IPPe Project

3 Address code interpreter

Guillermo Montes Martos - xmonte03

Project implementation: two files

tacy.py

- Main script of the project.
- Creates an object Interpreter for the XML we want to process, passed as argument in command line.

interpreter.py

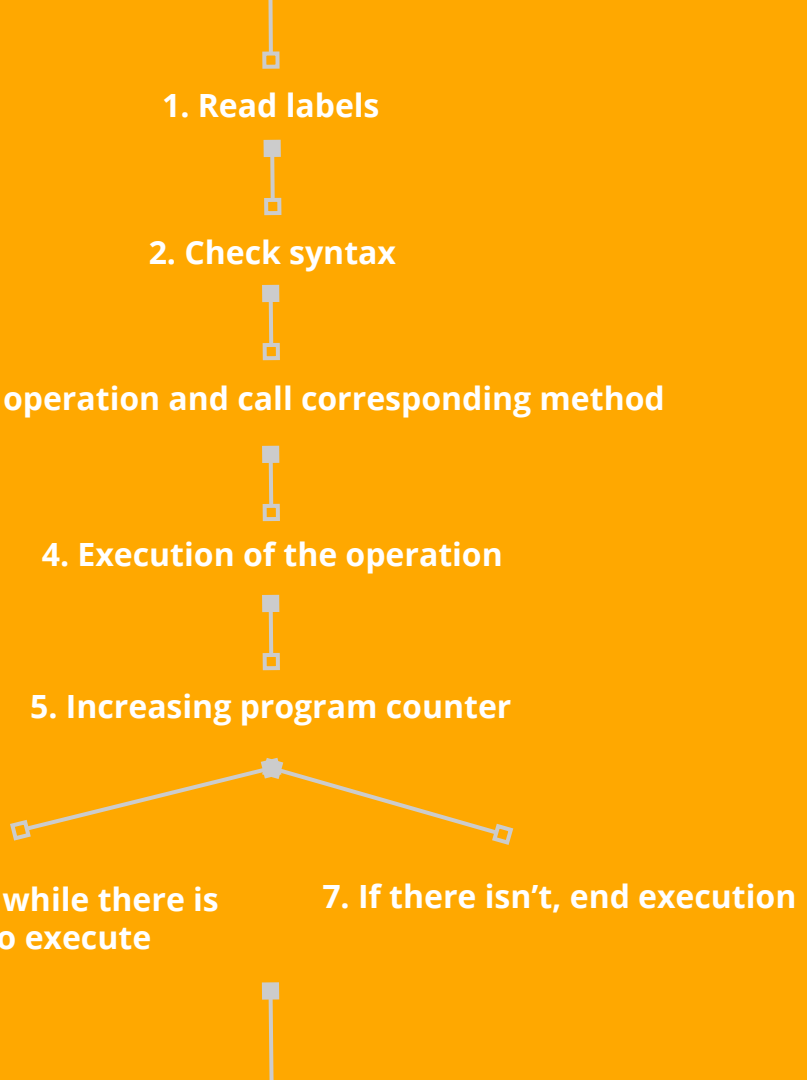
- Class interpreter.
- Needs a string with the name of the XML file.
- Object oriented programming solution
- Facilitates reutilization of the interpreter in future possible implementations
- An object Interpreter is needed for every XML file

Class Interpreter

Interpreter
+ program: xml.etree.ElementTree.Element + pc: int + variables: dict + labels: dict + data_stack: array
+ __init__(string): void + run(): void + read_labels(): void + check_args(): void + get_src_value(xml.etree.ElementTree.Element): int or string + mov(xml.etree.ElementTree.Element): void + add(xml.etree.ElementTree.Element): void + sub(xml.etree.ElementTree.Element): void + mul(xml.etree.ElementTree.Element): void + div(xml.etree.ElementTree.Element): void + read_int(xml.etree.ElementTree.Element): void + print_(xml.etree.ElementTree.Element): void + label(xml.etree.ElementTree.Element): void + jump(xml.etree.ElementTree.Element): void + jumpifeq(xml.etree.ElementTree.Element): void + jumpifgr(xml.etree.ElementTree.Element): void + call(xml.etree.ElementTree.Element): void + return_(xml.etree.ElementTree.Element): void + push(xml.etree.ElementTree.Element): void + pop(xml.etree.ElementTree.Element): void + readstr(xml.etree.ElementTree.Element): void + concat(xml.etree.ElementTree.Element): void + getat(xml.etree.ElementTree.Element): void + len(xml.etree.ElementTree.Element): void + strint(xml.etree.ElementTree.Element): void + intstr(xml.etree.ElementTree.Element): void

Methods

- Constructor
- Run → main function
- Read labels
- Get source value
- Operation methods



Interpreter.run()

Control flow

Future improvements

Debugger

Creation of class Debugger, which would contain:

- Object Interpreter
- Additional information

Allow us to find errors in 3 address code programs and delete them.

