

## Aufgaben zur Lehrveranstaltung Laborpraktikum Software SMIB , SMSB

### D. Aufgabenblatt

- D.1 Präsenzaufgabe (XML parsen, SAXParser)
- D.2 Präsenzaufgabe (RSS-Abfrage, URL, URLConnection)
- D.3 Hausaufgabe (RSS, XML parsen )
- D.4 Hausaufgabe (XML, mit String-Operationen parsen)
- D.5 Hausaufgabe (XML aus dem Netz abfragen und parsen)
- D.6 Hausaufgabe (XML, Datumsformate und Zeitzonen)
- D.7 Hausaufgabe (XML Parsen, String-Operationen)
- D.8 Hausaufgabe - >Bonus (Persistenz)

Die Präsenzaufgaben dieses Aufgabenblattes werden im Labor ab November bearbeitet. Die Hausaufgaben dieses Aufgabenblattes sind spätestens **bis spätestens 08:00 Uhr am 02.12.2019 abzugeben**. Verspätete Abgaben werden nicht berücksichtigt.

### D. Aufgabenblatt

#### D.1 Präsenzaufgabe (XML parsen, SAXParser)

Erweitern Sie das in der Vorlesung genutzte Beispiel zum SAXParser (Personen) um

1. Die Möglichkeit Hobby, Lieblingsgericht und Lieblingsband zu parsen und in die interne Datenstruktur zu schreiben. Dazu ist das XML-Format geeignet zu erweitern, die interne Datenstruktur anzupassen und der Handler zu erweitern.
2. Eingabe neuer Personen via Console.
3. Ausgabe einer XML Struktur die die intern gehaltenen Daten reflektiert.  
Tipp: Hierfür benötigen Sie keinen XML-Writer

#### D.2 Präsenzaufgabe (RSS-Abfrage, URL, URLConnection)

Laden Sie aus dem Internet den RSS-Feed <http://www.tagesschau.de/xml/rss2> und geben Sie ihn auf der Console aus.

Verwenden Sie dazu die in der Vorlesung gezeigten Klassen und Methoden.

### D.3 Hausaufgabe (RSS, XML parsen )

[1,5 Punkte]

Verarbeiten Sie den in Aufgabe D.2 geladenen RSS-Feed automatisch wie folgt mit einem SAXParser.

Die baumartige Struktur der zurück erhaltenen XML-Datei ist wie folgt:

- Ein `rss`-Element
  - Ein `channel`-Element
    - Ein `title`-Element
    - Ein `link`-Element
    - Ein `description`-Element
    - Diverse weitere optionale Elemente (`language`, `copyright`, etc.)
    - Ein oder mehrere `item`-Elemente mit jeweils folgenden Kindelementen:
      - Ein `title`-Element
      - Ein `link`-Element
      - Weitere Elemente
      -

Geben Sie für alle `item`-Elemente jeweils den Titel des Items auf der Console aus. Das Ergebnis sieht dann etwa so aus wie in Abbildung 1 (Achtung anderer Feed).

Für Interessierte: unter <http://blogs.law.harvard.edu/tech/rss> gibt es weiterführende Dokumentation zum RSS 2.0 Format.

```
Schlusslicht: Mathe tut weh
Chinas KP-Nachwuchs: Ein Parteibuch für die Karriere
Hintergrund: Die Beschlüsse des Koalitionsgipfels
Betreuungsgeld kommt, Aus für Praxisgebühr
Obama - nur ein Durchschnittspräsident?
Zweiter "Vatileaks"-Prozess im Vatikan
Koalition beschließt Betreuungsgeld und Aus für Praxisgebühr
Stadtwerke Bochum lenken im Streit mit Steinbrück ein
Videoblog: Eine Stadt als Laufsteg
Gipfel im Kanzleramt: Schwarz-Gelb verhandelt über Streitthemen
Kuwait: Erneut Tränengas gegen Demonstranten
Urwahl des Spitzenduos: Grüne beginnen mit Auszählung
Bundesliga: Leverkusen zittert sich gegen Düsseldorf zum Sieg
FDP-Fraktionschef Brüderle: "Wir müssen zeigen, dass wir handlungsfähig sind"
Bundesliga: Leverkusen zittert sich gegen Düsseldorf zum Sieg
NSU-Jahrestag: Nur kleinere Demonstrationen in mehreren Städten
Interview zur US-Wahl: "Dieses Land ist tief gespalten"
Formel 1: Räikkönen siegt - Vettel holt furios auf
Boot verunglückt - 70 Flüchtlinge aus Mittelmeer gerettet
Mehrheit der Israelis würde Romney zum US-Präsidenten wählen
```

