

Documentation of Project 1 Implementation for IPP 2022/2023

Name and surname: Matyáš Strelec

Login: xstre103

Introduction

This script is a PHP implementation of a parser for the IPPcode23 language. The script reads in IPPcode23 instructions from standard input and checks syntax and outputs the corresponding XML representation of the instructions to standard output.

Implementation

The code is documented using doxygen comments. Included here is the overview of the implementation, see the source code for more details.

First, the main program checks whether the argument "`--help`" is present and prints out the help message if it is. Then, it removes any unnecessary spaces and comments from the input file.

Once the header has been verified, the program goes through each line of the input. Each instruction is written to the output XML file using the `Writer` class. The program ensures that both the syntax of the instruction and its arguments are correct using the `Parser` class.

If the syntax is correct, the instruction is written to the output. To check syntax, the program uses a switch statement with cases for instructions that share a syntax check, grouped by the types of variables the instruction accepts. The program also ensures the correct number of elements are present. If the syntax is incorrect, the program prints an error message and exits with an error code.

There are two classes used in the program, to make the code more readable and easier to maintain.

Parser

This class contains methods for parsing and validating instructions and arguments used in the input script. It includes the following methods:

`check_number_of_tokens()`: checks if the number of tokens matches the expected number and exits with an error if not.

`check_var()`, `check_label()`, `check_symb()`: checks if the given variable token is syntactically correct, performs different checks to ensure the variable is valid depending on the type of the argument (symbol, type, variable).

`check_type()`: checks if the given type token is one of the allowed types (int, bool, string, nil).

All of these methods exit with an error code if the syntax is incorrect.

The class also contains two global variables for storing regular expressions used in the script for variable and label validation.

Writer

This class provides methods for generating XML representation of the given code, using `XMLWriter` library helps generate syntactically correct XML. It contains methods, including:

`create_xml_writer()`: creates an object of the `xmlwriter` class and sets up the indentation.

`write_xml_header()`: writes the XML header to stdout.

`begin_xml()`, `finish_xml()`: creates the XML writer object and writes the XML header or finishes the XML to stdout, respectively.

`write_arg()`: writes the XML of an instruction argument. It recognizes the type of the argument and writes the corresponding XML element with the appropriate attributes and value.

Conclusion

This PHP script provides a basic implementation of a parser for the IPPcode23 language. It uses a set of helper functions to check the syntax of various elements of the language and output the corresponding XML representation of each instruction to stdout.