Computer Science 1st semester
DMU e2016 S

Steffen Balje
sba@easv365.dk

TextFileDupeChecker

TextFileDupeChecker (TFDC) asks the user for a source file, and a list of candidate files to check. For each candidate file, the program should check whether the content is identical to the content in the source file. It should print all duplicates found.

Part 1

Groups of type A create the user interface.

Groups of type B create a method that checks the source file, against candidate files.

Base64 hint: bGlzdDEuZXF1YWxzKGxpc3QyKQ==

Before starting, team up with a group of a different type. A team consists of an A group and a B group. Discuss how the method that the B group is working on, should look like, and come to an agreement. Also discuss what should happen, when a user enters a file, which does not exist. Afterwards, start working in parallel, in the groups.

Part 2

Read https://docs.oracle.com/javase/7/docs/api/java/nio/file/DirectoryStream.html

This is a technical description of how to list all files inside a folder. Although this is the official Java documentation by Oracle, it is often still useful to find alternative sources, to understand the matter at hand. Note: A Path object can be converted to a File object, by calling `.toFile()`

Modify the working program from part 1, so that instead of a list of candidate files, a folder is given. All files within the folder, is then considered to be candidate files, to be checked against the source file.

Part 3

TFDC in its current state, works with text files. It may not work well with other types of files.

To make it work with any file, we can make use of something called hashing. In essence, it create a "fingerprint" for any file. If you change even a single bit in the file, the fingerprint will be completely different. The fingerprint is always the same size, even for file of several gigabytes!

Look up *hashing* for more details.

Computer Science 1st semester
DMU e2016 S

Steffen Balje
sba@easv365.dk

So we can take a single file as input, and generate a string as output. If two files have the same string as output, they are identical. The following method takes a file as input, and outputs a string. Use it to make a new FileDuplicateChecker program.

Steffen Balje
sba@easv365.dk

```
public static String getHash(File file) throws NoSuchAlgorithmException,
IOException {
        MessageDigest md = MessageDigest.getInstance("SHA");

        try (FileInputStream fis = new FileInputStream(file)) {
            byte[] buffer = new byte[4096];
            int read = 0;
            while (read != -1) {
                read = fis.read(buffer);
                if (read > 0) {
                    md.update(buffer, 0, read);
                }
            }
        }

        byte[] result = md.digest();
        StringBuilder sb = new StringBuilder();
        for (byte b : result) {
            sb.append(String.format("%02x", b));
        }

        return sb.toString();

    }
```