

Programmieren 3 – Grundlagen der Webprogrammierung Assignments

Bei allen **Assignments** handelt es sich um **Prüfungsleistungen**!

Die Lösungen sind als in Eclipse bzw. IntelliJ importierbares (Java-)Projekt spätestens bis zum jeweils angegebenen Abgabedatum, 24.00 Uhr auf den Git-Server des Labors für Multimedia hochzuladen und in der ersten Übung nach dem Abgabedatum kurz zu präsentieren! Verwenden Sie für jedes Assignment das entsprechende bereits auf dem Git-Server vorhandene, leere Repository und legen Ihre Abgaben jeweils im `master`-Branch ab! Zusätzlich ist dort eine Textdatei `members.txt` abzulegen, die die Namen und die Matrikelnummern der an der Abgabe beteiligten Personen enthält. Verwenden Sie für jedes Assignment höchstens einen Java-Source-Folder. Repositories, die eine Struktur enthalten, die sich nicht ohne Änderung in Eclipse/IntelliJ importieren lässt, werden nicht berücksichtigt!

Der Git-Server ist über die IP-Adresse `172.19.49.46` innerhalb des Hochschulnetzes direkt und von außerhalb über VPN erreichbar. Die Accounts haben die Form `WPWiSe23xx`; die Anmeldedaten werden den Gruppen auf Anforderung per E-Mail zugeschickt.

Das direkte Einloggen auf dem Server ist z. B. durch `ssh WPWiSe23xx@172.19.49.46` möglich; durch `git clone WPWiSe23xx@172.19.49.46:git/assignment01.git` kann z. B. das Git-Repository für das erste Assignment geklont werden.

Assignment 1

Abgabedatum: 4. Dezember 2023

Arbeitsgruppen: max. 4-er-Gruppen

Ausgangssituation

Es ist geplant, den Bestand der Informatik-Rechnerlabore der Hochschule in einem XML-Dokument zu erfassen. Damit das Dokument auch validiert werden kann, muss ein entsprechendes XML-Schema erstellt werden.

In dem Schema sind folgende Sachverhalte abzubilden:

- Für jedes Labor sind die Rechner, die vorhandene Software und eventuell vorhandene weitere Geräte (z. B. Drucker oder Scanner) zu erfassen (dabei soll pro Labor mindestens ein Rechner vorkommen)
- zu jedem Labor-Element ist außerdem als Attribut dessen Name (z. B. „Multimedialabor“) und die Raumnummer (z. B. „Z2320“) aufzunehmen
- jeder Rechner im Labor ist durch einen eindeutigen Schlüssel identifizierbar; außerdem werden für jeden Rechner folgende Angaben erfasst:
 - installiertes Betriebssystem
 - Anschaffungsdatum des Rechners

- eventuell vorhandene Erweiterungen des Rechners bestehend aus einer Beschreibung der Erweiterung (zusätzliche Festplatte, weiterer Speicher oder ähnliches) und optionalen Besonderheiten der Erweiterung
 - Angaben über eventuell außer dem Betriebssystem auf dem Rechner installierte Anwendungssoftware; da identische Software auf mehreren Rechnern installiert sein kann, wird hier nur auf den Datensatz der an anderer Stelle im Dokument erfassten Software verwiesen, ohne die Details der Software bei jedem Rechner wiederholt zu notieren. Allerdings ist für jede auf dem Rechner installierte Software anzugeben, ob die Software auf dem Rechner lizenziert ist oder nicht (es können für jede Software mehr Installationen als Lizenzen vorhanden sein); falls nicht anders angegeben, wird natürlich angenommen, dass die Softwareinstallation lizenziert ist
 - eine kurze Beschreibung des Rechners
 - eventuelle Besonderheiten des Rechners (z. B. wenn das Mainboard bereits mehrfach getauscht wurde oder das Gehäuse stark beschädigt ist)
- für jede im Labor vorhandene Software ist eine eindeutige Bezeichnung (Schlüssel) festzulegen; außerdem ist für jede Software eine kurze Beschreibung zu erfassen (z. B. „Textverarbeitung“ oder „IDE“ etc.) und die Möglichkeit zum Notieren von Besonderheiten der Software vorzusehen
 - für jede bei Geräten, Software oder Erweiterungen notierte Beschreibung gilt, dass diese aus einer einfachen textuellen Beschreibung besteht. Um jedoch sämtliche im Labor vertretenen Herstellerfirmen einfach ausfindig machen zu können, sind alle in den Beschreibungstexten auftretenden Herstellernamen durch eine spezielle Auszeichnung hervorzuheben (z.B. `<beschreibung><hersteller>Lenovo</hersteller>-PC</beschreibung>` oder `<beschreibung>25 TByte SSD von <hersteller>Seagate</hersteller></beschreibung>`)

