

Exercise 1

- Update the Shell from the third practicals (commands loaded via the ServiceLoader) to be a modular application
 - commands still loaded via the ServiceLoader
 - i.e. create a module from the Shell core
 - contains the Command interface and Main
 - the interface in an exported package, Main in an un-exported package
 - in the module-java.info, declares usage of Command interface implementations
 - a module for each command
 - in the module-java.info, declares the implementation of the Command interface

Assignment 1

- help – how to create a modular jar

Name of JAR

```
jar -c --file=shell-cmd-hello.jar  
-C out/production/cz.cuni.mff.java.shell.commands.hello/ .
```

A directory with
class files

A directory to be packed
to JAR, relatively to the
previous directory

Example 2

- Create an HTTP server
 - works with a regular browser
 - understands the HTTP 1.0 protocol
 - request

```
GET /stranka.html HTTP/1.0
Host: www.web.org
*other headers*
*empty line*
```
 - response

```
HTTP/1.0 200 OK
*empty line*
*body of the answer*
```
 - server content – a directory set as a parameter at launching

Example 3

- To the HTTP server, add support for dynamic content
 - in addition to static pages, the server can launch scripts and returns their results
 - scripts – via Scripting API
 - all, that can be found through Scripting API
 - in addition to the default JS, download another engine
 - the server decides based on the file extension, whether to return it or launch as a script
 - try passing variables from Java to a script
 - e.g. the required URL



Slides version PAJ04.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/).