Abbildung 1: Als Beispiel: Titel von ARD-RSS-Items

## D.4 Hausaufgabe (XML, mit String-Operationen parsen)

[2 Punkte]

Verwenden Sie geeignete Methoden wie bspw. „indexOf“ und „substring“ sowie ggf. weitere Methoden der Klasse `String` um aus dem `String`

```
<team>
  <person><vorname>Norrin</vorname><nachname>Radd</nachname><alias>Silver
  Surfer</alias></person>
  <person><vorname>Ben</vorname><nachname>Grimm</nachname><alias>Das
  Ding</alias></person></team>
```

die folgende Ausgabe zu produzieren:

```
Norrin Radd - Silver Surfer
Ben Grimm - Das Ding
```

Das Programm sollte so geschrieben sein, dass man die Tags auch **leicht** umbenennen kann. Weiterhin soll die Reihenfolge der Elemente „alias“, „vorname“ und „nachname“ innerhalb des „person“-Elementes beliebig sein und zum gleichen Ergebnis führen. Die Reihenfolge der Personen innerhalb des Teams soll hingegen relevant für die Ausgabereihenfolge sein.

Z.B. soll folgender `String`

```
<team><person><nachname>Radd</nachname> <vorname>Norrin</vorname><alias>Silver
Surfer</alias></person> <person><vorname>Ben</vorname><alias>Das
Ding</alias><nachname>Grimm</nachname></person></team>
```

zu folgendem Ergebnis führen (ggf. nach kurzer Programmänderung):

```
Norrin Radd - Silver Surfer
Ben Grimm - Das Ding
```

Dagegen sollte bspw. der `String` (ggf. nach kurzer Programmänderung):

```
<team><person><nachname>Radd</nachname><alias>Silver Surfer</alias>
<vorname>Norrin</vorname></person>
<mensch><vorname>Ben</vorname><nachname>Grimm</nachname><alias>Das
Ding</alias></person></team>
```

zu einer Fehlermeldung führen.

Die Verwendung eines vorgefertigten Java-XML-Parsers ist in dieser Aufgabe **nicht** erlaubt. Bitte schreiben Sie ein **robustes**, **wartbares** Programm. Bedenken und testen Sie u. a. die folgenden Fälle: Vorname oder Nachname fehlt oder doppelt vorhanden, `team`-tag, `person`-tag fehlt, mehrere `team`-tags, etc.

## D.5 Hausaufgabe (XML aus dem Netz abfragen und parsen)

[2 Punkte]

Diese und die folgenden beiden Aufgaben werden uns einige Wochen begleiten. Bitte bearbeiten Sie die Aufgabe, da Ihnen durch einen späteren Quereinstieg einige Zusatzaufwände entstehen werden. Zusätzlich nutzen wir das in den vorherigen Wochen entstandene Bibliothekssystem (Zettelkasten). Sollten Sie die entsprechenden Aufgaben nicht bearbeitet haben dürfen Sie das System eines Kommilitonen als Ausgangsbasis nutzen.

## Motivation und Überblick über das Projekt

„Wortschatz“ ist ein Projekt der Universität Leipzig, welches unter anderem einen Web-Service für die Suche nach Synonymen anbietet ([wortschatz.uni-leipzig.de](http://wortschatz.uni-leipzig.de)),

„WikiBooks“ ([de.wikibooks.org](http://de.wikibooks.org)) ist eine Bibliothek mit freien Lehr-, Sach- und Fachbüchern.

Aufgabe Ihres Projektes soll es sein, diese beiden Nachschlagewerke, zusammen mit dem Zettelkasten aus Aufgabenblatt C, in einem gemeinsamen Programm zu vereinen.

