# File management

## Accessing files - Additional exercises

> **Exercise 1:**
>
> Create a project named **RegexReader** containing the following things:
>
> - A class called `RegexReader` that inherits from `BufferedReader` (you can take a look at previous example to help you implement this class). It will receive a `Reader` in the constructor and a String containing a regular expression to match when reading lines. You'll have to override the `readLine()` method. It will use the parent's `readLine()` method and return the first line that matches the regular expression received in the constructor (use the `.matches(String)` method from the class String), ignoring all other lines (until it gets null, you'll have to return that too when that happens).
> - In the main method, create a `RegexReader` object that will give you only lines containing a date (*dd/mm/yy* or *dd/mm/yyyy* format), mixed with text. A regular expression in Java always tries to match all the String to be valid, so you'll have to add `.*` at the beginning and at the end of the expression, also `\d` is invalid, you'll have to use `\\d`. Create a file that contains a date in some lines and try it.

> **Exercise 2:**
>
> Create a program called **FolderCopy** that copies all the files from an existing folder into another. The program must previously delete the contents of the destination folder (if any), and we assume that in the source folder there will only be files (not folders)
>
> **NOTE**: `Files.delete` method only works on empty folders, so we need to empty the destination folder before deleting it.

> **Exercise 3:**
>
> Create a program called **MovieManagement** that stores the movies you have already watched. Define a `Movie` class to store movie information (title and genre). Then, the main program will try to load a file called *movies.dat* at the beginning, and show you your current list of movies. Finally, it will ask you if you want to enter more movies in the list. You can enter as many movies as you want, until you type an empty title. Then, the program will finish, writing the updated list in the serialized file.