Distributed Programming II

A.Y. 2014/15

Assignment n. 1 – part a)

The material for this assignment is in the *.zip* archive where you have found this file. Extract the archive to an empty directory that you will use as your working area and that we will call `[root].`

DP2-FDS is a (simplified) distributed system that manages flight information in an airport.

The Java interfaces defined in the package `it.polito.dp2.FDS` (available under folder `doc` and at https://pad.polito.it:8080/enginframe/dp2/assignments/lab1/doc/index.html) give read-only access to some information about aircrafts, flights, and flight instances. The javadoc of the interfaces documents the kind of information that can be retrieved. The main interface, from which all the information can be accessed, is interface `FlightMonitor`. The methods in this interface can be used to get the available information about known aircrafts (`getAircrafts`), flights (`getFlights`) and flight instances (`getFlightInstances`). The latter two methods give lists of `FlightReader` and `FlightInstanceReader` interfaces, by which information about flights and flight instances can be obtained. From a `FlightInstanceReader` interface it is possible in particular to get a list of passengers registered for that flight instance. The information about each passenger is provided by a `PassengerReader` interface.

The solution to part a) will be submitted along with the solution to part b).

Assignment description

1. Design an *XML* application that can be used to store all the information that can be retrieved using the *Java* interfaces defined in the package `it.polito.dp2.FDS`. The XML format must be such that all the data that can be retrieved using the interface can also be obtained from the *XML* document, and redundancies should be avoided. The *XML* application must be specified using a *DTD*, which must be saved in the file `[root]/dtd/flightInfo.dtd`.
2. Write a short documentation of your design choices about the DTD (max 1 page) and save it as an ASCII file in `[root]/dtd/doc.txt`.
3. Write a valid *XML* file that references the designed *DTD locally*. The file must be saved as `[root]/dtd/flightInfo.xml`.

Correctness verification

Before submitting your files, please verify their correctness. The submitted solution must at least satisfy the following requirements, in order to be considered acceptable:

- the file `flightInfo.dtd` must be syntactically correct;
- the file `flightInfo.xml` must be valid and must reference the DTD stored in the same directory in `flightInfo.dtd`;
- the file `doc.txt` must exist.

The validity of the XML file can be checked by any XML validation program. For example, it can be checked by the Eclipse validate command or by running the `DomParseV` program (distributed in the *XML* examples bundle on the course site).