



Fachprüfung: Hyper- und Multimedia

Hypermedia- und Multimediasysteme (1,2)

19.07.2002

Name (bitte in Blockschrift)	
Matrikelnummer	
Unterschrift	

Hinweise:

- Überprüfen Sie Ihr Klausur-Exemplar bitte vor Beginn der Klausur auf Vollständigkeit.
- Bitte halten Sie Ihren Lichtbildausweis sowie den Studentenausweis zur Kontrolle bereit.
- Dauer der Klausur: 120 min.
- maximal erreichbare Punktzahl: 200
- Bitte füllen Sie das Deckblatt vollständig aus, beschriften jedes Blatt mit Ihrer Matrikelnummer und unterschreiben Sie dieses Klausur-Exemplar.
- Jedes Verlassen des Prüfungsraums muss ausdrücklich mit der Aufsicht vereinbart werden.
- Zugelassene Hilfsmittel sind ausschließlich Schreibutensilien, nicht-programmierbare Taschenrechner und das eigene(!) Gedächtnis.
- Bitte vermeiden Sie die Verwendung von roter Farbe.
- Die nach jeder Frage eingeklammerte Zahl ist die bei dieser Frage maximal erreichbare Punktzahl.
- Beachten Sie die in vielen Fragen enthaltenen Teilfragen!
- Falls der Platz für die Beantwortung einer Frage nicht ausreichen sollte, verwenden Sie bitte die Rückseite.
- Nutzen Sie im Falle von Unklarheiten hinsichtlich der Fragestellung die Möglichkeit zu Rückfragen!

Viel Erfolg!

Punkte Teil 1	
Punkte Teil 2	
Punkte gesamt	

1. Prüfer Note
2. Prüfer	



– I. Prüfungsfragen zur Veranstaltung im SS 2001 (HMS-1) –

1. Unterscheiden Sie Java-Applets und –Applikationen hinsichtlich der zum Ausführen erforderlichen Umgebung sowie der jeweiligen Möglichkeiten und Einschränkungen. (16)

- Möglichkeiten (Applikation) bzw. Einschränkungen (Applet):
 - File I/O (++)
 - Aufbau von Netzverbindungen (++) (außer Ursprung)
 - Definition nativer Methoden (++)
 - Laden von Bibliotheken (++)
 - Programmstart auf Client-Host (++)
 - Lesen von Systemeigenschaften (++)
- Umgebung Applikation: JVM auf OS-Basis (++)
- Umgebung Applet: JVM in HTML-Browser oder Appletviewer (+)
bei Aufruf über Applet-Tag in HTML-Seite (+)

2. Beschreiben Sie die „Multithreading“-Fähigkeit von Java-Programmen. Gehen Sie dabei besonders auf kritische Zustände (potentielle Konflikte) beim Zusammenwirken verschiedener Threads ein. (7)

- Multithreading: gleichzeitiger Ablauf verschiedener separater Programmteile. (+)
- Priorität: CPU-Zuteilung an gleichzeitig ablaufende Threads; gegenseitige Beeinflussung der Thread-Prioritäten möglich. (+++)
- Synchronisation: gleichzeitiger Zugriff auf gemeinsame (veränderliche) Speicherbereiche/Variablen. (+++)



3. Welche Zustände kann ein Thread im Laufe seiner Existenz einnehmen und wie können diese ineinander überführt werden? (12)

Zustände: (je +, max.4)

- newborn, runnable, running/blocked, dead

Überführung: (je ++, max. 8)

- <alle> → dead mit stop()
- newborn → runnable/running mit start()
- running → runnable mit yield()
- runnable/running → blocked mit suspend()/sleep()/wait()
- blocked → runnable mit resume()/notify()

4. Beschreiben Sie zwei grundsätzlich verschiedene Möglichkeiten, Threads in Java zu programmieren. Gehen Sie dabei auch besonders auf die Unterschiede ein. (10)

Thread-Klasse: (+)

- eigene Klasse extends Thread(++)
- run() Methode wird durch start() aufgerufen (++)

Runnable-Interface: (+)

- eigene Klasse implements Runnable(++)
- start() Methode muss programmiert werden (++)



5. Beschreiben Sie das Prinzip der Event-Verarbeitung in Java-Applets. Worauf muss bei der Programmierung besonders geachtet werden? Nennen Sie typische Beispiele von Events zur Anwender-Interaktion. (10)

