Implementation documentation for the 1. task for IPP 2018/2019
Name and surname: Sabína Gregušová
Login: xgregu02

Introduction

My task was to create a script in PHP for lexical and syntactical analysis of language IPPcode19 that is derived from IFJcode18.

My Solution

I started implementing `parser.php` as soon as possible, so that I would have more time on other projects during the semester. My script `parser.php` starts with opening file that was given in STDIN or accepting input from STDIN. After successfully opening/accepting the input, the script reads the first line to confirm that it has the correct header. If the header is correct, I initialize memory for the XML output by using functions from `XMLWriter`, which formats the output accordingly. It only outputs the XML document if no error was encountered during lexical and syntactical analysis.

The input is read line by line and I've used regular expressions to extract the opcode and check whether it's acceptable. I implemented functions that handle zero, one, two or three arguments and each of them checks whether the argument type matches the required type(s) for given opcode. Each argument is checked lexically and I've implemented function `write_arg` in order to make writing the arguments in XML easier and quicker.

While checking the arguments, I made sure to also check the number of arguments. Script can finish successfully only if there's exactly the number of arguments that the instruction needs. When evaluating each argument, scripts also checks the types and whether it matches the real input. I edited the regexes, so that there can be multiple white spaces between operands and comment can be added after the last operand without any whitespace, which means that as long as the input file has correct opcodes and arguments, the format really doesn't affect correct functionality.

Bonus task

I have decided to implement the bonus task STATP, which was quite easy due to the organizational structure of my script. Each statistic has its own counter that is written to the output file and I keep track of all the necessary statistics inside an array.