Documentation of Project Implementation for IPP 2020/2021

Name and surname: Dušan Čičmiš
Login: xcicmi00

Objective

The goal of the project is to create a set of 3 scripts (parse.php, interpret.py, test.php) and interpret the structured and imperative language IPPcode21 using them.

Script parse.php

The job of this script is to check the lexical and syntactic analysis. It loads the program from the standard input and in case of correct code in terms of syntax and lexis, outputs XML representation of the code to standard output. There is a possible extension, which allows user to collect various statistics related to code.

The main part of the script is based on a never-ending loop, which is based on loading of a single line of code until there is no line to work with. On every line, the script checks the correct writing of the instruction and in the case of mandatory arguments of the instructions calls function `ArgCheck`, which decides whether the type of a variable and its value is correct. If every line of code is written correctly, it outputs the XML representation of the code, which includes the order and name of instruction (opcode), as well as the value of given arguments.

The extension **STATP** allows user to output desired statistics to given output file in the case of correct usage of the command line arguments passed to the script (`--stats` followed by wanted statistics). There are seven possible statistics to be gathered: number of labels, jumps, comments, instructions, back jumps, forward jumps and bad jumps. The order of these statistics must be preserved, t.j. they have to be outputted in the same order as they were written to command line. Function `customGetOpt` checks every argument given and stores them into array `arguments`. For every `--stats` argument, there has to be same number of valid output files, what is also handled by function `customGetOpt` (stores the name of the files in array `files`). This function then allows function `checkArg` to write desired statistics in correct order, by looping through the array `arguments.`