Implementation Documentation for IPP Project 2017/2018
Name and surname: Maros Orsak
Login: xorsak02

This is documentation about my semestral project from IPP. My process of doing this project was decomposed into few main parts.

1. Specifications
- As usual my first part was about design a specify specifications and then transform it to the programming part
- I was using trello to write some checklists and i divided my project into logical parts.

2. Detailed design
- 1.part
  - writing in structural way
  - my plan was to do this with only using regexes
- 2.part
  - combination of OOP and structural
  - created class diagram and i was thinking about data structures which i
  - will use in implementation
  - was harder and my trello checklist was huge

3. Implementation
- 1.part
  - i design finite-state machine with help of regexes and also functions to catch all errors
  - detailed description is in the script
- 2.part
  - In this part i was using data structures like dictionaries , stacks , lists etc.
  - Python was new for me so all this stuff was a little bit tricky
  - Also in this part i designed finite-state machines with help of functions
  - MileStones :
    - Xml and Argument class
    - Error class
    - Frame classes
    - Instruction classes
      - Make 33 instructions (not done)
    - Arg Class

4. Testing
- Testing was a member of implementation
- When i reached milestone i immediately tested that concrete module

Conclusion
- I was great programming experienced for me and i learned a lot
- Furthermore i discovered that ,, *A little knowledge is a dangerous thing*''.
- My Advice for everybody is ,;*Broaden your horizons and be flexible*''

Class Diagram of 2.part of project Interpreter.py

Main

Instruction

interpret
+ parseArgs(): int
+ xmlParse(): string

Frame

instruction
- arguments: [ ]
- nameOfInstruction
- options: map
- orderOfInstruction: int
+ choiceOperator(): void
+ inicialization(): void
+ parseInstructions(): void

frame
- dictOfVariables: dict { }
+ addVariable(): void
+ getVariableValue(): dict { }
+ printVariableValue(): void
+ setVariableValue(): void

Args

args
- argElement: string
- text: string
- type: string
+ argBool(): void
+ argInt(): void
+ argLabel(): void
+ argString(): void
+ argType(): void
+ argVar(): void
+ inicialize(): void
+ parseInstructArgument(): void

parseArgsAndXml
- myInstruction: Instruction
+ parseArgs(): int
+ xmlParse(): string

instructions
- arrayOfInstructions: int
- stackOfInstructions: int
+ interpretInstructions(): void
+ parseInstruction(): void

frames
- arrayOfFrames: [ ]
- globalFrame: Frame
- tempFrame
+ addVariableToTheFrame(): void
+ createFrame(): void
+ getLenFrames(): void
+ getVariableType(): void
+ getVariableValue(): void
+ isLocalFrame(): void
+ isVarInGlobalFrame(): void
+ isVarInLocalFrame(): void
+ isVarInTempFrame(): void
+ popFrame(): void
+ pushFrame(): void
+ setVariableValue(): void

«import»

«import»

Err

err
+ retErrMessage(): string
+ retMessage(): string
+ retMessageAndValue(): string , int

«import»

«import»