3.11.1 Testfragen

Folgende Fragen sollten Sie ohne große Schwierigkeiten beantworten können. Eine oder mehrere Antworten sind jeweils richtig. Die erläuterten Auflösungen mit zusätzlichen Hintergrundinformationen finden Sie in Anhang A, *Lösungen*.

- **Frage 1:** Bei mit `xsl:import` importierten Stylesheet-Modulen ...
  a) ... hat das zuerst importierte Stylesheet-Modul Vorrang.
  b) ... hat das zuletzt importierte Stylesheet-Modul Vorrang.
  c) ... sind beide Stylesheet-Module gleichrangig.
- **Frage 2:** Bei mit `xsl:include` inkludierten Stylesheet-Modulen ...
  a) ... gilt bei Konflikten die zuerst stehende Regel.
  b) ... gilt bei Konflikten die zuletzt stehende Regel.
  c) ... gelten beide Regeln, da hier keine Importpräzedenz greift.
- **Frage 3:** Was ist bei benannten Templates wichtig?
  a) Es muss ein Template mit dem aufgerufenen Namen existieren.
  b) Das `mode`-Attribut kann beim Aufruf nicht verwendet werden.
  c) Das benannte Template darf kein `select`-Attribut besitzen.
  d) Es kann ein Parameter übergeben werden.
- **Frage 4:** Ein globaler Parameter ...
  a) ... wird immer in einer Template-Regel deklariert.
  b) ... besitzt einen Default-Wert.
  c) ... kann während der Laufzeit des Stylesheets einen neuen Wert erhalten.
  d) ... behält seinen Wert während der gesamten Laufzeit des Stylesheets.
- **Frage 5:** Bei zwei gleich benannten Variablen ...
  a) ... verdeckt die lokale die globale Variable.
  b) ... ist immer die globale Variable sichtbar.
  c) ... gilt die zuletzt deklarierte.
  d) ... gibt es bei zwei lokalen Variablen einen Fehler.
- **Frage 6:** Welche dieser Aussagen zu `id()` und `generate-id()` sind richtig?
  a) `id()` kann in der DTD deklarierte Identifier finden.
  b) `id()` kann durch `generate-id()` generierte Identifier finden.
  c) `generate-id()` kann mehrere Identifier pro Element erzeugen.
  d) `generate-id()` erzeugt je Knoten immer den gleichen Wert.

- **Frage 7:** Welche dieser Aussagen zu Schlüsseln sind korrekt?
  a) Es kann mehrere gleich benannte `key`-Listen geben.
  b) Einem Knoten können mehrere Schlüsselwerte zugeordnet sein.
  c) Ein Schlüssel muss in der DTD deklariert werden.
  d) Ein Schlüssel muss im Haupt-Stylesheet stehen.

3.11.2 Aufgaben

- **Aufgabe 1:** Benannte Templates
  In einem in XML vorliegenden Artikel soll am Ende jedes Kapitels mittels eines als Subroutine fungierenden benannten Templates eine formatierte Trennlinie eingefügt werden.
  Benötigte Datei: *aufgaben/kap03/a1-artikel.xml*
- **Aufgabe 2:** Variablen und Parameter
  Die vorige Lösung wird erweitert. Nun soll am Ende eines Kapitels als Trenner »Ende von Kapitel 1« etc. ausgegeben werden. Da bereits ein benanntes Template vorliegt, das den Trenner einfügt, soll es auch diese Arbeit übernehmen. Dem benannten Template muss das aktuelle Kapitel mitgeteilt werden. Dies geschieht durch die Übergabe eines Parameterwertes.
  Benötigte Datei: *aufgaben/kap03/a2-artikel.xml*
- **Aufgabe 3:** Laufzeitidentifier
  Mittels `generate-id()` soll ein Inhaltsverzeichnis mit Sprunglinks der Ausgabe eines in XML vorliegenden Artikels vorangestellt werden.
  Benötigte Datei: *aufgaben/kap03/a4-generate-id.xml*
- **Aufgabe 4:** Unparsed Entities
  Eine Variation der Datei mit Buchverkäufen soll hier noch mal hergenommen werden, um zu zeigen, dass man ihr noch ein paar zusätzliche Features abgewinnen kann. Es liegen eine XML-Datei, Grafiken und die DTD vor.
  In der DTD sind Buchtitel-Grafiken (*buch1.gif* bis *buch10.gif*) als unparsed Entities deklariert. Diese sollen in Zusammenhang mit den Buchinformationen ausgegeben werden.
  Benötigte Dateien: *aufgaben/kap03/a5-buchdaten.xml, buch1.gif-buch10.gif*
- **Aufgabe 5: Unparsed Entities**
  In der XML-Datei ist je Buch ein Element `<verkauf>` vorhanden, das aktuelle Verkaufszahlen enthält. Die Zahl soll als Balkendiagramm umgesetzt werden, wozu ein HTML-`<img>` (Grafik: *rot.gif*) dient, dessen `width`-Attribut entsprechend gesetzt wird.
  Benötigte Dateien: wie eben; zusätzlich *rot.gif*

