



Fachprüfung: Hyper- und Multimedia

Hypermedia- und Multimediasysteme (1,2)

02.02.2004

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 2002 (HMS-1) –

1. Nennen Sie typische (allgemeine) Eigenschaften von Scriptsprachen für HTML. Beschreiben Sie die jeweiligen Vorzüge von Client- und Server-seitigen Sprachen. Ordnen Sie Perl, JavaScript, PHP und ASP den beiden Gruppen zu. (15)

typische Eigenschaften:

- interpretative Ausführung (++)
- kaum bis keine Typisierung (+)
- keine komplexen Datenstrukturen (+)
- integrierte Betriebssystem-Kommandos (+)
- unterstützte String-Manipulationen (+)
- einbinden externer Programme (+)
- ...

Client-seitig:

- kein Verbrauch von Server-Ressourcen (+)
- geringere Bandbreite der Leitung erforderlich (+)
- bessere Datensicherheit (+)
- offline möglich (+)
- Bsp.: JavaScript (+)

Server-seitig:

- keine Abhängigkeit von Client-Ressourcen (+)
- Datenaustausch möglich, inkl. Datenbank-Anbindung (+)
- Verbergen von Quellcode einfacher (+)
- Plattform-unabhängig (+)
- Bsp: Perl, PHP, ASP (+++)

2. Woran erkennt ein HTML-Browser, wenn eine eingebundene externe Datei durch den JavaScript-Interpreter zu bearbeiten ist? (4)

Im **<script>** Tag (++) wird die Sprache über das Attribut **language** bzw. **type** (++).

3. Wie wird JavaScript-Code als Markup (direkt) in HTML-Dokumente eingebunden? Formulieren Sie ein konkretes Beispiel mit vollständigem HTML-Code für die Definition einer Variablen und deren Ausgabe als im Browser zu rendernder Text. (12)

- als Markup im Script-Tag: (++)
- Beispiel: (10)

```
<html><head></head>
<body>
  <script Language="JavaScript" type="text/JavaScript">
    var brmpf = 1;
    document.write("Wert: "+brmpf+"\n");
  </script>
</body></html>
```

4. Wie werden in JavaScript mehrdimensionale Arrays erzeugt und initialisiert? (6)

1. Erzeugung eines eindimensionalen Arrays über Konstruktor:
`new Array()` (++)
2. Funktionsdefinition: `function ...() {}`; (++)
enthält geschachtelte `for`-Schleifen mit jeweils neuen Konstruktor-Aufrufen
3. Initialisierung durch Funktionsaufruf (++)



5. Beschreiben Sie die Kommunikationsbeziehungen beim Aufruf eines Perl-Skripts über die CGI-Anbindung durch einen Web-Client. (10)

- Client fordert beim Webserver per POST Request eine Aktion ein. (++)
- Webserver ruft über CGI / I SAPI (++) mittels Stdin das Perl-Programm lokal auf. (++)
- Das Perl-Programm sendet seine Ausgabe (z.B. HTML-Text) über Stdout an den Webserver. (++)
- Der Webserver leitet den Perl-Output weiter an den Client. (++)

6. Wodurch unterscheidet sich die Syntax zur Einbindung von PHP-Skript-Kommandos zwischen direkter Einbindung und Einbettung in ASP? (4)

- direkt: `<? code?>` bzw. `<?php code ?>` (++)
- ASP: `<% code %>` (++)

7. Unter welchen Voraussetzungen kann ein Java-Quellcode (mit grafischer Ausgabe) sowohl als Applet als auch als Java-Applikation verwendet werden? (9)

- `main()`-Methode --> Applikation (++)
- `import java.applet.*;` --> Applet (++)
- Programm-Klasse als Erweiterung der Klasse Applet (++)
- `import java.awt.Graphics;` --> grafische Ausgabe (+)
- Graphics-Objekt in Methode `paint()` für Browser-Fenster --> Applet (++)



8. Erläutern Sie das Prinzip der Prioritätenvergabe bei „Multithreading“ in Java-Programmen. Welche Methoden werden bei der dynamischen Prioritätskontrolle eingesetzt? (8)

- Prioritäten: CPU-Zuteilung an gleichzeitig ablaufende Threads; (++)
- verschiedene Prioritätszustände möglich; (++)
- gegenseitige Beeinflussung der Thread-Prioritäten (und Laufzustände, d.h. auch unterbrechen und reaktivieren) möglich (++)
- Methoden: setPriority(), sleep(), yield(), run(), stop(), suspend(), resume(). (++)