- Events: Ereignisse, die i.d.R. durch User-Interaktion Programm-Aktivitäten auslösen. (++)
- Prinzip:
 - Methode übergibt Event-Objekt (aus `java.awt.event`) und Parameter (z.B. Koordinaten). (++)
 - Applet implementiert `EventListener` (gemäß Observer-Pattern) (++)
 - In `paint()` des Applets wird `EventListener` durch (z.B.) `addMouseListener(this)` für das Applet-Fenster aktiviert. (++)
- Beispiele: `mouseMove()`, `mouseUp()`, `mouseClicked()`, ... (++)

6. Nennen Sie drei verschiedene Java-Swing Components und ordnen Sie diese der Komponentenhierarchie von Swing zu. (6)

- Applet, Dialog, Frame, ... -- Top-Level Container (++)
- Panel, Scroll pane, Tabbed pane, Toolbar, ... -- General Purpose Container (++)
- Internal frame, Layered pane, Root pane, ... -- Special Purpose Container (++)
- [Button, Combo box, List, Menu, Slider, Text Field, ... -- Basic Control (++)]
- [Label, Progress bar, Tooltip, Color chooser, File chooser, Table, Text, Tree ... -- Info Display (++)]
- [allg. Zuordnung auch:
Top-level Container -- Intermediate Component -- Atomic Component]

7. Was unterscheidet Kompressionsverfahren der Redundanz- und der Irrelevanzreduktion?
- Nennen und beschreiben Sie je ein Beispiel (Huffman-Codierung ausgeschlossen). (10)

- Redundanzreduktion: verlustfrei, statistische Methoden (++)
- Irrelevanzreduktion: verlustbehaftet, physiologische Grundlage (++)
- Beispiel Redundanzreduktion:
Laufängencodierung (Angabe der Wiederholungszahl eines Wertes) od. Prädiktion (Beschreibung einer Signalfolge auf der Basis eines vorangegangenen Signals) (+++)
- Beispiel Irrelevanzreduktion:
Farbraumkonvertierung (RGB → YUV mit verringerter Chrominanz-Auflösung) od. DCT - Diskrete Cosinustranzformation (Transformation von Orts- in Frequenzraum zur Vorbereitung von Laufängencodierung) (+++)

8. Worin besteht das grundsätzliche Prinzip der Huffman-Kodierung? Auf welche Weise (und unter welchen Voraussetzungen) trägt diese zur Datenreduktion bei der JPEG-Bildkompression bei? Wie sähe eine solche Codierung konkret für die Zahlenfolge [5 5 0 3 3 3 3 2] aus? (12)

- Huffman-Kodierung: redundanzfreie präfixfreie Codes variabler Wortlänge in Abhängigkeit von Auftretenshäufigkeit. (+++)
- [Vorgehensweise: Wahl der 2 Werte geringster Wahrscheinlichkeit und Zuordnung je eines Wertes 0/1; Zusammenfassung dieser beiden Werte zu neuem Hilfssymbol und analoges Vorgehen mit nächstwahrscheinlichem Wert, bis alle Werte erfasst; dabei erhält der häufigste Wert nur ein einstelliges Symbol.]
- Datenreduktion durch kurze Wortlängen für häufig auftretende Werte; (+++) besonders effizient bei hoher Wiederholungszahl einzelner Werte. (+)
- [5 5 0 3 3 3 3 2] → [10101100000111] (+)
3 → 0 (+); 5 → 10 (+); 0 → 110 (+); 2 → 111 (+)



9. Beschreiben Sie Prinzip und Funktionsweise der Bewegungskompensation bei der MPEG-Kompression von bewegten Bilddaten. Wodurch wird hier zusätzlich Speicherplatz eingespart? Wo entsteht erhöhter Rechenaufwand? (10)

- Prinzip: bewegliche Bildelemente (+) werden in folgenden Frames an neuer Position wiedererkannt (+) und durch ihren Bewegungsvektor (++) kodiert.
- Zur Bewegungsabschätzung werden Bildblöcke betrachtet. (++)
- Gewinn: geringerer Prädiktionsfehler gegenüber regulärer bidirektionaler Kodierung. (++)
- Erhöhter Rechenaufwand entsteht nur auf Encoder-Seite. (++)