More data structures

Adding support for other typical data structures like:

- Float
- Boolean
- Array
- Dictionary

Conclusions

No found errors

Any error has been found during testing the interpreter. It is working properly according to the specification.

Extensions

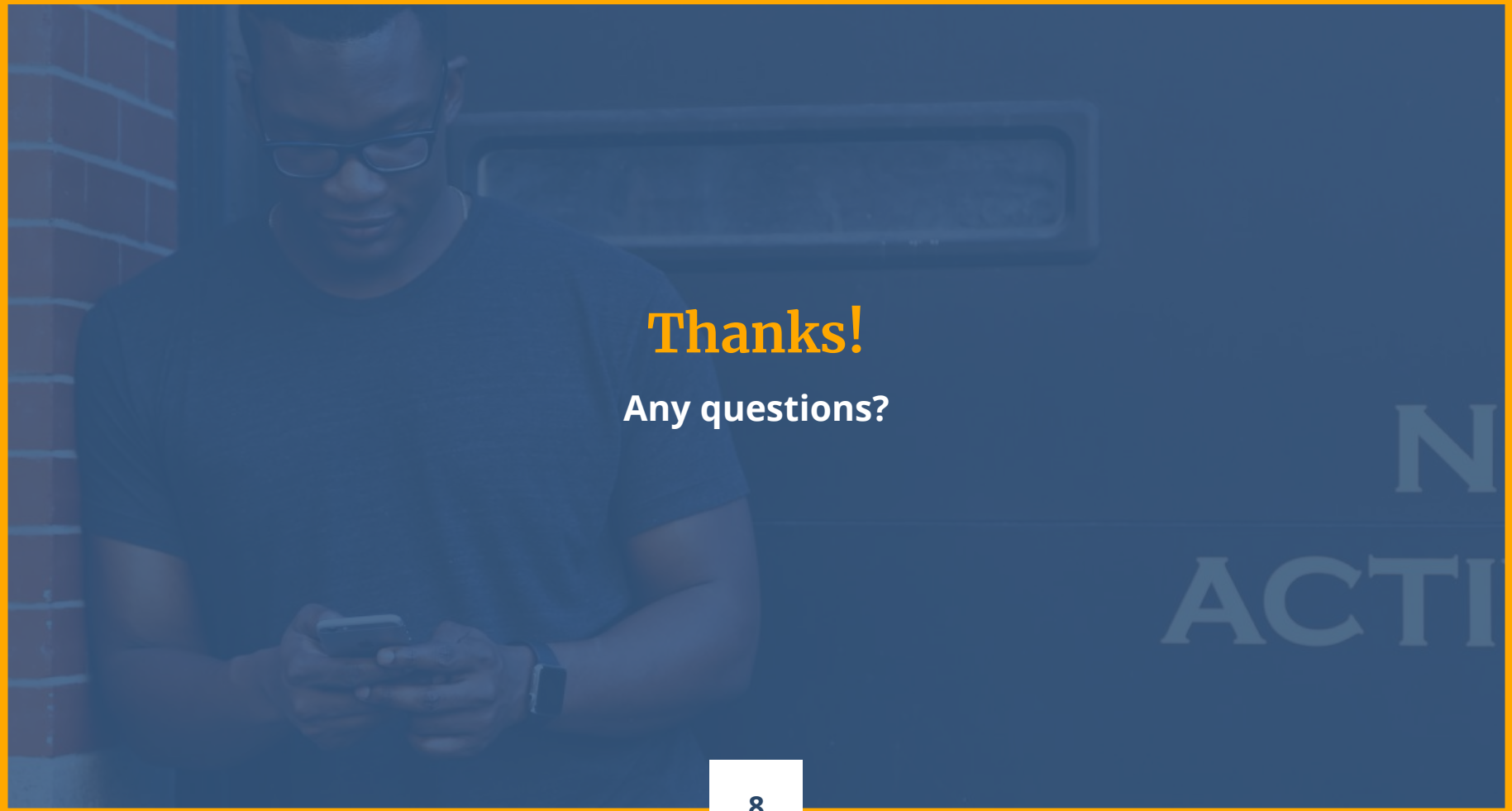
Support for functionality with strings.

Reutilizable solution

Thanks to the OO programming paradigm.

Difficulties

Some troubles at the beginning to understand the goal of the project and the 3 address code language.



Thanks!

Any questions?