**Implementation documentation for the 2nd task in IPP 2022/23 university course**

Author: Alina Vinogradova  
Login: xvinog00

**interpret.py script**

The script was implemented using object-oriented programming in the Python language. The most basic and main class that represents the project is Interpret, which connects all the other implemented modules.

After running the program, the first method called is the parse\_args() method, which processes the arguments given with the script. If successful, the data from these arguments will be collected: the file with the source XML code and/or the file with the input for the interpreter.

Next, the main program method interpret\_start() is called, which first creates an instance of XMLParse class, passing during initialization data about where the file with XML code is located. The parser checks if the structure of the XML tree is correct, then processes the tree one instruction at a time, checking it and its operands (in the context of the tree element). In case no errors are found, the parser creates an Instruction class instance, more specifically its subclass, according to this instruction and then writes the created object to the instructions attribute, which is an instance of the InstructionsList class, and which is subsequently used by the main interpret program.

After all the raw data have been successfully processed and transformed into the necessary objects for the interpretation, the script processes the objects from the InstructionsList by calling the get\_next\_instruction() method within the while loop until it reaches the end of the list (in my implementation, in this case method returns None), successively executing each instruction and updating frames and possible variables within them, as well as call stack, data stack and other attributes responsible for the current state and the future progress of the interpretation.

