Implementační dokumentace k 2. úloze do IPP 2021/2022

Jméno a příjmení: Rastislav Duránik

Login: xduran03

1. Introduction

Assignment was to implement interpret.py script in python3.8. This script is supposed to interpret given XML file.

2. Implementation

I decided to try and learn how to code using OOP, since I have never actually used it before. Program is launched by interpret.py, which calls the instance of Interpret class, which is imported from body.py module located in interpret.lib directory. After that program parses arguments using argparse library and checks for input files, in case of missing source or input it loads the input from STDIN.

Program continues by checking xml file syntax, for instance parsing of file for which i used xml.etree.ElementTree library, instruction order number and argument order number checking according to assignment. After check, instruction objects are created using Factory class method, which is design pattern. It automatically creates new specific objects inherited from Instruction class and their arguments are being assigned to them by another Factory, which creates instances of Argument class ,as attributes.

Then labels are set and label instructions are replaced by instance of class, which only skips them for further use in STATI extension. After that, the main while loop starts looping and in each loop it executes 1 instruction. It breaks when it reaches None element at the end of an Instruction list which was put in there solely for this purpose.

After execution of all instruction, program ends either with exit code 0 (if there was no issue), or prints an error with its exit code on stderr according to assignment.

Due to the lack of time, although i have never underestimated this assignment, I did not implement test.php .

3. Extentions

As i mentioned above, I tried to apply the OOP technique and used Factory design pattern, therefore NVI extension was implemented. Furthermore I implemented STACK and STATI extension.