9. Unter welchen Umständen ist es nötig, einen Java-Thread als Runnable Interface zu implementieren? Wie unterscheidet sich diese Methode grundsätzlich von der Verwendung der Klasse Thread? (12)

Java erlaubt keine Mehrfachvererbung. (++) Wenn die betreffende Klasse also bereits von einer anderen abgeleitet werden soll, kann die Thread-Funktionalität nur durch die Implementierung des Runnable-Interfaces erreicht werden. (++)

Thread-Klasse:

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

Runnable-Interface:

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



10. Nennen Sie vier verschiedene Java Layout Manager und beschreiben Sie kurz deren Eigenschaften. (12)

- FlowLayout (+) – Anordnung der Elemente zuerst horizontal, dann vertikal (++)
- BorderLayout (+) – Anordnung der Elemente nach Ortsangabe [Center, North, West, East, South] (++)
- GridLayout (+) – Anordnung der Elemente nach Reihenfolge des Auftretens innerhalb eines vorgegebenen Gitters mit gleichmäßig großen Zellen (++)
- CardLayout (+) – Anordnung der Elemente hintereinander in z-Ebene mit Karteireitern (++)
- GridBagLayout (+) – Anordnung der Elemente innerhalb eines frei konfigurierbaren Gitters (++)
- BoxLayout (+) – Anordnung der Elemente nach Reihenfolge des Auftretens untereinander (++)

11. Beschreiben Sie das Konzept der keyframe Animation („tweening“) in Macromedia Director oder Flash. Welche Eigenschaften lassen sich auf diese Weise animieren? (8)

- Einzelne Bilder werden als *Sprites* in bestimmtem Zustand bestimmten Zeitpunkten (*frames*) zugeordnet, zu denen sie Schlüsselzustände annehmen (--> *keyframes*). (++)
- Für die zwischen den *Keyframes* liegenden Einstellungen werden die jeweiligen Zustände (für jeden *frame*) linear interpoliert und so eine kontinuierliche Veränderung erreicht. (++)
- Folgende Eigenschaften von *Sprites* können „getweent“ werden: Position, Orientierung, Größe, Form, Farbe, ... (++++)



- II. Prüfungsfragen zur Veranstaltung im SS 2003 (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. 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>
```




6. 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.

7. 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.



8. 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.

9. Was ist der Unterschied zwischen „innerText“ und „outerText“? (5)

Bei lesendem Zugriff verhalten sich „innerText“ und „outerText“ gleich. Schreibend wird bei „outerText“ das gesamte Element mit dem neuen Text ersetzt. Bei „innerText“ bleibt das Element mit allen Attributen selbst erhalten, nur der Inhalt wird durch den neuen Text ersetzt.

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

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



11. 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.

12. 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; tbody.

13. Was ist der Unterschied zwischen den CSS-Anweisungen „style.display=none“ und „style.visibility=hidden“? (5)

Beide Anweisungen führen dazu, dass das entsprechende Element nicht auf der Seite angezeigt wird mit dem Unterschied, dass bei „display=none“ der nachfolgende HTML-Streaminhalt neu positioniert wird, während bei „visibility=hidden“ keine Neupositionierung der übrigen Elemente erfolgt.



14. Welche Aufgaben erfüllt das Textrange-Objekt in DHTML? (5)

Das Textrangeobject dient umfangreicheren Operationen mit Text im HTML-Stream: Text in einem bestimmten Kontext (Element oder Position) zu identifizieren; Text im Dokument zu suchen, selektieren und ersetzen; Text in logische Einheiten zu zerteilen; Text und HTML zu manipulieren.

15. Was versteht man unter Event-Bubbling, wie lässt es sich beeinflussen, gilt es für alle Events? (5)

Unter Event-Bubbling nennt man den bei einigen Events, wie z. B. den Mausevents, vorhandenen Mechanismus, dass das Event in der Elementhierarchie aufsteigend mehrfach ausgelöst wird. Er kann durch das Setzen der Property „cancelBubble=true“ unterbrochen werden. Event-Bubbling ist ein Teil des Internet Explorer Event-Modells und gilt nur für einige Events.

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

Viel Erfolg!