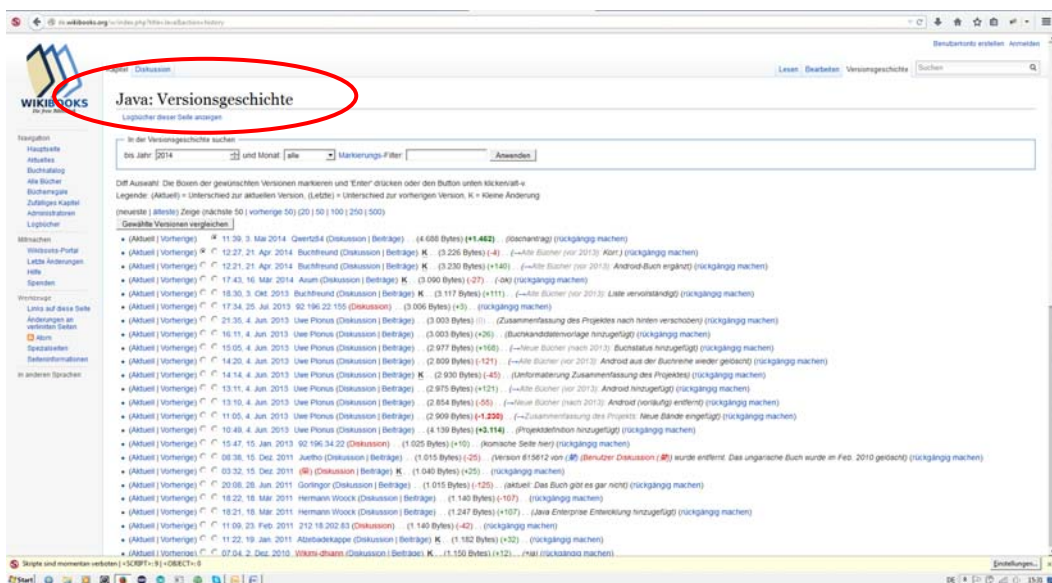
Ihr Auftraggeber, der dieses Projekt initiiert hat, schlägt vor das bereits erstellte Bibliotheksprogramm (Zettelkasten) zu erweitern. Das Programm soll es erlauben, zusätzlich zur Suche im Zettelkasten auch auf WikiBooks Bücher anhand eines Schlagwortes zu suchen. Was ihm zusätzlich fehlt, ist die gleichzeitige Ausgabe der Synonyme des eingegebenen Suchbegriffs. Die Synonyme würde er nutzen, um nach weiteren Büchern zu fahnden. Zwar gibt es hierzu bereits vorgefertigte Lösungen im Netz allerdings sind diese nur wenig komfortabel und nicht im Rahmen unseres Projektes nutzbar.

Beispiel:

- Der Benutzer gibt den Suchbegriff (z. B. „Java“) ein und klickt den Knopf „suchen“.
- Das Programm zeigt dem Benutzer einerseits die gefunden Bücher an, die das Schlüsselwort im Titel enthalten.
- Weiterhin zeigt das Programm eine Liste der Synonyme an („Programmiersprache, Espresso, Insel, Jamocha, Tinte, ...“).
- Der Benutzer klickt z. B. auf das Synonym „Insel“.
- Das Programm zeigt umgehend die Bücher mit dem Begriff „Insel“ im Titel an.
- sowie die zugehörigen Synonyme („Oase, Sandbank, ...“).
- Nun kann der Benutzer wiederum ein Synonym wählen, usw.

Wir werden dieses Projekt in den kommenden Wochen realisieren.

Ein weiterer Punkt, der Ihrem Auftraggeber am Herzen liegt, ist folgender: In WikiBooks werden Einträge von einer Vielzahl von Nutzern gepflegt. Die Aktualität eines Eintrags und insbesondere der Urheber (des Eintrags und nicht des Buches) sind jeweils auf der „Versionsgeschichte“-Seite (s. Abbildung 2) einsehbar. Ihr Auftraggeber möchte die Information, von wem die letzte Änderung am angezeigten Artikel stammt, sofort sehen können, ohne auf diese-Seite wechseln zu müssen.



## Abbildung 2: Versionsgeschichte in WikiBooks

### Erste Teilaufgabe

Erweitern Sie Ihr Bibliotheksprogramm aus dem letzten Aufgabenblatt (bzw. falls notwendig die Lösung eines Kommilitonen) **um eine Klasse und entsprechende Methoden**, die ein Buch mittels eines Suchbegriffs in WikiBooks sucht und aus dem zurückgegebenen XML-Dokument den letzten Bearbeiter herausucht.

