Datenbanken und Web-Techniken
Exercise 4:
XML Access Techniques 2: DOM

DOM
Document Object Model

- Specification of an API to access XML and HTML documents
- Features:
  - well-defined object model
  - creates document object for parsed document
    - high memory usage
  - document elements consist of nodes with several types: document, element, attribute, text, ...
  - nodes are in relationship with each other: parent, child, sibling
- Usage:
  - change document structure or content
  - document traversal (for example to search something or get a special order of content)
  - small XML files

Tasks
Hints and more description follow on the following pages

DOM Test Task:

1. Load the provided DOMTest-Project in your IDE
2. Modify DOMTest.java to write pizza information from pizzenExample.xml to console log

DOM Usage Task:

3. Load the provided PizzaDOM-Project in your IDE
4. Modify the project to load the pizza data from pizzenExample.xml by using DOM
5. Modify the project to save the order as XML file

DOM Test Task 1: Load the provided DOMTest-Project in your IDE

- If you don’t have a running NetBeans IDE, follow the Preparation Task 2 from exercise 2
- Download and extract the provided **`DOMTest.zip`**
- Open your NetBeans IDE
- Select **`File → Open Project…`** and choose the **`DOMTest`** folder
  - there should be a different folder icon (brown coffee pot icon or NetBeans project icon)

DOM Test Task 2: Modify DOMTest.java to write pizza information from pizzenExample.xml to console log

- Open the file **`DOMTest.java`** in the editor
- Read through the code (and the comments) and try to understand, what is done
- Modify the **`parseXMLList()`** to print the attributes of all pizzas from **`pizzenExample.xml`**
  - XML file is located in **`resources`** folder
- Compile and run your project and check the console log
  - take a look at the XML file to verify your output

DOM Usage Task 3: Load the provided PizzaDOM-Project in your IDE

- Download and extract the provided **`PizzaDOM.zip`**
- Open the **`PizzaDOM`** project in your NetBeans IDE
  - there should be a different folder icon (blue globe icon or NetBeans project icon)
  - if the project is not recognized, go back to Preparation Task 2 from exercise 2 and activate Java EE in NetBeans
- The project is basically the same like PizzaJDBC or PizzaSAX task from exercises 2 and 3

DOM Usage Task 4: Modify the project to load the pizza data from pizzenExample.xml by using DOM

- Open the file **`PizzaList.java`** in the editor
- Go to method **`parseXMLList()`**
- Add some code, to read the pizza data from the XML file using DOM and fill the **`pizzas`** object
- Compile and run your project and check the result at the **`Bestellen`** page

DOM Usage Task 5: Modify the project to save the order as XML file

- Open the file **`Order.java`** in the editor
- Go to method **`saveOrder()`**
- Add some code, to build up a DOM representation of the submitted order
  - just follow the **`@TODO`** sections and implement things as requested
- Compile and run your project and submit an order at the **`Bestellen`** page
- Check the result in your **`orders.xml`** (path will be shown on last page)
  - make sure to look at the correct place, as there are two **`orders.xml`** in your project tree
    - one file is the source file and your order will be added to this file and saved to the other file
      - that's why there will only be your latest order in the result file
        - you might change this, so that source and destination are the same file
          - if your code has some errors and your XML file gets corrupted, your project will not work any longer
            - use the original **`orders.xml`** to repair (and correct your code)