MTSubreflector Communication Class

Ivan Sharankov

November 2019

1 Example of Checked Exception

- IOException : It is thrown when an input-output operation failed or interrupted .IOExceptions are thrown when there is any input / output file operation issues while application performing certain tasks accessing the files. IOException is a checked exception.

```
print("test")

def funct(s):
    return s+1

Class NOPE():
    def __init__(self):
        self.nope = False
        x = 45452
    # This is a comment
```

- FileNotFoundException : This Exception is raised when a file is not accessible or does not open.

2 Example of Unchecked Exception

- Arithmetic Exception : It is thrown when an exceptional condition has occurred in an arithmetic operation.

3 Bugs

Due to the short time frame of the project, and since some of the concepts I used to build this were very new to me, there are some noted bugs in my software that are important to mention. Of course, I tried my very best to minimize these bugs, as well as spent many days trying to build and restructure the whole project to be as clear and logical as possible.

3.1 socket timeout on first instance of resetting the connection to the subreflector

This is a known bug, though to my knowledge, it does not impede or affect any of the code other than a socket time out. This only happens once, which is during the very first call to reset the connection with the subreflector. At time of writing, that is with the command EFFELSBERG:MTSUBREFLECTOR:OTHER:RESETCONNECTION. Once called, a method in the main program (subreflector_program.py) that calls a method in subreflector_client.py, which triggers a flag that starts the cleanup process of closing the socket. Once this is complete, the initial method from the main program then calls make_connection, which reconnects to the subreflector. The issue is that the first time this happens, the handler thread that returns messages to the user is blocked, and a socket timeout is reached. The connection is still reset, but due to the timeout exception, it seems like it is not. This only occurs on the first instance of RESETCONNECTION, and not subsequent ones. After some debugging, I discovered this is because two threads alternate control of the handler, this can be seen by the debug log, and taking not of the thread ID and how they swap. To my understanding this is an issue with an improper implementation of the threading, which proved to be a difficult module to implement correctly, including proper mutex and thread safe code; as it is very easy to implement threading incorrectly.

3.2