10. Beschreiben Sie kurz ein Konzept zur Verbreitung von Radio- und/oder TV-Inhalten über das Internet. Wählen Sie dafür ein konkret in Entwicklung bzw. Erprobung befindliches Beispiel, das die erweiterten Möglichkeiten des Internet gegenüber traditioneller Broadcast-Technologie nutzt. Welche Eigenschaft der aktuellen Internet-Technologie steht Broadcast-Anwendungen grundsätzlich im Wege? (7)

- z.B.: (++++)
 - TV-Anytime (Nutzung wachsender Speicherkapazitäten: Digitaler Videorecorder mit Festplatte statt Band; Programm-Agenten; ...)
- oder:
- [
- TeleWeb (Mini-WWW über traditionelles Fernsehen: ...)
 - tv-radio.com (Kombination von Radio und TV im Internet: ...)
 - 3D Webcast (Sportberichterstattung in 3D über WWW: ...)
-]
- Broadcast kann im Internet grundsätzlich nur als Multicast realisiert werden → Bandbreitenprobleme. (+++)

Punkte Teil 1	/ 100
---------------	-------



- II. Prüfungsfragen zur Veranstaltung im SS 2002 (HMS-2) -

1. Erstellen Sie ein eigenes Objekt in JavaScript. (10)

Beispiel:

```
function Farbe(Farbwert_R, Farbwert_G, Farbwert_B)
{
  this.Farbwert_R = Farbwert_R;
  this.Farbwert_G = Farbwert_G;
  this.Farbwert_B = Farbwert_B;
}

function TestObjekt()
{
  Test = new Farbe("33","99","C0");
  alert("Der Rotwert meiner Farbe ist Hexadezimal " +
  Test.Farbwert_R);
}
```

2. Auf welche (4) verschiedenen Arten können Cascading Stylesheets (CSS) in HTML Seiten verwendet werden? Lassen sich die unterschiedlichen Arbeitsweisen in Bezug auf ein HTML-Element kombinieren bzw. wie werden sich konkurrierende Anweisungen priorisiert? (10)

Inline über das Attribut Style auf Ebene des einzelnen Elementes,
Embedded im Style-Tag im Headbereich,

External über das Link-Tag als Verweis auf eine separate CSS-Datei
oder per Scripting.

Die genannten Formen lassen sich mischen bzw. kombinieren.

Dabei gilt folgende Priorität in absteigender Reihenfolge bei sich
widersprechenden Anweisungen: Scripting > Inline > Embedded >
External.



3. Was ist der Unterschied zwischen „wellformed“ und „validated“ XML-Dokumenten? Welchen Umständen muss ein XML-Dokument genügen um wellformed bzw. valide zu sein? (5)

„wellformed“-Dokumente sind formal richtig aufgebaut (z. B. schliessende Tags, Gross-/Kleinschreibung, Nesting) während „validated“-Dokumente im inhaltlichen Aufbau gegen eine Dokumentenvorlage (DTD oder Schema) geprüft sind.

4. Was sind die Vorteile von Schemas gegenüber DTD's? (5)

Schemas werden selbst auch in XML geschrieben, Element- und Attributinhalt lassen sich näher typisieren. Das Contentmodell lässt sich gezielter, aber auch flexibler beschreiben.

5. Was sind die Unterschiede zwischen HTML und XHTML? (5)

Für XHTML gilt u. a.:

- case-sensitive, schließende Tags,
- geordnetes Nesting,
- Attributwerte in Anführungszeichen,
- Script- und Styleblöcke in CDATA-Abschnitten.

6. Wie können Sie ab dem MS Internet Explorer 5.x eigene Tags definieren? (10)

Über die Definition eines Namespace „xmlns“ im HTML-Tag und der Deklaration entsprechender Styles für die innerhalb des Namespace vorkommenden Tags

```
<HTML xmlns:FH="urn:FH">
  <HEAD>
    <STYLE>
      @media all{FH\;SEMESTER{COLOR:RED;}}
    </STYLE>
  </HEAD>
  <BODY>
    <FH:SEMESTER>6. Semester</FH:SEMESTER>
  </BODY>
</HTML>
```

7. Wie lassen sich Datenbankabfragen mit ADO weiterverarbeiten? (10)

Die Ergebnisse von SQL-Abfragen über ADO sind zunächst in einer temporären Tabelle enthalten („recordset“).