Aufgabe

Entwerfen Sie für den in der Ausgangssituation geschilderten Sachverhalt ein XML-Schema. Entscheiden Sie dabei selbst, welche Informationen als Elemente und welche als Attribute erfasst werden. Achten Sie insbesondere darauf, welche Daten verbindlich bzw. optional sind und für welche Angaben Standardwerte nützlich sind.

Notieren Sie beispielhaft für (fiktive) Labore die entsprechende Ausstattung in einem XML-Dokument. Selbstverständlich muss das XML-Dokument wohlgeformt und bezüglich Ihres Schemas gültig sein. Verwenden Sie für Ihre XML-Elemente den Namensraum `assignment01.webprog.de`.

Damit die Daten auch in lesbarer, gut strukturierter Form präsentiert werden können, entwickeln Sie zu Ihrem Dokument ein XSLT-Stylesheet, das direkt im Browser¹ eine XHTML-Ausgabe der Labordaten produziert (achten Sie dabei auf eine geeignete XSLT-Version). Im Stylesheet setzen Sie die rekursive Anwendung von Templates ein!

Auf der im Browser generierten XHTML-Seite sind neben den Labordaten auch ein vom Stylesheet erzeugtes Inhaltsverzeichnis mit Links auf die Daten der einzelnen Labore, sowie mit Links auf eine zu erzeugende Softwareliste, eine Liste sonstiger Geräte und eine Liste sämtlicher in den Laboren anzutreffender Hersteller darzustellen. Die Einträge in den Listen sind selbstverständlich alphabetisch sortiert und enthalten keine Duplikate. Ergänzen Sie schließlich die XHTML-Seite noch mit einer Tabelle, in der (mindestens) die Anzahl der Labore, der Rechner, der Softwareprodukte und die durchschnittlichen Rechneranzahl pro Labor aufgeführt sind.

¹ Verwenden Sie eine aktuelle Version aus der Mozilla- oder Firefox-Serie.

Kommentieren Sie das XML-Dokument, das Schema und das Stylesheet!

Zusätzliche Hinweise

- a) Die Verarbeitungs des Stylesheets geschieht ausschließlich im Browser. Es ist keine explizite XHTML- oder HTML-Datei zu erzeugen.
- b) Die meisten Browser besitzen die Einschränkung, dass Dateien nur über gesicherte Quellen nachgeladen werden dürfen. Insbesondere wird z. B. das Nachladen von XSLT-Dateien aus einer über das Protokoll `file://` geladenen XML-Datei von neuen Browserversionen in der Regel als unsicher angesehen und daher ignoriert.

Abhilfe schafft hier nur das Ausliefern mittels `http://`- bzw. `https://`-Protokoll über einen Webserver. Die XML- und XSLT-Dateien können Sie dazu z. B. in das bei den Accounts WPWiSe23xx bereits vorhandene web-Verzeichnis kopieren.

Eine dort eingestellte Datei `foo.xml` ist dann z. B. über den URL `https://mulaweb.hs-bremerhaven.de/WPWiSe23xx/foo.xml` abrufbar.

