

# Implementační dokumentace k 1. úloze do IPP 2022/2023

Jméno a příjmení: Dávid Jókay

Login: xjokay00

## General

The project is divided into two scripts, where `'parse.php'` serves as the main script and `'const.php'` serves as a helper script containing constants and various macros. My goal was the object-oriented approach. I decided to try to implement the “Chain of responsibilities” design pattern. The program also includes a debug mode, which allows displaying the line of code where the program failed. To test my script I used the tests provided in the assignment and also various testing scripts found on discord. I also used the opportunity of earlier submission where I verified the correctness of my implementation.

## Classes

The **InputValidator** class is responsible for input validation tasks such as checking input arguments, verifying headers, and removing comments.

The **Analyzer** class is responsible for the syntactic validation of the program. It uses the auxiliary script `const.php` to check the program's structure, the correctness of instructions, and their arguments.

The **IPPcode23** class takes the validated input and translates it into XML language. The result is then returned to the output. To generate xml output I am using the **SimpleXML** library and its object **SimpleXMLElement** in `toXml()` function that runs in for loop and generates xml output using methods like: `addChild()` and `addAttribute()`.

## Extension

The idea behind the extension is illustrated in the diagram below, which shows how the initial input is passed from object to object with the goal of implementing the "Chain of Responsibilities" design pattern. Output of each handler is taken as the input of the next one resulting in “chain” that I believe is a way of implementing this design pattern.