WikiBooks sollen im Zettelkasten gespeichert werden. Erweitern Sie Ihr Programm entsprechend. Beachten Sie dabei evtl. Gemeinsamkeiten mit den Klassen „Buch“ und „Elektronisches Medium“ und überlegen Sie wie diese bspw. mittels Vererbung genutzt werden können, bspw. über Interfaces. (Tipp: Wird für Aufgabe D7 benötigt).

Zum Auslesen verwenden Sie eine `URLConnection` auf die Web-URL

`https://de.wikibooks.org/wiki/Spezial:Export/`. Z. B. lautet die URL für das Buch Java-Standard:

„`https://de.wikibooks.org/wiki/Spezial:Exportieren/Java_Standard`“. Die Rückgabe dieses Requests ist ein XML-Dokument. Sie können und sollten sich nun zuerst das zurückgegebene Dokument einfach mit einem Webbrowser ansehen, in dem Sie obige URL eingeben. Sie erhalten mit dem Firefox in etwa folgendes:

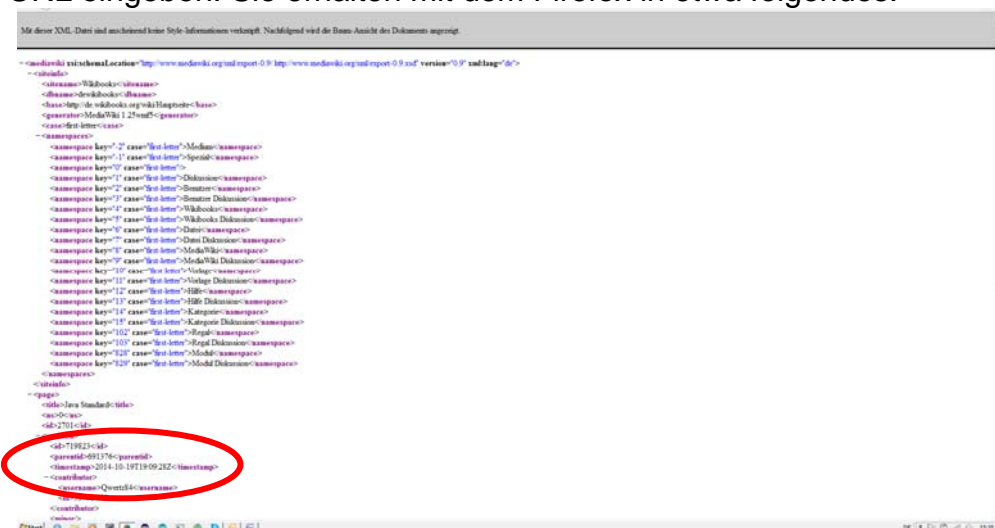


Abbildung 3: Ausschnitt aus der Export-Seite des Wikibooks Eintrag „Java Standard“ (Stand 30.10.2014)

Uns interessiert das markierte Element mit folgendem Pfad im XML-Dokument:

- mediawiki - page - revision - contributor

An dieser Stelle ist verzeichnet, wer die letzte Änderung am Eintrag vorgenommen hat. Der Verursacher wird entweder als IP-Adresse angegeben (wie im unteren Beispiel), oder als username (wie im obigen Beispiel in Abb. 3).

Tipp: Testfall „RFID-technologie“ (Stand 28.11.2019)

```
<revision>
  <id>630304</id>
  <parentid>623857</parentid>
  <timestamp>2012-03-09T19:27:11Z</timestamp>
  <contributor>
    <ip>109.43.20.218</ip>
  </contributor>
```

Abbildung 4: Ausschnitt aus der Export-Seite von Wikibooks zu „Die Kunst, glücklich zu leben“ (30.10.2014)

Detail-Informationen über das Format der Export-Seite erhalten Sie unter <http://meta.wikimedia.org/wiki/Help:Export>.

Verwenden Sie nun eine `URLConnection`, um das obige Dokument aus einem Programm heraus zu laden.