Assignment 2

Abgabedatum: 15. Januar 2024

Arbeitsgruppen: max. 4-er-Gruppen

Aufgabe

Nachrichten aus einem „RSS-Feed“ sollen mithilfe eines in Java zu erstellenden „Filterprogramms“ nach bestimmten Schlüsselwörtern durchsucht und die Treffer in einer XML-Datei gespeichert werden.²

Genauer soll das Filterprogramm aus dem Feed genau die `item`-Elemente herausfiltern, in deren `description` eines der Wörter „Android“, „Java“, „Welt“, „Mensch“ oder „Berlin“ auftritt. Dabei gelten auch Wortbestandteile wie z. B. „Menschheit“ oder „Umwelt“ als Treffer.

Welcher Feed zu verarbeiten ist, wird dem Filterprogramm als Laufzeitparameter übergeben. Beispielsweise können die Feeds der [Tagesschau](#), des Nachrichtenmagazins [Der Spiegel](#) oder von [heise online News](#) untersucht werden.

Es ist ein Filterprogramm `RssFilterStax` mit den oben beschriebenen Eigenschaften zu erstellen, das zur Verarbeitung des Feeds die Iterator-API von StAX und zum Speichern der Treffer JAXB verwendet.

In `RssFilterStax` ist für jeden Treffer der vollständige Text der `description` zusammen mit anderen „wichtigen“ Angaben zu dem zugehörigen `item`-Element in ein neues XML-Dokument zu übernehmen. Für dieses „Treffer-Dokument“ ist ein geeignetes XML-Schema zu erstellen. Als Namensraum wird im Treffer-Dokument die Bezeichnung `assignment02.webprog.de` festgelegt. `RssFilterStax` verwendet den Namensraum als explizit benannten Namensraum mit dem Präfix `stax`.

Der Zugriff auf das Treffer-Dokument erfolgt innerhalb des Filterprogramms mithilfe von JAXB. Daher sind unter Verwendung des JAXB-Binding Compilers `xjc` die zu Ihrem Schema passenden Datenklassen zu generieren. Jeder Aufruf eines Filterprogramms löscht alle bisherigen Einträge im Treffer-Dokument und nimmt nur die aktuell gefundenen Treffer in das Dokument auf. Beim Speichern der Daten in dem Trefferdokument ist eine Validierung gegen das Schema durchzuführen.

`RssFilterStax` ist in Java unter Verwendung der Java Version 21 zu erstellen. JAXB ist in der Version 4.0.x zu verwenden. Der Einsatz zusätzlicher Bibliotheken ist nicht erlaubt.

Um die Ausführung des Filterprogramms zu vereinfachen, ist dafür eine eigene `jar`-Datei `RssFilterStax.jar` zu erstellen, die sämtliche Abhängigkeiten enthält und direkt ausgeführt werden kann; z. B. soll die Verarbeitung des Tagesschau-Feeds angestoßen werden können durch

```
java -jar RssFilterStax.jar https://www.tagesschau.de/xml/rss2
```

Eine verständliche Kommentierung aller relevanten Teile Ihrer Lösung wird vorausgesetzt.

² Zu Informationen, wie ein gültiges RSS-Dokument aussehen kann, siehe z. B. www.rssboard.org; dort ist auch ein [RSS Validator](#) zum Validieren von `rss`-Dateien verlinkt.

Tipp

Damit Sie beim Implementieren und Testen unabhängig von den sich ständig ändernden Live-Feeds sind, empfiehlt es sich, auf der Basis eines Feeds geeignete Testdaten anzulegen. Beachten Sie, dass innerhalb einer `description` auch Entities (z.B. `";`, `'` oder `>`;) vorkommen können!

