

**Übungsblatt 5 WT:IV**

Bis zum 29.06.2022, 23:59 sind Lösungen zu folgenden Aufgaben abzugeben: 1 2a-c 3a,b,c,f, 5b,c, 6b, 7b-e.

**Aufgabe 1 : XML-API Wikipedia (1+2+2 Punkte)**

Laden Sie sich die bereitgestellte [XML-Datei](#) herunter, die alle Wikipedia-Artikel in simpler englischer Sprache enthält.

- (a) Welche XML-API sollten Sie am besten verwenden, um diese Datei zu verarbeiten? Begründen Sie Ihre Entscheidung.
- (b) Schreiben Sie ein Python-Programm, dass die XML-Datei mithilfe der vorgeschlagenen API verarbeitet. Ermitteln Sie dabei die Anzahl der enthaltenen Titelemente aller Artikel und geben Sie diese aus.
- (c) Erweitern Sie Ihr Programm so, dass zusätzlich die Anzahl der Titel der Wikipedia-Artikel ausgegeben wird, die eine Jahreszahl enthalten. Als Jahreszahl betrachten Sie alle vierstelligen Zahlen, die nicht in der Zukunft liegen (d.h. 1000–2022). Benutzen Sie zur Umsetzung einen regulären Ausdruck.

**Aufgabe 2 : CGI, Servlets, PHP (2+1+1+0 Punkte)**

- (a) 100 Benutzer greifen gleichzeitig auf ein CGI-Programm, das als Shell-Skript realisiert ist, und auf ein Servlet zu. Beschreiben Sie, wie der Web-Server jeweils mit den gleichzeitigen Anfragen umgeht und stellen Sie die Unterschiede der Verarbeitung heraus.
- (b) Wie kann Session-Tracking in einem CGI-Programm implementiert werden? Nennen Sie zuerst die dafür nötigen Funktionen, Objekte, Variablen, etc. und beschreiben Sie dann zweitens knapp, was dabei im Web-Server geschieht. Beziehen Sie sich bei Ihrer Antwort auf eine einzelne Sprache für CGI-Programme.
- (c) Wie kann Session-Tracking in einem Servlet implementiert werden? Nennen Sie zuerst die dafür nötigen Funktionen, Objekte, Variablen, etc. und beschreiben Sie dann zweitens knapp, was dabei im Web-Server geschieht.
- (d) Entwerfen Sie ein PHP Programm, das Session-Tracking realisiert. Das Programm soll dazu manuell einen Cookie setzen und (ebenfalls manuell) wieder auslesen und ausgeben. "Manuell" heißt, dass auf die Verwendung von `set_cookie()`, `$_COOKIE`, und `session_start()` verzichtet werden soll. Stattdessen sollen `header()` und `getallheaders()` verwendet werden.

Aufgabe 3 : Multiple Choice (1+1+1+0+0+1 Punkte)

*Hinweis:* Für die Beantwortung von Teilaufgabe (a) gibt es einen Bonuspunkt.

Kreuzen Sie Zutreffendes an:

(a) Folgende Programme und Dokumenttypen können unmittelbar als CGI-Programm verwendet werden:

- ☐ Jede Java-Servlet-Klasse
- ☐ Jedes in einer Shell ausführbare Programm
- ☐ Jedes Shell-Skript
- ☐ Jedes JSP-Dokument

(b) Bei jeder HTTP-GET-Anfrage an ein Servlet wird dieses zunächst in Maschinencode der Zielpattform übersetzt.

- ☐ Richtig
- ☐ Falsch
- ☐ Es wird jedes Mal Bytecode erzeugt

(c) Folgende Aussagen treffen auf einen Apache-2-Server zu:

- ☐ Er läuft standardmäßig auf einem dynamischen Port
- ☐ Er läuft standardmäßig auf Port 80
- ☐ Er läuft standardmäßig auf Port 443
- ☐ Er kann XHTML-Seiten ausliefern
- ☐ Er ist ein Servlet-Container
- ☐ Er kommuniziert auf Bitübertragungsschicht

(d) Jakarta HTTPServlets kommunizieren

- ☐ auf ISO/OSI-Schicht 2
- ☐ auf ISO/OSI-Schicht 3
- ☐ auf ISO/OSI-Schicht 4
- ☐ auf ISO/OSI-Schicht 7

(e) Jakarta Servlets kommunizieren im Allgemeinen

- ☐ auf ISO/OSI-Schicht 2
- ☐ auf ISO/OSI-Schicht 3
- ☐ auf ISO/OSI-Schicht 4
- ☐ auf ISO/OSI-Schicht 7

(f) Welche der folgenden Aussagen aus dem Gebiet der formalen Sprachen sind wahr?

- ☐ Es gibt kontextfreie Sprachen die sich nicht mit einem regulären Ausdruck definieren lassen.
- ☐ Die Beschreibungsmächtigkeit endlicher Automaten ist äquivalent zu der von regulären Ausdrücken.
- ☐ Die Beschreibungsmächtigkeit von Kellerautomaten ist äquivalent zu der von regulären Ausdrücken.