Die Inhalte können in einer („for-while“) Schleife für die einzelnen Datensätze sowie einer Schleife über die Feldinhalte ausgelesen und weiterverarbeitet werden.

Der Inhalt kann aber auch als XML-Stream („recordset.Save(xml, adPersistXML“) oder als String („Variant = recordset.GetString(StringFormat, NumRows, ColumnDelimiter, RowDelimiter, NullExpr“) ausgegeben werden.



8. Was ist der Unterschied zwischen einem DOM- und einem SAX- XML-Parser. Wo liegen die jeweiligen Vor- und Nachteile, wo die Einsatzgebiete? (5)

Simple-API-for-XML-Parser arbeiten zeilenorientiert und sind (wesentlich) weniger speicherintensiv und schneller bei der Analyse und Verarbeitung großer XML-Dokumente. Dokumenten-Objekt-Modell-Parser dagegen bauen zunächst im Speicher das gesamte DOM auf und ermöglichen dann gezieltere Zugriffs- und Manipulationsmechanismen auf XML-Dokumente.

9. Beschreiben Sie die Funktion von XPath und die wichtigsten Anwendungsprinzipien. (10)

Xpath ist eine Abfragesyntax, mit deren Hilfe sich Elemente eines XML-Dokumentes genau lokalisieren lassen. Xpath ist damit das SQL für XML. Ein wesentlicher Bestandteil ist das „/“-Zeichen, mit dem ein Knotenpunkt symbolisiert wird. Der Platzhalter „..“ bezeichnet den gerade gültigen Knoten. „../“ kennzeichnet somit Knoten die dem gerade gültigen in der nächsten Hierarchiestufe innerhalb des DOM's folgen. „//“ bezeichnet Knoten, die vom aktuellen ausgehend beliebig tief verschachtelt liegen dürfen. Mit „@“ werden Attribute gekennzeichnet. In eckigen Klammern stehende Ausdrücke dienen der Filterung nach bestimmten Kriterien. Diese können auch logische Vergleichsoperatoren enthalten.

Die auch als „Patterns“ bezeichneten Abfrage-Strings können in XSLT oder in bestimmten DOM-Methoden („selectNodes (.AUSDRUCK)“, „selectSingleNode (.AUSDRUCK)“) des XML-Parsers Verwendung finden.



10. Nennen Sie die Standardobjekte eines Internet Information Servers und deren Funktion. Wie können weitere Objekte serverseitig instanziiert werden? (10)

- Application : Verwaltung von Zuständen in einer Multiuserumgebung
- ObjectContext : Zusammenarbeit mit dem MTS
- Request : Auswertung vom Client übermittelter Angaben
- Response : Senden von Ausgaben an den Client
- Session : Verwaltung von Zuständen im Rahmen einer Benutzersitzung
- Server : Verwaltung des Webservers
- Mit der Methode `Server.createObject()` werden weitere Objekte instanziiert.

11. Was ist der Unterschied zwischen „innerText“ und „innerHTML“? (5)

Mit „innerText“ werden nur rein textuelle Elementinhalte gelesen oder geschrieben, über „innerHTML“ zusätzlich alle innerhalb des betreffenden Tags ggf. weiter enthaltenen Elemente und deren Attribute.

12. Was bedeutet CDATA bzw. PCDATA, was ist der Unterschied? (5)

Parseable bzw. Non-Parseable Character Data, also Stringinformationen, die geparkt oder nicht geparkt werden.



13. Wie lassen sich innerhalb von JavaScript globale Variablen deklarieren? Wo sind deren Grenzen? (5)

Globale Variablen werden außerhalb von Funktionen deklariert. Sie gelten nur innerhalb der Seite, in der sie in einem Scriptabschnitt deklariert sind, d.h. z.B. für benachbarte Frameseiten sind sie nicht unmittelbar gültig.

14. Nennen Sie 5 Beispiele für Collections in DHTML? (5)

all; anchors; applets; areas; cells; children; elements; embeds; filters; forms; frames; images; imports; links; options; plugins; rows; rules; scripts; styleSheets; tbodies.

Punkte Teil 2	/ 100
---------------	-------

Viel Erfolg!