Hinweise zum Arbeiten mit JAXB

Da das benötigte Modul `jakarta.xml.bind` nicht in der Java Standard Edition enthalten ist, müssen die für das Arbeiten mit JAXB benötigten Pakete teilweise separat installiert werden.

xjc-Binding Compiler

Der `xjc`-Binding Compiler kann als sogenanntes „JAXB RI standalone zipped bundle“ über den folgenden Link <https://repo1.maven.org/maven2/com/sun/xml/bind/jaxb-ri/4.0.4/jaxb-ri-4.0.4.zip> bezogen werden.

In der IntelliJ Ultimate-Version ist das für den Binding Compiler relevante Plugin „Jakarta EE: Web Services (JAX-WS)“ schon vorinstalliert; beim Aufruf bitte die voreingestellte Versionsnummer beachten!

Setzen Sie beim Aufruf des Binding Compilers den Schalter `-mark-generated`, damit im generierten Quelltext eine `@Generated`-Annotation aufgenommen wird. Dieses gilt sowohl für den Aufruf aus der Kommandozeile als auch für das Starten über das IntelliJ-Plugin.

Installation der für JAXB benötigten jar-Dateien

Unter Eclipse oder IntelliJ gelingt die Installation der zum Übersetzen und Ausführen benötigten jar-Dateien am einfachsten, indem das Java-Projekt in ein Maven-Projekt umgewandelt bzw. direkt als Maven-Projekt angelegt wird, und die folgenden Abhängigkeiten in `pom.xml` hinzugefügt werden:

```
<!-- https://mvnrepository.com/artifact/jakarta.xml.bind/
      jakarta.xml.bind-api -->
<dependency>
<groupId>jakarta.xml.bind</groupId>
<artifactId>jakarta.xml.bind-api</artifactId>
<version>4.0.1</version>
</dependency>

<!-- https://mvnrepository.com/artifact/com.sun.activation/
      jakarta.activation -->
<dependency>
<groupId>com.sun.activation</groupId>
<artifactId>jakarta.activation</artifactId>
<version>2.0.1</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.glassfish.jaxb/
      jaxb-runtime -->
<dependency>
```

```

<groupId>org.glassfish.jaxb</groupId>
<artifactId>jaxb-runtime</artifactId>
<version>4.0.4</version>
</dependency>

<!-- https://mvnrepository.com/artifact/jakarta.annotation/
      jakarta.annotation-api -->
<dependency>
<groupId>jakarta.annotation</groupId>
<artifactId>jakarta.annotation-api</artifactId>
<version>2.1.1</version>
</dependency>

```

Alternativ können die entsprechenden jar-Dateien vom Maven-Repository auch direkt bezogen und manuell installiert werden (URLs siehe oben in den Kommentaren vor den Maven-Abhängigkeiten).

Explizit benannte Namensräume in JAXB

Um im Ausgabedokument einen explizit benannten Namensraum zu verwenden, kann direkt im Marshaller über die Property `org.glassfish.jaxb.namespacePrefixMapper` der entsprechende Präfix gesetzt werden.

So wird beispielsweise durch die folgende Anweisung in der Ausgabe der Präfix `foo` verwendet:

```

marshaller.setProperty("org.glassfish.jaxb.namespacePrefixMapper",
    new NamespacePrefixMapper() {
        @Override
        public String getPreferredPrefix(String namespaceUri,
            String suggestion, boolean requirePrefix) {
            return "foo";
        }
    }
);

```

Assignment 3

Abgabedatum: 29. Januar 2024

Arbeitsgruppen: max. 4-er-Gruppen

Ausgangssituation

Gegeben sei eine beliebige XHTML-Seite, die unter anderem mehrere Instanzen der Elemente `h1`, `h2`, `img`, `p` und `a` enthält. Außerdem sollen Listen mit mehreren `li`-Instanzen und Tabellen mit `tr`-Instanzen vorhanden sein. Tabellen und Listen können auch geschachtelt sein.