Beachten Sie, dass zu einer Frage mehrere Antworten zutreffen können. Eine Frage gilt als richtig beantwortet, falls alle zutreffenden und keine unzutreffende Antwort angekreuzt ist.

#### Aufgabe 4 : Grammatiken und reguläre Ausdrücke

- (a) Was versteht man unter dem Wortproblem einer regulären Sprache?
- (b) Gegeben sei eine Grammatik  $G = (\{S, A, B, C\}, \{0, 1\}, P, S)$  mit

$$P = \{S \rightarrow 01A, A \rightarrow 00B, B \rightarrow 1A, B \rightarrow 0C, C \rightarrow 00\}$$

Geben Sie einen regulären Ausdruck an, der die zu  $G$  gehörende Sprache spezifiziert.

- (c) Gegeben sei die Sprache  $L = \{w | w \in \{0, 1\}^* \text{ mit } w = 1^m 0^n 1, m \geq 0, n = 3m\}$ . Geben Sie eine kontextfreie Grammatik  $G$  an, die  $L$  erzeugt.
- (d) Wenn die Produktionen einer Grammatik der Form  $A \rightarrow wB$  oder  $A \rightarrow w$  sind, wobei  $A, B$  Nichtterminale und  $w$  eine (möglicherweise leere) Zeichenkette von Terminalen ist, so nennen wir die Grammatik rechtslinear.  
Geben Sie für die von  $R = (011)^* 00(011)^*$  spezifizierte Sprache eine rechtslineare Grammatik an.

#### Aufgabe 5 : Web-Server (0+1+1 Punkte)

- (a) In der Vorlesung wurde der Apache Webserver vorgestellt. Nennen sie mindestens zwei weitere Webserver und erläutern sie kurz, wie sich diese vom Apache unterscheiden. In welchen Anwendungsfällen würden sie die Alternativen dem Apache vorziehen?
- (b) Informieren sie sich über FastCGI und nennen sie den wichtigsten Unterschied gegenüber CGI. Erklären sie die Vorteile von FastCGI, die sich daraus ergeben.
- (c) Erklären sie kurz die folgenden Konzepte: HTTP-Redirects, das HTML Tag `<meta http-equiv="refresh">` und Proxy Server. In welcher Situation würden Sie welche Technologie einsetzen?

#### Aufgabe 6 : Reguläre Ausdrücke mit PHP (0+2 Punkte)

- (a) Erstellen Sie Funktionen in PHP zur Überprüfung, ob eine Eingabe die folgende Form hat (Verwenden Sie dazu reguläre Ausdrücke):
  - (a1) eine IPv4-Adresse (Stellen Sie sicher, dass Sie nur Blöcke im Bereich 0–255 zulassen)
  - (a2) eine E-Mail-Adresse.
  - (a3) eine URL nach [RFC 3986](#) (Beschränken Sie sich auf die URL-Bestandteile Scheme, Authority und Path)

Hinweis: In den obigen Beispielen können die korrekten Ausdrücke mitunter sehr lang und schwer verständlich werden. Wenn dies der Fall ist, beschränken Sie sich auf eine Lösung, die möglichst präzise ist, deren regulärer Ausdruck aber auch in maximal 3 Zeilen Code passt.

- (b) Informieren Sie sich über sogenannte “Capturing Groups”, “Backreferences” (Rückwärtsreferenzen), und “Lookahead/Lookbehind” in regulären Ausdrücken.
  - (b1) Erklären Sie in diesem Zusammenhang den Unterschied zwischen den Ausdrücken `/(aa)?bb/` und `/(?:aa)?bb/`.

(b2) Erklären Sie weiterhin den Unterschied zwischen den drei Ausdrücken:

```
/hello(=? world) /  
/hello( world) /  
/hello world/
```

Geben Sie an, welcher Gesamtausdruck zurückgegeben wird, sowie (falls vorhanden) den Inhalt aller Capture-Groups.

#### Aufgabe 7 : Programmiermodul: HTTP-API P (0+2+1+3+2+0 Punkte)

Schreiben Sie einen [Flask-Server](#), der das Model Ihrer Notiz-App als HTML rendert und Änderungen durch den Benutzer über den Browser erlaubt und zurück in die Datei serialisiert.

Das [Repository](#) enthält ein Grundgerüst für Ihren Server, das Sie verwenden können, wenn Sie die vorigen Programmiermodule nicht fertiggestellt haben. Installieren Sie die benötigten Abhängigkeiten mittels `pip3 install -r requirements.txt`. Starten Sie den Server über die Hauptdatei `server.py` (starten Sie die Datei direkt über den Python-Interpreter, nicht über `flask run`).

