Implementační dokumentace k 2. úloze do IPP 2022/2023

Jméno a příjmení: Denys Petrovskyi
Login: xpetro27

1 Program structure

At the beginning of the code global variables and 2 classes(`Instruction` and `Argument`) are initialized. First one is used to store all the data about instructions that program will get, second class is used to store data about arguments that each instruction has. After classes there are functions, all the analysis is in function called `interpret()`, other functions are helping it.

2 Implementation procedure

2.1 Main function

Program starts in `main` with parsing arguments. To implement this was used import `sys`. After that program starts working with source file. Using import `xml.etree.ElementTree` and loops program goes through *XML* code and detects errors, then all the instructions and arguments are being written to objects of classes, objects of class `Instruction` are being added to the `listOfInstructions`.

2.2 Preperation before analysis

After that function `mySort()` is being called. Its purpose is to sort all the instructions based on their order which was given in *XML* code, so they would have the right order in the list. Then there is function called `fisrt_walktrough()`, it is being called before the main analysis starts, so it can detect all the initialized labels and add the to the list.

2.3 Interpret

If the previous code passes without errors, the function `interpret()` starts. Using a *for* loop it goes through every instruction in the `listOfInstructions` and then depending on the name of each instruction it performs analysis. There are several functions that help with analysis:

- `check_arg_amount()` checks if the right amount of arguments instruction has.
- `check_var_existance()` function goes through each frame (*global, local, temporary*), which are defined as dictionaries, and looks for a specified variable.
- `check_var_value()` checks if variable has a value and returns it if such exists. Works similarly as `check_var_existance()`.
- `check_var_type()` gets type of variable and returns it.
- `control_escapeseq()` is created to change all the escape sequances in string to *ASCII* symbol.
- `find_inst_by_order()` function gets the instructions position in `listOfInstructions` based on its order.
- `add_value_to_var()` adds value to the variable.

After processing all the instructions, program ends.

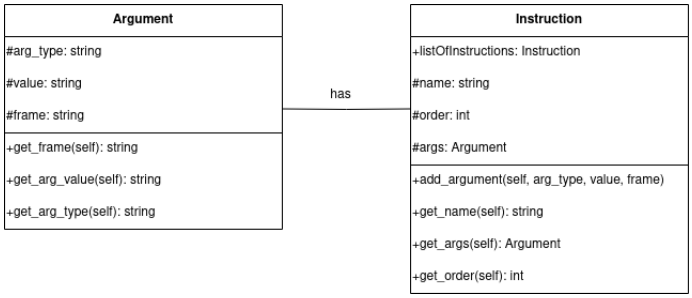

Figure 1: UML class diagram