

Übungsblatt 5 WT:IV

Bis zum 27.06.2019, 23:59 sind Lösungen zu folgenden Aufgaben abzugeben: 1a,c, 2a,b, 3, 5, 6, 7, 8a,b.

Für die Programmieraufgaben dieses Übungsblattes benötigen Sie einen Web-Server zur Ausführung von CGI und (optional) PHP-Skripten sowie einen Servlet-Container zur Ausführung von Servlets und JSP-Skripten. Wir empfehlen die Nutzung des [Apache HTTP Server](#) sowie von [Apache Tomcat](#). Unter Windows bietet sich die Nutzung der Distribution [XAMPP](#) an.

Bringen Sie bitte zum Tutorial einen Laptop mit installiertem Tomcat mit.

Aufgabe 1 : CGI (1+0+2 Punkte)

- (a) 100 Benutzer greifen gleichzeitig auf ein CGI-Programm zu, das als Shell-Skript realisiert ist. Beschreiben Sie, wie der Web-Server mit den gleichzeitigen Anfragen umgeht.
- (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) Entwerfen Sie mit Hilfe von CGI ein endloses Dokument: Jedes generierte Dokument soll eine Seitenzahl nennen und einen Anchor-Link auf das nächste Dokument (Seitenzahl + 1) haben, das wiederum dynamisch generiert wird.

Aufgabe 2 : Servlets (1+1+0 Punkte)

- (a) 100 Benutzer greifen gleichzeitig auf ein Servlet zu. Beschreiben Sie, wie der Servlet-Container mit den gleichzeitigen Anfragen umgeht.
- (b) 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.
- (c) Entwerfen Sie ein Servlet um "Schere, Stein, Papier" zu spielen. Wählen Sie auf einer statischen HTML-Seite zwischen Schere, Stein und Papier und schicken Sie die Auswahl zum Server, der zufällig wählt und beide Auswahlen sowie den Gewinner anzeigt. Geben Sie Ihr Projekt als lauffähiges `.war`-File ab.

Aufgabe 3 : JSP (4 Punkte)

- (a) Schreiben Sie ein JSP-Skript, das [netspeak.org](#) anfragt um Sätze zu bauen. Der Anfang des Satzes ist "you". Der Nutzer soll per Link eines der 5 häufigsten nächsten Worten – basierend auf den letzten 2 Worten des Satzes – auswählen, das dann an den Satz angefügt wird. Dann soll eine neue Wort-Auswahl für den aktuellen Satz erscheinen.

Beispiel: Der Nutzer startet mit dem Satz "you" und der [Auswahl](#) zwischen "can", "are", "have", "will" und "to". Er wählt "have". Jetzt hat er den Satz "you have" und die [Auswahl](#) zwischen "a", "to", "any", "not" und "the". Er wählt "to". Jetzt hat er den Satz "you have to" und die [Auswahl](#) zwischen "be", "do", "say", "go" und "pay"...

Aufgabe 4 : Multiple Choice

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

- Er läuft standardmäßig auf Port 2
- Er läuft standardmäßig auf Port 80
- Er kann XHTML-Seiten ausliefern
- Er ist ein Servlet-Container
- Er kann jedes Verzeichnis im Document-Root mit einem anderen Passwort schützen
- Er kommuniziert auf Bitübertragungsschicht

(c) Wenn ein Java HTTP-Servlet durch eine HTTP-Anfrage angesprochen wird, führt der Tomcat-Servlet-Container jedes mal u. a. folgende Aktionen aus:

- Überprüfung ob eine Instanz dieses Servlets läuft
- Übersetzung des Servlets in den Maschinencode der Plattform des Browsers, von dem die Anfrage stammt
- Erzeugung eines `HttpServletRequest`-Objekts
- Starten der Java-Virtual-Machine

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 5 : Grammatiken und reguläre Ausdrücke (1+1+1+1 Punkte)

(a) Was versteht man unter dem Wortproblem für eine reguläre 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 $L = \{w \mid w \in \{0, 1\}^* \text{ mit } w = 1^m 0^n 1, m \geq 0, n = 3m\}$. 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 6 : Web-Server (1+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, um Weiterleitungen zu realisieren: HTTP-Redirects, Proxy Server und das HTML Tag `<meta http-equiv="refresh">`. Wann würden sie welche Methode bevorzugen?

Aufgabe 7 : Reguläre Ausdrücke mit PHP (3+2 Punkte)

- (a) Erstellen Sie in PHP Funktionen 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” und “Backreferences” (Rückwärtsreferenzen) in regulären Ausdrücken.
 - (b1) Erklären Sie in diesem Zusammenhang den Unterschied zwischen den Ausdrücken `"/(aa)?bb/"` und `"/(?:aa)?bb/"`.
 - (b2) Erstellen Sie eine Funktion, die alle Vorkommen von URL-Schemas `"([a-z]+)://"` durch Anhängen von `"s"` mit ihrer sicheren Variante ersetzt (z.B. `"http://"` zu `"https://"`). Nutzen Sie hierzu die PHP-Funktion `preg_replace()` und Backreferences (siehe hierzu: php.net).

Aufgabe 8 : Programmiermodul: HTTP-API P (3+2+0 Punkte)

Schreiben Sie ein Servlet, das HTTP Lese- und Schreib-Anfragen für den FeedReader implementiert.

Arbeiten Sie für diese Aufgabe in der Ordnerstruktur des `feed-reader-http-api` aus dem Repository dieses Beleges. Die mitgelieferte `readme.txt` enthält Hinweise zum Deployment und zur Einbringung Ihrer Dateien aus den vorangegangenen Übungen.

- (a) Schreiben Sie ein Servlet, das den Feed unter localhost:8080/feed-reader/Feed zur Verfügung stellt. Schreiben Sie eine entsprechende `WEB-INF/web.xml`.
- (b) Implementieren Sie das Hinzufügen eines neuen Eintrags im Servlet. Die Datei `add-entry.html` (zu erreichen unter localhost:8080/feed-reader/add-entry.html) enthält das Formular, das die Daten mittels POST an das Servlet sendet. Fügen Sie den Eintrag mit der entsprechenden Methode aus der letzten Übung dem Feed hinzu. Nutzen Sie eine Response mit See Other (303) Statuscode und Location-Header um den Nutzer nach dem Hinzufügen auf die Ausgabe des Feeds aus (b) weiterzuleiten.

- (c) Erweitern Sie die bereits implementierten Methoden, so dass neben eines herkömmlichen GET-Requests auch ein Conditional GET-Request auf Basis des letzten Änderungsdatums des Feeds benutzt werden kann.

Hinweise zur Abgabe

- Erstellen Sie *eine* PDF-Datei, in der Texte und Grafiken zu den Aufgaben enthalten sind.
- Quellcode (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.
- Referenzieren Sie Ihre Quellcode-Dateien in dem PDF-Dokument, so dass sie einer Aufgabe eindeutig zuzuordnen sind. Verlinken sie sie entsprechendem Repositories.
- Abzugeben ist *eine* PDF-Datei die wie folgt benannt ist:
<Nachname>-<MatrikelNr>-webtec-blatt<Übungsblattnummer>.pdf.
- 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
- Legen sie die PDF-Datei in das GitLab Repository, in dem auch der Code ihrer Gruppe verwaltet wird.