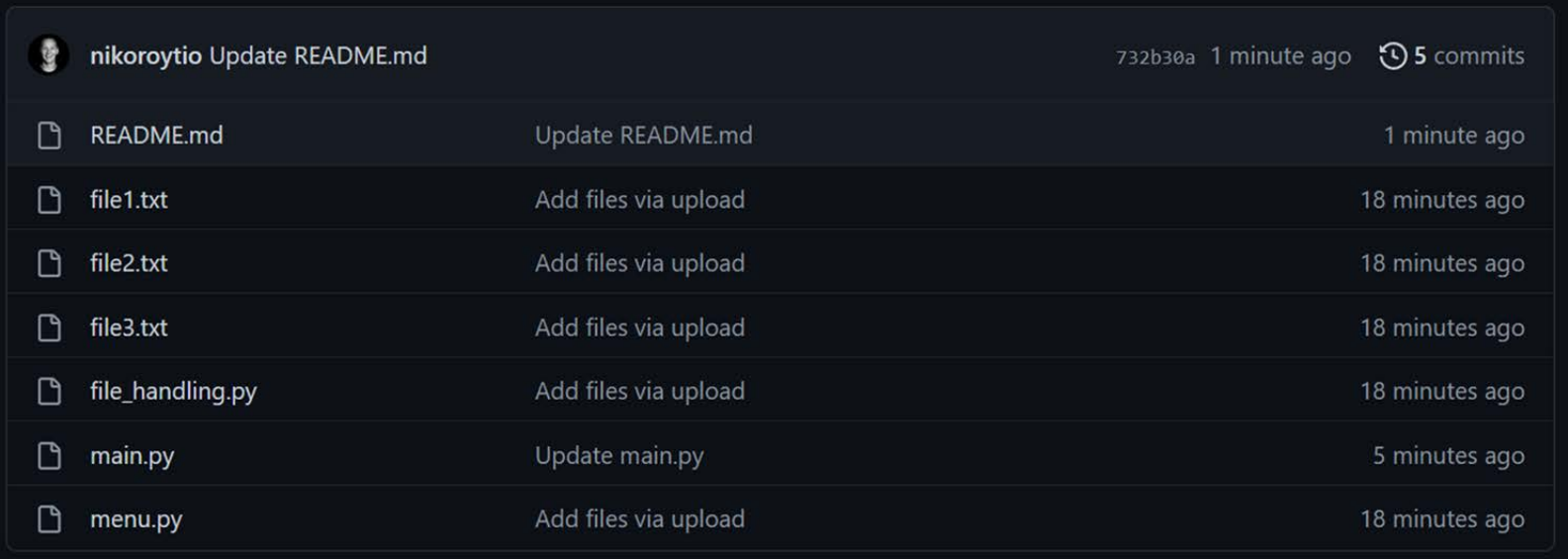
Linkki repoon: https://github.com/nikoroytio/L4



nikoroytio Update README.md — 732b30a — 1 minute ago — 5 commits

README.md	Update README.md	1 minute ago
file1.txt	Add files via upload	18 minutes ago
file2.txt	Add files via upload	18 minutes ago
file3.txt	Add files via upload	18 minutes ago
file_handling.py	Add files via upload	18 minutes ago
main.py	Update main.py	5 minutes ago
menu.py	Add files via upload	18 minutes ago

L4

Ylimääräistä viikkotehtävää varten oleva repo lähdekoodille

OP - File handling Idea of this example is to look at the file handling by using Object-oriented principles within the Python programming language. Examples include structural way (no OOP applied) and a oop way implementation of file handling. OOP Example consists of FileHandler class which is used to create file handling objects that are suppose to make things more understandable.

OOP - Example explained In the examples folder there are 3 important files. main.py, file_handler.py and menu.py. Then there are 3 empty text files: file1.txt, file2.txt and file3.txt The main.py is used to start the program. It defines the files in the project that can be used and also creates object menu from the Menu class that is defined in the menu.py. The main program itself starts using the menu objects where the menu takes a certain role. It prompts user to make decisions in the program. Once the program has prompted file name and the operation for it, the main program calls the runOperation method from the menu. The menu object remembers all the decisions made previously and acts upon requests by calling file_handler object. It fills all the necessary parameters, so that the file_handler can do the required operations for the files themselves. If some key aspect is still missing, runOperation acts and prompts for the missing details, before proceeding with the method calls. Once the file_handler object receives the call on some of the methods, it starts to work with the files as required. One aspect in the example is that it acts more like typical object, but file_handlers could work in a static way too. It means, that it doesn't have to maintain many states (property values) in the objects as most of the operations can be completed on their own, based on the method inputs. This static way might be implemented during the lecture.

File operations These examples have few operations:

print content
print content with row numbers
print line
add new line (append)
edit line
delete file

Added functionalities compared to original:

1. Option to delete files
2. The main program works in a loop, so you can continue using it instead of automatically shutting it down