Distributed Systems
Exercise Sheet 7, Monday, 11:45
Klingemann, SS 2022

Deadline: 4th July 2022

4th Assessed Exercise

1. Tomcat configuration

Note, that the following steps assume that you manually start Tomcat from a command prompt using Windows.

1. Download Tomcat as a zip-file and extract it.
2. Set the environment variable `JAVA_HOME` to the directory in which the Java SDK is installed, e.g. `C:\Program Files\Java\jdk1.8.0_144`.
3. Modify the file `<your tomcat directory>\conf\context.xml`. Change the line
   `<Context>`
   to
   `<Context reloadable="true">`
   Thus, we have to add the corresponding attribute.
4. Start Tomcat. To do so, you can start a command prompt and call `<your tomcat directory>\bin\startup.bat`. To stop Tomcat, you can call `shutdown.bat`.
5. Test your installation. Tomcat uses by default the port 8080. Therefore, you can use in your Web-browser the URL `http://localhost:8080`. When you can see the welcome-page of Tomcat, you have successfully started Tomcat.

2. Installation of Servlets

To compile a servlet, the library
`<your tomcat directory>\lib\servlet-api.jar`
has to be contained in the environment variable `CLASSPATH`. If you compile and execute your programs from a command prompt, you can achieve this by executing `cp1.bat` (you have to adjust the path-information so that it fits to your computer) or you can set this variable in the Windows control panel. If you are using an IDE like Eclipse you have to configure it appropriately.
Create in the directory `webapps`
a directory for your Web-application, e.g. `webapps\a6`.
Create in `a6` a directory `WEB-INF` and in `WEB-INF` a directory `classes`.
Copy your compiled classes into the directory
`webapps\a6\WEB-INF\classes`.
HTML-files are put into the directory `webapps\a6`.
Your HTML-file (for example `ask_name.html`) can be called within the browser using the URL
`http://localhost:8080/a6/ask_name.html`.
If you create a servlet, add before the definition of the class an annotation with the following syntax:
`@WebServlet("/Hallo")`
This allows you to call your servlet within the browser with the URL
`http://localhost:8080/a6/Hallo`.

3. Web-client for your system for the management of shopping baskets

Create based on Servlets (not JSP!) a Web-client for your system for the management of shopping baskets from Sheet 3. You should present to the user different Web-pages in a process that consists of three steps. You can assume that there exists exactly one shopping basket-object on the server and the functions offered by the server operate on this object.

1. Selecting the task. The user can choose among
   - Search for a shopping item with a particular name.
   - Add a new shopping item.
   - Show all shopping item-objects of the shopping basket.
   - Change the quantity of a shopping item.
2. If the chosen task requires input, the user is presented another Web-page that allows entering the necessary data.
3. Afterwards the user is presented yet another Web-page to display the result. In addition to this, this Web-page also allows selecting a task like in step 1 and therefore, the process is started again.

Organizational issues

- You have to solve the exercise completely on your own! (No working in groups!)
- It is necessary but not sufficient to present a working program. Moreover, you have to be able to explain all parts of your program, be able to answer questions with respect to your program and make small extensions of you program.
- Your program has to be created completely within the exercise slot.
- If you violate one of the rules above, this implies that you definitely fail in this exercise.
- You can only present solutions that correspond to the exercise slot you are assigned to.
- It is in your responsibility to present your solution in time before the deadline. The assessment of your solution can only be guaranteed if you finish your program 60 minutes before the end of the exercises.
- To take part in the exam it is required to solve at least three of five assessed exercise sheets.