Implementační dokumentace k 1. úloze do IPP 2018/2019
Jméno a přijmení: Dominik Juriga
Login: xjurig00

parse.php

This script reads IPPcode19 lines from standard input and returns XML representation to standard output. It takes care of lexical and a part of syntactical analysis. Every line is checked for multiple traits (such as jumps, labels, comments). Comments are being stripped from every line. If the given line is valid and contains an operation, statistic of $stats->numberOflines is raised by 1.

This implementation uses 4 classes .

`err`	Calls errors.
`instruction`	Represents an instruction and it's arguments.
`instructionList`	List of all possible instructions and their argument types.
`statistics`	Stores code statistics for STATSP extension.

Available instructions are stored in an 2D array inside the instructionList object. Each entry contains correct argument count and argument types. Regular expressions were used for comparing these traits. Every instruction from source is kept in a separate object derived from instruction class. Syntactical analysis is performed afterwards. There are 2 functions taking care of it - validateTokens and validateParameters.

`validateTokens`	Checks opcode and argument count of an instruction.
`validateParameters`	Checks types of parameters and writes instructions to standard output after a succesful check.

If there were no errors, XML code is printed into the standard output. I have decided not to use the short form of certain instructions (such as `CREATEFRAME`).

parse.php STATP extension

This extension has been implemented as a statistics class. $stats object* is instantiated after a succesful *.IPPcode19* header check. By using regular expressions, every line of code is compared. If a certain trait is found, it's count is raised in the *$stats object. At the end of the program, if no errors have been found, requested statistics are written into a file.