Concepts of C++ Programming
winter term 2022

Technical University of Munich
CIIP Group

Exercise sheet 12 2023-02-02

Due date: 2023-02-09 16:59

Create a program which monitors the creation, modification, and deletion of files within a specified location and record the observed changes into a log file. For this, we will be using multiple features of the standard template library, which we did not use thus far. This includes the `filesystem` and `chrono` libraries, and file input and output streams for the logger.

Exercise 1:

Complete the definition of the `Logger` class. This should be able to create and modify a log file at the specified location

- What standard libraries need to be included for the class to function properly?
- Should a previous log file by the same name exist, then rename it by appending "`_old`" to the end of the filename
- If such a file already exists, then the backup simply gets overwritten
- When or how should the file get closed?

Exercise 2:

Complete the implementation of the `FileMonitor` class. This sets up a logger upon construction, and formally starts checking for additions, modifications, and deletions at a provided `interval`, and until a prespecified `timeout`

- To maintain the state of the observed directory, the monitor keeps a mapping of a path to its last modification time
- Add the class constructor for the monitor, which should initialize the map and the logger
- Add the monitoring function `start`, which consistently checks for updates until the specified timeout

Further exploration:

Extend the defined system to make it even more versatile

- What should be done if we wished to monitor more than one directory...
  - with different log files?
  - within the same log file?
- What needs to be modified if the application should also output to the standard console output?
- Active polling like we do here is often a bad approach. What would the proper approach for an event-based change notification delivery from your operating system?