Ziel ist es, diese Art von XHTML-Seiten auf Wunsch derart zu „modifizieren“ (oder zu verunstalten), dass Instanzen desselben Typs auf der Seite untereinander vertauscht – also neu gemischt – werden. Beispielsweise können alle vorhandenen `h1`-Überschriften, alle `img`-Bilder oder die Zeilen `tr` von allen Tabellen neu positioniert werden.

Beachte: Das Mischen bzw. Vertauschen der Knoten ist so zu verstehen, dass nicht die Inhalte der Knoten kopiert oder neu angelegt, sondern nur die Verknüpfungen der bestehenden Knoten untereinander geändert werden!

Aufgabe

Entwerfen Sie zunächst in XHTML eine einfache Beispielseite, die mindestens die oben aufgezählten Sprachelemente enthält.

Erstellen Sie in einer separaten Datei JavaScript-Code, der in solche Seiten eingefügt werden kann und die beschriebene „Modifikation“ unter Verwendung der DOM-API und von DOM Level 2-Eventhandlern durchführt.

Dabei erfüllt Ihr JavaScript-Code insbesondere folgende Aufgaben:

- a) Nach dem Laden der Seite in einem Browser³, wird z. B. unten auf derselben Seite oder in einem neu zu öffnenden Browserfenster ein „Steuerabschnitt“ eingefügt.

Im Steuerabschnitt ist für die oben angegebenen Sprachelemente jeweils die Anzahl der in der konkreten XHTML-Seite vorhandenen Instanzen angegeben.

- b) Nach Auswahl eines der oben angegebenen Sprachelemente kann z. B. durch „Anklicken“ eines Buttons das Mischen der entsprechenden Instanzen angestoßen werden.

Das Mischen wird durch mehrere, für den Benutzer sichtbare, hintereinander ausgeführte paarweise Vertauschungen von jeweils zwei zufällig ausgewählten Instanzen realisiert.⁴

Beachten Sie, dass Instanzen desselben Elementtyps teilweise rekursiv verschachtelt sein können!

Sie können auch weitere XHTML-Sprachelemente zum Mischen vorsehen oder zusätzlich zum Mischen eigene Modifikationen implementieren.

³ Entsprechend den Vorgaben aus der Vorlesung werden nur neuere Browser der Mozilla- oder Firefox-Serie verwendet.

⁴ **Tipp:** Um die einzelnen paarweisen Vertauschungen für den Benutzer sichtbar zu machen, empfiehlt es sich, die einzelnen Iterationen etwas verzögert auszuführen. Das verzögerte Ausführen von Methodenaufrufen kann z. B. unter Verwendung von `window.setInterval()` oder `window.setTimeout()` erreicht werden.

Zusatzaufgabe

Es sind weitere Funktionalitäten zu implementieren, die bei der Darstellung der Beispielseite auf Mobilgeräten zur Anwendung kommen:⁵

- Zusätzlich zum „Anklicken“ können die zu mischenden Sprachelemente durch „Drag and Drop“ innerhalb des Steuerabschnitts ausgewählt werden.
- Das Mischen kann unter Auswerten des Beschleunigungs- oder des Neigungssensors durch „Schütteln“ des Gerätes gestartet werden.
- Es sind zwei unterschiedliche, sinnvolle Hintergrundfarben A und B zu definieren, die abhängig von der aktuellen Ausrichtung des Gerätes verwendet werden: Ist das Gerät in Richtung „Norden“ (bzw. 0) ausgerichtet, wird Farbe A angewendet; bei einer Ausrichtung in Richtung „Süden“ (bzw. 180) kommt Farbe B zur Anwendung. Liegt die Ausrichtung zwischen diesen beiden Extremwerten, wird entsprechend zwischen den Farben A und B linear interpoliert.

⁵ Durch das Bearbeiten der Zusatzaufgabe wird entweder ein maximaler Punktebonus in Höhe von 20 % gewährt oder im Falle der Anrechnung dieses Kurses als Wahlpflichtfach werden die anzurechnenden Credit Points auf 6 CP erhöht.