Documentation of Project Implementation for IPP 2022/2023
Name and surname: Jan Osuský
Login: xosusk00

Task:

The task of the first part of the project to IPP was a complier interpret.py that is suppose to interpret IPPcode 23 converted to its XML representation.

Solution:

For my solution included ithe implementation of the OOP design. Therefore the final solution consist of 8 classes linked to the main file of interpet.py; In file intepret.py are loaded Arguments using the class Args. This class parses arguments checks its validity and in case of presence of –stat argument also creates a object of Stats class that is used to count stats. Later is created the object of Control that contains the main method of the program interpet. This method calls all the ohter methods for parsing the xml creating list of instruction etc.

Design patters

Almost all of the classes implements a design pattern. Mainly the singleton pattern is applied as it is implemented by all the classes except Instruction class. For the case of a interpet I find quite hard to implement other patterns. We just need more require just one instance of interpet etc. Thankfully I found one spot where an other patter could be applied and it is in InstructionLIst. InstructionLIst it self implements Singleton patter, but at the same time InnstructtionLIst implements Factory pattern for the Class Instruction at least partially.

Resume

Overall intepret.py an fully working program that suits all the requirements, however it’s definitely not perfect. The design could be better performed, because at the end of the project it leaded to a lot of duplicated code.