Hinweis: Der Wikipedia-Server antwortet mit einer Fehlermeldung, wenn der `UserAgent` nicht gesetzt ist. Setzen Sie deshalb bitte in Ihrem Request die Eigenschaft `UserAgent` auf einen gültigen Wert, z. B. "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; .NET CLR 1.1.4322)". oder Mozilla/5.0 (Linux; <Android Version>; <Build Tag etc.>) AppleWebKit/<WebKit Rev>(KHTML, like Gecko) Chrome/<Chrome Rev> Safari/<WebKit Rev>

Weitere mögliche Werte finden Sie z. B. unter [http://en.wikipedia.org/wiki/User\\_agent](http://en.wikipedia.org/wiki/User_agent).

Sie müssen aus dem XML-Dokument mit Hilfe eines XML-Parsers (SAX, StAX) den Pfad `mediawiki - page - revision - contributor` heraussuchen. In `contributor` müssen Sie prüfen, ob `username` oder `ip` angegeben ist. Entnehmen Sie dann das angegebene Element.

Das Programm soll den zu suchenden Begriff als Kommandozeilenparameter entgegennehmen und die gefundene Information zum Urheber auf der Konsole ausgeben. Das sieht nach einem Export des Projekts als JAR dann z. B. so aus:

```
C:\...\D.5> java -classpath D5.jar D5.WikipediaBooksContributorRequest
Java_Standard
Urheber: QWertz84
```

```
C:\...\D.5> java -classpath D5.jar D5.WikipediaContributorRequest
Die_Kunst_Glücklich_Zu_Leben
Urheber: 109.43.20.218
```

Hinweis: Sie können bei einem Konsolenprogramm auch aus der IDE heraus Kommandozeilenparameter beim Start übergeben. Und zwar so:

In IntelliJ

- ALT+SHIFT+F10 (Windows) oder CTRL+ALT+R (Mac)
- Klick auf „+“ und hinzufügen einer Konfiguration für „Application“ oder Auswahl der zu ändernden Konfiguration
- Eintragen der Parameter in der Zeile „Program Argument“

In Eclipse:

- Im Projektmappen-Explorer: rechte Maustaste auf dem Projekt → Properties → Run/Debug Settings → Launch configuration auswählen → Edit... → Register „(x)= Arguments“ → in die Textbox Kommandozeilen-Argumenten eingeben → Apply/OK → aus Eclipse heraus starten (Run/Debug)



## D.6 Hausaufgabe (XML, Datumsformate und Zeitzonen)

[2 Punkte]

Wenn Sie diese Aufgabe bearbeiten, können Sie sie in Gestalt eines Programms zusammen mit Aufgabe D.5 abgeben.

Erweitern Sie das Programm aus Aufgabe D.5 um die Fähigkeit, den Zeitpunkt der letzten Bearbeitung zu ermitteln. Sie finden diesen Zeitpunkt in dem bereits in Aufgabe D.5 verarbeiteten XML-Dokument unter dem Pfad „mediawiki – page – revision – timestamp“.

Beispielhaft betrachten wir das XML-Dokument zum Artikel „Java Standard“ in Abbildung 3. Der dort angegebene Zeitstempel hat das Format *2012-09-30T22:02:04Z*. Das **T** und das **Z** fungieren als Trennzeichen. Es handelt sich hierbei um das ISO 8601-Format (oder zumindest eine Abwandlung davon).



Abbildung 5: Ausschnitt aus dem Wikibooks-Eintrag „Java Standard“ (Stand 31.10.2014)

Betrachten wir nun den Original-Artikel „[http://de.wikibooks.org/wiki/Java\\_Standard](http://de.wikibooks.org/wiki/Java_Standard)“, so erkennen wir, dass ganz unten auf der Seite ebenfalls der letzte Änderungszeitpunkt angegeben ist (siehe Abbildung 5). Das Datum im XML-Dokument ( `<timestamp>2014-10-31T09:06:44Z</timestamp>` ) scheint dem Datum in Abbildung 5 zu widersprechen. Der Grund der Abweichung von in diesem Fall einer<sup>1</sup> Stunde ist, dass das Datum in der XML-Datei in der UTC-Zeit notiert wird, das Datum im Browser hingegen in lokaler Zeit (Details: [http://de.wikipedia.org/wiki/Koordinierte\\_Weltzeit](http://de.wikipedia.org/wiki/Koordinierte_Weltzeit)).

