Documentation of 1.st Project Implementation for IPP 2019/2020

Name and surname: Daniel Weis

Login: xweisd00

Iplementation:

At first, I created a while loop which I used to load an input from `STDIN` using function `fgets`. The next step was handling the input, I needed to replace all unnecessary white space and tabs with just one space. With having just one space between inputs I used function `explode` using " " as delimiter, which created an array of my inputs separated. The main construction of my script is a switch. Every instruction is a `case` in the switch. I separated instruction/cases into four groups by number of arguments they have (so with 0, 1, 2 or 3).

The first xml function was a xml header function where the xml version, encoding and our programing language IPPcode20 is displayed in the XML code. Then I also created four xml functions for every instruction type (also by number of arguments of instructions). The hardest part of giving them correct arguments was with `ins_type_X` because it had to be an output from regex function that I needed to create.

The regex function (`is_it_right`) contains 7 `if` statements, one for each argument type (var, int, bool, string, nil, type, label). The regex function checks if the syntax of represented instructions with arguments is correct, if yes it returns the correct instruction type that is used as an argument of xml function when it is called in the switch. In the end an `xml_end` function is called which ends the first element that was created (program) and closes the XML files and also prints out the xml code. The last thing I did was taking care of the correct exit codes, e.g. in switch as default or in syntax checking through regex function.