Documentation of 1st Project Implementation for IPP 2018/2019

Name and surname: Natália Holková

Login: xholko02

1 Introduction

This documentation describes my implementation of script `parse.php`, whose main task was to perform lexical and syntax analysis of provided code and output its XML representation.

2 Implementation

I have chosen to divide my implementation into several parts:

2.1 Lexical analysis

Lexical analysis is carried out by function `line_lexical_analysis()`, which takes as input single line of code (single instruction) and return an array of tokens the instruction consists of. For this purpose, I have created class Token, which has attributes such as token type or its attribute and several functions related to determining the type of token. In the case of lexical error function promptly stop program execution.

2.2 Syntax analysis

After lexical analysis function `line_syntax_analysis()` begins syntax analysis. Its job is to check if first instruction begins with valid opcode – therefore the first token in the returned token array has to be of correct type. Afterward, it compares the number of arguments provided for instruction with its correct number. In case of the very first line of code, it checks for a correct header.

2.3 Generating XML

I have used `DOMDocument` class for generating XML. At the start of the program, header and main node of XML is created. The XML representation of instruction is added right after its syntax analysis. It once again works with the token array to first determine opcode attribute and then type of each argument.

3 Extensions

My solution implements the STATP extension. All statistics are stored in a single object from class Statistics. Apart from storing all numerical statistics output file location, it also has to remember the order of output.