Ihr Programm soll nun folgende Ausgabe erzeugen:

```
C:\...\D.5> java -classpath D5.jar D5.WikibooksContributorRequest Java_Standard
Suche nach: Java Standard
Letzte Änderung: 31.10.2014 um 10:06 Uhr
Urheber: QWertz84
```

Das heißt, Ihr Programm soll die in UTC-Zeit vorliegende Zeichenkette in eine Zeitangabe lokaler Zeit umwandeln.

Natürlich kann man die in der XML-Datei enthaltene Zeichenkette nun von Hand auseinander nehmen und versuchen, die Zeitzonenunterschiede von Hand nachzubilden. Es geht aber wesentlich eleganter. Hierzu erfolgt nun eine kleine Hilfestellung:

- Die Klasse `SimpleDateFormat` unterstützt das Parsen und Formatieren einer Zeichenkette in ein `Date`-Objekt. Dabei können Zeitzonen-Unterschiede Berücksichtigung finden.
- Lesen sie dazu in „Java ist auch eine Insel“, Kapitel 15.

## D.7 Hausaufgabe (XML Parsen, String-Operationen)

[2 Punkte]

Wenn Sie diese Aufgabe bearbeiten, können Sie sie in Gestalt eines Programms zusammen mit Aufgabe D.5 und D.6 abgeben.

<sup>1</sup> Bei Datumsangaben im Winter (Winterzeit) beträgt die Abweichung nur 1 Stunde ansonsten 2 Stunden.

WikiBooks unterscheidet u.a. zwischen Regalen (Kategorien), Büchern und Kapiteln. Zurzeit (Stand 30.10.2017) findet sich z.B. das Buch „Die Sprache der Mathematik“ im Regal „Mathematik“ und hat die Kapitel „Logik“, „Mengenlehre“, „Mathematische Grundstrukturen“ und „Die Evolution der mathematischen Sprachformen“. Dies ist in Abbildung 6 erkennbar.

Abbildung 6: Übersicht zu „Die Sprache der Mathematik“ (Stand 30.10.2017)

**Abbildung 7: Ausschnitt aus der Export-Seite für „Die Sprache der Mathematik“ (Stand 30.10.2017)**

**Tip:** Sollte der Parser nicht den vollständigen Inhalt des Text-Tags liefern finden Sie in der Datei „Große Tags“ – ILIAS, eine kleine Hilfestellung.

```
C:\...\D.5> java -classpath D5.jar D5.WikibooksContributorRequest
Die_Sprache_Der_Mathematik
Regal: Mathematik
Kapitel:
1 Logik
2 Mengenlehre
3 Mathematische Grundstrukturen
4 Die Evolution der mathematischen Sprachformen
Letzte Änderung: 28. November 2015 um 16:50 Uhr
Urheber: Klaus Eifert
```



[OPTIONAL]

Neben der Auswertung des Text-Tags ist es auch möglich, die gewünschten Daten über die Wikimedia-API zu erhalten. Vgl.: <https://www.mediawiki.org/wiki/API:Tutorial>

Weiterführende Links:

Regal:

[https://de.wikibooks.org/w/api.php?action=parse&format=xml&prop=links&page=Java\\_Standard&redirect](https://de.wikibooks.org/w/api.php?action=parse&format=xml&prop=links&page=Java_Standard&redirect)

Inhaltsverzeichnis:

[https://de.wikibooks.org/w/api.php?action=parse&format=xml&prop=sections&page=Java\\_Standard&redirect](https://de.wikibooks.org/w/api.php?action=parse&format=xml&prop=sections&page=Java_Standard&redirect)

### ***D.8 Hausaufgabe (Bonus – Bearbeitung freiwillig)***

[1 Punkt]

Wenn Sie diese Aufgabe bearbeiten, können Sie sie in Gestalt eines Programms zusammen mit Aufgabe 5 bis C.7 abgeben.

Zusätzlich zu den geplanten Sicherungsoptionen aus Aufgabenblatt C soll es auch möglich sein die Daten im XML-Format zu sichern. Nutzen Sie das gegebene Interface „*Persistency*“ und implementieren sie eine Klasse „*XMLPersistency*“. Diese soll das Speichern und Laden eines Zettelkastens in XML-Repräsentation erlauben.