- (a) Laden Sie Ihr `NotesAppModel` in der `main`-Funktion über die `create`-Methode, starten Sie den Flask-Server und liefern Sie für die HTTP-Route `"/` das gerenderte HTML-Template `templates/index.html` mit dem aktuellen `Last-Modified`-Datum des Models aus.

Code für diese Teilaufgabe ist im Gerüst vorgegeben.

- (b) Vervollständigen Sie das Template `templates/index.html`, sodass die Inhalte der geladenen `NotesAppModel`-Instanz im Browser dargestellt werden. Die hierfür benötigten Template-Variablen werden bereits in der mitgelieferten Hilfsfunktion `render_index` gesetzt. Das Layout der Einträge muss den Vorgaben aus Übungsblatt 2 folgen.

Hinweis: Das Template verwendet die [Jinja2-Syntax](#).

Variablen werden ausgegeben mithilfe von `{{ variable }}`. Kontrollanweisungen werden geschrieben als `{% anweisung %} ... {% endanweisung %}`.

Beispiel `if`-Statement:

```
{% if variable == "value" %}  
    <div>Variable ist {{ variable }}.</div>  
{% endif %}
```

Beispiel `for`-Schleife:

```
{% for i in list_variable %}  
    <div>{{ i }}</div>  
{% endfor %}
```

- (c) Fügen Sie dem Template ein HTML-Formular hinzu, über das der Nutzer einen neuen Eintrag hinzufügen kann. Das Formular soll über POST an `"/create"` gesendet werden und enthalten:
- Radio-Buttons für den Typ (Notiz/ToDo),
  - Textfelder für Titel und Autor,
  - eine Textarea für den Inhalt.
- (d) Ergänzen Sie den Flask-Server um eine HTTP-Route `"/create"`, die die Inhalte des Formulars über POST akzeptiert. Die Handler-Funktion soll...
- (d1) die Inhalte auf Vollständigkeit und den Eintragstyp auf Korrektheit validieren,
- (d2) im Fehlerfall eine Fehlermeldung im Template ausgeben und dabei die eingetragenen Formularinhalte erhalten,

- (d3) bei erfolgreicher Validierung den Inhalt der Textarea zeilenweise in Notizen oder Todo-Items splitten und dem Model einen entsprechenden Eintrag hinzufügen,
- (d4) nach erfolgreichem Hinzufügen (und nur dann) per HTTP-Status 303 auf "/" zurückleiten.

Hinweise: Auf die aktuellen POST-Formularinhalte können Sie über `flask.request.form` zugreifen. Fehlermeldungen und Formularwerte können Sie dem Template als Variablen übergeben. HTTP-Redirects können Sie über `flask.redirect` vornehmen.

- (e) Fügen Sie jedem Eintrag ein `<footer>`-Element mit einem „Löschen“-Button hinzu. Beim Klick auf den Button, soll die ID des angeklickten Eintrags per POST an `/update` gesendet werden. Auf der Serverseite implementieren Sie einen Handler für die Route, der...
  - bei existierendem Eintrag diesen löscht und per HTTP-Status 303 auf "/" zurückleitet,
  - im Fehlerfall (fehlende oder falsche ID) einen HTTP-Status 400 zurückgibt (Sie können hierfür `flask.abort` nutzen).

- (f) **Bonus (2 Punkte):** Ersetzen Sie die Bulletpoints der Todo-Listen durch HTML-Checkboxen und fügen Sie den `<footer>`-Elementen der Todo-Einträge jeweils „Speichern“-Buttons hinzu. Die Checkboxen sollen den `done`-Zustand der einzelnen Items widerspiegeln. Beim Klick auf einen Button soll der jeweilige Eintrag per POST mit der aktuellen Auswahl aktualisiert werden.

Tipp: Sie können im Template über die spezielle Variable `{{ loop.index0 }}` den aktuellen (von null gezählten) Index einer `for`-Schleife abrufen.

## Hinweise zur Abgabe

- Erstellen Sie *eine* PDF-Datei, in der Texte und Grafiken zu den Aufgaben enthalten sind.
- Quellcode (Python, Java, JavaScript, PHP, HTML, CSS, XML, XSL, XSD, etc.) innerhalb des PDF-Dokuments wird nicht korrigiert. Quellcode jeder Aufgabe soll, wie im Tutorium vorgestellt, über GitLab verwaltet werden.
- Stellen Sie sicher, dass Sie keine in der `.gitignore`-Datei gelisteten Dateien und Verzeichnisse versehentlich mit committen.
- Referenzieren Sie Ihre Quellcode-Dateien in dem PDF-Dokument, so dass sie einer Aufgabe eindeutig zuzuordnen sind. Verlinken sie sie entsprechendem Repositories.
- Gruppenabgaben mit bis zu drei Personen sind erlaubt; pro Gruppe genügt dann eine Abgabe, in der alle Gruppenmitglieder mit Namen, Matrikel, und E-Mail-Adresse verzeichnet sind.
- Abzugeben ist *eine* PDF-Datei über Moodle.