Documentation of Project implementation
1. task for IPP 2018/2019

Name and surname: Viktória Halečková
Login: xhalec00

Lexical and syntactic analysis of code IPPcode19 (parse.php)

Script loads source code written in IPPcode19 from standart input, checks lexical and syntax corectness and prints XML representation of the program on the standart output.

Implemented functions

There are auxiliary functions implemented to make code easier to read. Function `help_info()` is called, when user runs program with argument - *-help*. On the other hand, function `check_arguments($argv, $argc)` takes care of arguments, which user runs the program with.

Array of instruction names

Instruction opcodes in program are implemented with an associative array `$ins_array`. Keys to these opcodes represent their needed values as input arguments. For example: `"int2char"-> "vs"` (instruction INT2CHAR needs two arguments, variable and symbol).

Regular expressions

Regular expressions are used as patterns to check corectness of the program. Functions such as `preg_match()` are implemented for work with strings. There are auxiliary variables such as `$empty_line, $comment_line` and many more.

Main program

Main program is implemented as while loop, which is reading lines from standard input. First of all, cuts whitespaces and checks the corectness of header. In case of right header, xmlwriter creates element `program`, sets language and header. Then program takes care of comments. Checking corectness of instruction name leads to adding xml element, especially `order` and `opcode`. After it is all about work with arguments, checking their count and validity and adding it to xml output.

XML

There are many ways how to parse XML output format of program. I have chosen XML writer. There is implemented global variable called xml. Main program uses functions for elements and attributes. Then is XML representation printed on standart output.

Error handling

Implemented function `exit_program($return_value,$message)` takes care of error handlings. This function returns 3 different values, 21 for non defined or non valid header, 22 for unknown or misspelled opcode, 23 for other lexical or syntactic errors, for example not enough or too much arguments used with specific instruction.