

# Assignment 1

- create a program, which monitors a folder (that is given as the program argument) and periodically prints out changes in the folder

# Assignment 2

- create a program that works as a simple interactive shell (i.e., a program providing a command line for executing commands) for file manipulations
  - supported commands
    - help – prints out a list of all commands and their help
    - pwd – prints out a full path of the current folder
    - dir – prints out a content of the current folder
    - cd <folder> – changes the current folder
    - cp <file> <path> – copies a file
    - mv <file> <path> – moves a file
    - cat <file> – prints out a textual file
    - ... any other command
  - create a suitable interface to represent a command
  - individual commands are implementations of the interface
    - i.e., it is easy to add new commands

Tests...

# Test

- Which of the following interfaces are functional (i.e., they can be annotated with `@FunctionalInterface` and the compiler compiles them)?



*Exam test*

- A) 

```
public interface AnInterface {
    int foo(int a, int b);
}
```
- B) 

```
public interface AnInterface2 extends AnInterface {
    int foo(double a, double b);
}
```
- C) 

```
public interface AnInterface3 {
}
```
- D) 

```
public interface AnInterface4<E> {
    int foo(E a, E b);
    default InInterface4<E> bar() {
    }
}
```



Slides version PJ07plus.en.2020.01

This slides are licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).