Task documentation CLS: C++ Classes in Python for IPP 2016/2017

Name and surname: Róbert Kolcún
Login: xkolcu00

April 12, 2017

1 Problem & Program structure

Main objective in this project is to implement C++ classes in Python. These classes are declared and can inherit from each other. This program is divided into several sections(scripts). Main script is *cls.py*. Script *cObjects.lib.py* contain several Python classes, which represent C++ class, method, argument and attribute. Next one *argumentParser.lib.py* is responsible for parsing input arguments, reading from input file and writing to output file. Classes located in *parser.lib.py* script, parse input file to tokens and check format of input file.

2 Argument parser & I/O

Class *ArgumentParserAndFile* located in *argumentParser.lib.py*, parse input arguments from command line by function *parse(self)*. Program support only long type of input arguments, for example –help. This class also work with input and output files. Input data are read from input file or if is not given, from stdin. Also output data are written to output file or if is not given, to stdout. This class contain two methods responsible for correct data in output format. First method *printTree(self, classesData)* is used if argument *–details* is not set and write tree of C++ classes to output file. Second method *printDetails(self, classesData)* is used if argument *–details* is set and write tree of C++ classes, methods and attributes to output file.

3 Token parser & State machine

Method *__init__(self, inputText)* located in *TokenParser* class parse input file to tokens by regular expression:

```
output = re.split('(\ |{|}|\(|\)|=|;|::|:|,|\n|\*+|\&\&|\&|\~|\t)',inputText)
```

This array is filtered because contain spaces, tabs and new lines. And it is reversed.

```
self.tokenList = [x for x in output if (x != "" and x != " " and x != "\t" and x != "\n")]
self.tokenList.reverse()
```

Next class *ClassParser* contain state machine in method *parse(self)*. This state machine is represented in while loop, this loop end when token list is empty. State machine contain 14 states. 6 states are responsible for correct format of classes and inheritance. Special states are for *using*, *destructor* and *constructor*. Next state read name of attribute or method and special state is for correct format of data type. And last 3 states are for method, method arguments and end of the class.

4 C++ Class representation & Inheritance

Script *cObjects.lib.py* contain several Python classes that represent C++ Class, method, attribute and method arguments.

Python class *C_Class* represent C++ class, this Python class contain name, kind, dictionary of children, dictionary of methods, dictionary of attributes, dictionary of attributes that are inherited by term using and list of classes that this class inherit from. Also this Python class contain important methods as *inheritUsing()*, *inheritEverything(self)* and *addMethod() or addAttribute()*.
Next Python class *C_Method* represent C++ method, this class store method's name, data type, scope, dictionary of method arguments. Also contain information if is virtual, pure virtual, constructor, destructor and if is printable. Method is not printable if it was private in the class that new class inherit from.
Class *C_Argument* contain only simple information as name and data type.
And last class *C_Attribute* store information about it's name, data type, scope, inherit from and also if is printable. Attribute is not printable in the same case as method.