skript pracuje kromě argumentů `-input="file"`, `--source="file"` pro zadání vstupních souborů a `--help` pro vypsání nápovědy také s argumenty `-stats="file"`, `--insts` a `--vars` pro vypsání statistik do souboru s údaji o počtu provedených instrukcí a maximálním počtu najednou inicializovaných proměnných v platných rámcích.

Dále se načítá a zpracovává xml ve funkcích `xmlGet` a `xmlParse`, kde probíhá rovněž velká část syntaktických kontrol. Xml je nejdříve po řádcích načteno do proměnné a poté zpracováno pomocí `ElementTree` do stromové struktury, se kterou se poměrně dobře pracuje. Jelikož jsou všechno odkazy, lze si strom snadno přizpůsobit. Konkrétně je toho využito pro prohazování argumentů dle abecedy před kontrolou samotného jména a pro prohazování instrukcí samotných, aby byly ve správném pořadí a stromem se tak dalo při zpracování kódu jednoduše iterovat.

Po získání stromové struktury se začne kód interpretovat ve funkci `interpreter`, která přijímá parametry `arg` nesoucí údaje o vstupu a výstupu statistik, a `i` pro indexaci, které je předdefinováno na 0, které je pozůstatkem hraní si s myšlenkou rekurzivního fungování funkce při instrukci `call` kvůli snadnému návratu, od toho se však šlo pryč, protože by bylo problematická navracování se kvůli statistikám bez instrukce `return` a jejich následnému výpisu. Interpretace využívá třídu `ipp2019`, která obsahuje až na výjimky metody pro všechny instrukce a několik pomocných instrukcí bokem, aby se nemusely stejné chyby kontrolovat na mnoha místech a podobně. Tyto metody obvykle přijímají pouze 1 parametr – odkaz na samotnou instrukci ve stromu, kterou dále zpracují, výjimkou jsou pak instrukce jako `jumpifeq`, které zároveň přijímají aktuální řádek pro případ, že skok nebude proveden.

V průběhu bylo mnohokrát naraženo na to, že by se musela celá metoda napsat znovu s mírnými odlišnostmi, to bylo řešeno ve většině případů namísto toho přidáním předdefinovaného parametru funkce, který může upravovat její fungování – jako třeba změnit chybový kód, povolit či zakázat nil s další instrukcí, dodatečná kontrola hodnoty pomocí regulárního výrazu na chybu a tak dále.

V poslední fázi byly přidávány rozšíření, funkce z rozšíření většinou nebyly kompatibilní s dříve používanými pomocnými funkcemi pro svou odlišnost, kdy přijímají argumenty ze zásobníku a jelikož již nezbývalo mnoho času, byly pro zjednodušení napsány bez tvoření dalších pomocných funkcí a jejich části jsou si tak velmi často podobné.

Test.php

Ve skriptu `test.php` bylo taktéž započato zpracováním argumentů. To probíhá ve funkci `arguments`, skript přijímá argumenty `-directory="file"` pro nastavení složky s testy, `--recursive` pro rekurzivní prohledávání složky, `--parse-script="file"` pro nastavení cesty ke skriptu `parse.php`, `--int-script="file"` pro nastavení cesty ke skriptu `interpret.py`, `--parse-only` a `int-only` pro testování samotných skriptů a `--help` pro vypsání nápovědy.

Pomocí iterátoru jsou načteny všechny soubory ve složce a případně jejich podložkách, které končí příponou `.src` a v pomocné funkci `make_files` jsou následně zkontrolovány existence odpovídajících souborů s příponami `.in`, `.our` a `.rc`, které se případně pokusí dovytvořit s předdefinovaným obsahem.

Pomocí jednoduchého cyklu se následně generuje jméno, které nebude kolidovat s žádným jiným ve složce pro vytvoření dočasných proměnných, které budou na konci skriptu opět smazány.

Následně se vytvoří základ HTML stránky a v cyklu, který se jemně liší dle varianty (`--parse-only`, `--int-only`, nebo test obou skriptů), pro každý `.src` soubor, kde se kontroluje, zda odpovídá kód, se kterým skončil program s tím, který je v souboru `.rc` a pokud tento kód je 0, tak se kontroluje i výstup s k němu patřícímu `.out` souborem. Ohledně rovnosti či nerovnosti se následně vypíše kód tvořící přehlednou HTML tabulku s názvem testu a informací, zda prošel, jaký byl kód, který program vrátil a jaký měl být.

Na konec se vypíše konec HTML stránky a počet prošlých testů s celkovým počtem testů a vymažou se dočasné soubory vytvořené pro potřeby testů.