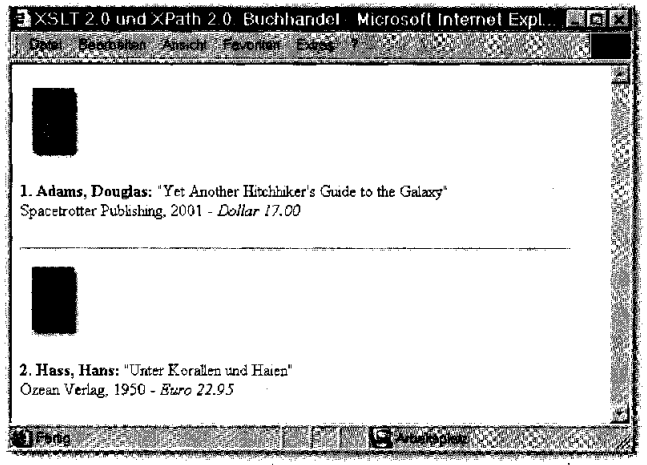

Abbildung 3.11 Buchdatei mit Illustrationen

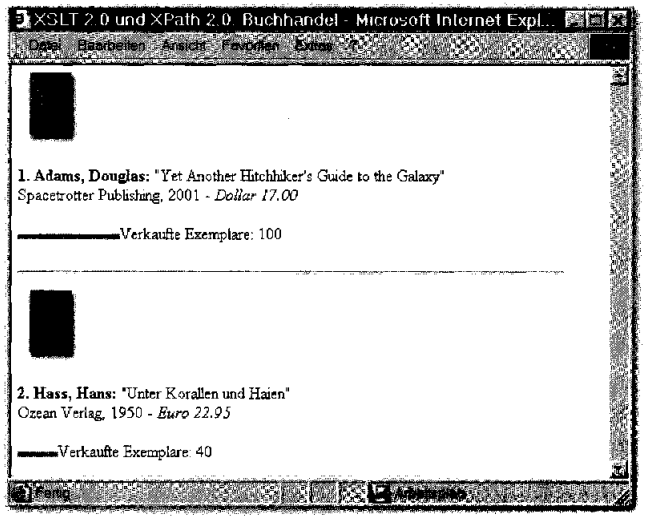

Abbildung 3.12 Buchdatei mit Balkendiagramm

- **Aufgabe 6:** Die Funktion `document()`
  Zur gleichen Datei von Aufgabe 5 soll ein weiteres Dokument hinzugenommen werden, das Inhaltsangaben zu den Büchern der Liste enthält. Diese

Inhaltsangaben sollen bei den jeweiligen Büchern eingefügt werden. Hierbei muss das externe Dokument mittels `document()` einbezogen und in eine Variable abgelegt werden.
Benötigte Dateien: *aufgaben/kap03/a6-buchdaten.xml, aufgaben/kap03/a6-buchdaten-inhalte.xml*

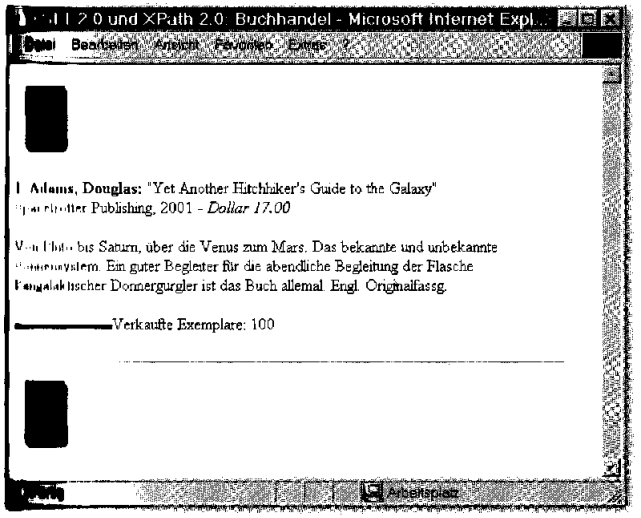

Abbildung 3.13 Die Buchdatei mit Kurzkritiken

3.12 Ausblick auf den zweiten Teil des Buches

Dies ist das Ende der allgemeinen Einführung in XSLT 2.0 und XPath 2.0, aber nicht das Ende dieses Buches. Dieses erste Viertel des Werks dient als grundlegender Überblick über die Konzepte von XSLT und XPath. Vollständigkeit wurde dabei weder angestrebt noch könnte sie in diesem Zusammenhang mit einem halbwegs folgerichtigen und gleichzeitig auch für Einsteiger zugänglichen Aufbau erreicht werden.

Viele Themen sind unberücksichtigt geblieben, darunter die Einbeziehung von Schema-Typen in die Verarbeitung, einige Funktionalitäten in Bezug auf Zeit- und Datumswerte und deren Verarbeitung, weitere Aspekte, die XSLT in Verbindung mit regulären Ausdrücken bietet, und zahlreiche andere gröbere und feinere Details. Es bleiben also weiße Flecken, die vom anschließenden Referenzteil ausgefüllt werden.