Documentation of Project Implementation for IPP 2022/2023
Name and surname: Michal Ondrejka
Login: xondre15

The main part of the script reads the input code line by line from the standard input and removes any comments from the input code. Then, it verifies the header of the input code, and generates the opening XML tags for the program. Next, it reads and splits each line of the input code by whitespaces, switches between individual instructions, since they have different argument counts and types. Using function checkCount I check if there is correct amount of arguments and then I proceeded to printing them using printVar, printSymb, printLabel and printCount.

The functions included in the script are as follows:

1. `removeComment($code)`: This function removes any comments from the input code. It takes a string as an argument and returns a string with any comments removed.
2. `printVar($var, $index)`: This function prints the argument of an instruction if it is a variable. It takes two arguments: the variable to be printed and its index, and generates the corresponding XML tags.
3. `printSymb($symb, $index)`: This function prints the argument of an instruction if it is a symbol. It takes two arguments: the symbol to be printed and its index, and generates the corresponding XML tags.
4. `printLabel($label, $index)`: This function prints the argument of an instruction if it is a label. It takes two arguments: the label to be printed and its index, and generates the corresponding XML tags.
5. `printType($label, $index)`: This function prints the argument of an instruction if it is a type. It takes two arguments: the type to be printed and its index, and generates the corresponding XMLtags.
6. `checkCount($splitted_line, $count)`: This function checks whether the number of arguments of an instruction matches the expected count. It takes two arguments: the array of arguments of theinstruction and the expected count.

The script also contains a series of conditional statements that check the input code's syntax and generateerror messages if the code is invalid.

The output of the script is an XML representation of the input code with additional information, such as the order of the instruction and the instruction's opcode. The XML output starts with the opening `<program>` tag and ends with the closing `</program>` tag. Each instruction is represented by an **<instruction>** tag with its opcode and order attributes, and its arguments are represented `<arg>` tags bywith their type attributes.