Implementační dokumentace k 2. úloze do IPP 2022/2023 Jméno a příjmení: Tomáš Běhal Login: xbehal02

Popis

Skript slouží pro interpretaci zdrojového kódu v IPPcode23. Pomocí argumentů na příkazové řádce se určuje, zda skript načítá XML, nebo vstup pro READ ze souboru nebo ze standardního vstupu. Skript vypisuje na standardní výstup výsledky instrukcí a na standardní chybový výstup hlášky o chybách. Skript je v jazyce python a je složen z různých souborů, propojených přes interpret.py.

\$ python interpret.py [--help] [--source=filepath] [--input=filepath]

Kde alespoň jeden z dvojce argumentů source a input musí být zadán. Pokud je zadán pouze help, tak se vypíše nápověda a skript se ukončí. Jinak se jedná o chybné spuštění programu.

Zpracovani argumentů

První částí skriptu je zpracování argumentů příkazové řádky pomocí knihovny argparse, v souboru arguments.py.

V třídě parseArguments se nachází metoda getArgs, která si volá vnitřní metody. Pomocí addArguments se přidávají argumenty do parseru. Pomocí parse args se argumenty zpracují a pomocí checkArgs se zkontrolují, které to jsou a otevřou se soubory, nebo se přesměruje standardní vstup.

Zpracování XML

XML schéma je zpracováno pomocí knihovny xml.etree.ElementTree, v souboru xmlparse.py.

Ve třídě Parse je metoda parse, ve které se nachází try blok, který zamezuje chybám při zpracování XML, nebo nenalezení souboru.

Pomocí metody getroot se získá kořenový element, metoda _checkRoot kontroluje jeho správnost a _checkInstructions seřadí instrukce podle pořadí a kontroluje duplicitu a validitu. Dále kontroluje, pokud instrukce mají validní počet argumentů a nejedná se o špatně očíslované argumenty. Přídá je do seznamu a vrací ho

Přeformátování instrukcí

Přeformátování instrukcí je zpracováno pomocí souboru instructions.py.

Seznam s instrukcemi je převeden na seznam instancí třídy Instruction , která obsahuje metody pro zjištění "opcode" , "order" a argumentů . Jednotlivé argumenty jsou převedeny na instanci třídy Argument , která obsahuje metody pro zjištění typu a hodnoty argumentu.

Interpretace instrukcí

Interpretace instrukcí je zpracována pomocí souboru run.py.

V třídě Run je metoda run, která prochazí instrukce podle aktualního ukazatele na instrukci, které můžou jednotlivé instrukce měnit a určovat tak tok programu. Všechny instrukce mají svojí privátní metodu, kde jsou zpracovány. V jednotlivých metodách se kontroluje validita argumentů instrukce. Pokud jsou validní, tak se vykoná konkrétní operace. V průběhu zpracování instrukcí můžou nastat různé chybové situace, pospané v zadání. Tyto chyby jsou zachyceny a vypisovány na standardní chybový výstup.

Ukončení skriptu

Jakmile skript úspěšně zpracuje všechny instrukce a skript se ukončí s návratovým kódem 0.

Nil

ErrorCodes

Enum of codes

exitMessage(errorC, errorM | "")

Main

lineArgs: namespace

instructions : list[Instruction]

run : Run

ParseArguments

args: None

get_args()

Parse

tree : None

instructions: list

parse(source): list



Instruction

order : str

opcode : str

args : list(Argument)

get_order()

get_opcode()

get_args()

len_args()



Argument

name: None

type : Any

value : Any

get_arg()

Run

instructions : Any

lineArgs: namespace

g_frame : Frame

I_frame : list

t_frame : None

data_stack : list

call_stack : list

labels : dict

SYMB : list[str]

LABEL : list[str]

TYPE : list[str]

run()



Frame

frame : dict

get_frame()

def_var(var)

set_var(var, value)

set_var_type(var, value, typ)

get_var(var)

get_var_value(var)

get_var_type(var)

get_var_tyoe_str(var)

write_frame()

write_var(var)