

Implementační dokumentace k 2. úloze do IPP 2020/2021

Jméno a příjmení: Ondřej Kepřt

Login: xkepřt03

Popis souboru `interpret_functions.py`:

V tomto souboru nalezneme funkce, které kontrolují získaný argument, zda odpovídá požadavkům instrukce (zda je argumentem proměnná, konstanta nebo jiné).

Funkce `get_argument()` je volána z konstruktoru objektů jednotlivých instrukcí. Jako parametry požaduje XML reprezentaci instrukce, pořadí, který poziční argument chceme získat a jeho typ. Zde jsou volány funkce `is_TYP` pro vykonání kontroly.

Třída `frame` je místo, kde se budou ukládat námi definované proměnné. V konstruktoru se vytvoří pouze prázdný slovník a pomocí metod s ním pracujeme. Metody pro práci s proměnnými frámci pracují pouze s daným rámcem, proto je třeba tyto metody volat ze speciálních funkcí, které za nás rozhodnou, se kterým rámcem budeme pracovat (`def_var`, `get_var` apod.). Veškeré hodnoty uložené v proměnných jsou typu `string`!

Popis souboru `interpret.py`:

Na začátku souboru definujeme globální proměnné, které jsou potřeba pro chod interpretu.

Funkce `log_inside()` byla využita převážně při ladění programu, ale je využita v instrukci `BREAK`. Funkce vypíše aktuální stav globálních proměnných.

Funkce `string_repair()` převádí escape sekvence z XML souboru na znaky, které jsou reprezentovány v Python řetězci.

Následují funkce, které vybírají správný rámec, se kterým budeme pracovat, pokud budeme chtít manipulovat s proměnnými.

Následují definice tříd pro dané instrukce. Třídy pro jednotlivé instrukce dědí metody od třídy `instruction`. Svoje mají definované pouze konstruktor a `run_instruction()`, protože jsou specifické pro každou třídu. V konstruktoru objektu dané instrukce se volají funkce `get_argument()`, pro uložení argumentů do objektu. Také se odkaz na daný objekt uloží do pole instrukcí, ze kterého jsou následně instrukce prováděny. Výjimkou je konstruktor třídy `label`, který kontroluje, zda bylo zadáno návěští již definováno, případně jej nadefinuje. Funkce `run_instruction()` provede operace dané instrukce.

Poté v souboru nalezneme zpracování argumentů programu a přiřazení, odkud bude načítat zdrojový kód a vstupy programu.

Následují kontroly XML souboru a definice přepínače, podle kterého budeme vybírat danou instrukci.

Poté načítám jednotlivé XML reprezentace instrukcí, na základě přepínače vybírám správnou instrukci provádím konstruktor dané instrukce. Současně kontroluji, zda XML reprezentace obsahuje všechny potřebné atributy.

Dostáváme se k cyklu, který vykonává instrukce. Pokud pořadové číslo neobsahuje žádnou instrukci pokračuje se další instrukcí. Skoky jsou realizovány změnou pořadového čísla `instr_walker` v těle instrukce.

