



Fachprüfung: Hyper- und Multimedia

Hypermedia- und Multimediasysteme (1,2)

13.02.2001

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.
- Bevor Sie zur Bearbeitung der Teilklausur für Teil 2 zusätzliche Hilfsmittel einsetzen, muss das vorliegende Teilklausur-Exemplar (Teil 1) abschließend bearbeitet und abgegeben werden!
- 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!

Note	
1. Prüfer	
2. Prüfer	



- I. Prüfungsfragen zur Veranstaltung im SS 1999 (HMS-1) -

1. Welchen Typ eines Filteralgorithmus würden Sie zur Glättung eines stark verrauschten Bildes bei minimalem Rechenaufwand wählen? Wie würden Sie einen Gauß-Filter konfigurieren, um regionale Extrema weitgehend zu erhalten? – Begründen Sie jeweils Ihre Entscheidung, ggf. durch Vergleich mit anderen Verfahren. (18)

- Mittelwertfilter (++) glättet mit minimalem Rechenaufwand: Summation und eine Division pro Pixel (++++) (gegenüber Gauß (+) mit zusätzlicher Berücksichtigung von Wichtungsfaktoren (++) und Medianfilter (+) mit aufwendiger Sortierung (++)).
- Gauß-Filter erhält regionale Extremwerte am besten, wenn die Umgebung weiträumig (mehrere Nachbarschaftsringe) einbezogen (+++) und mit zunehmendem Abstand geringere Gewichtung (+++) verwendet wird.

2. Worin besteht die Aufgabe von Bildsegmentierungs-Verfahren? Nennen Sie eine typische Anwendung. Unter welchen Voraussetzungen sind Gebietswachstums-Methoden Schwellenwert-Verfahren vorzuziehen? (9)

- Segmentierung: Trennen mehrerer zusammenhängender Bildbereiche voneinander (+++)
- typisches Beispiel: Knochen und Organe in Röntgenbildern (++)
- Bei stark inhomogenen Bildern (++) fassen Gebietswachstums-Verfahren zusammengehörige Bereiche besser zusammen als starre Schwellenwerte, weil die charakteristischen Absolutwerte in verschiedenen Bildbereichen deutlich differieren können. (++)



3. Durch welche drei grundsätzlichen Bestandteile (abgesehen von Materialeigenschaften) ist ein einzelnes Polygon einer 3D-Oberfläche in gängigen Dateiformaten i.d.R. repräsentiert? Unter welchen Voraussetzungen kann auf eine davon verzichtet werden? (12)

- Vertexpunkte (++) , Verbindungslinien (++) , Normal-Vektoren (++)
- In geschlossenen Flächen sind Außen- und Innenseite eindeutig definiert, weil bei jeder Ansicht jede Innenseite von einer Außenseite verdeckt wird. Alternativ kann auch durch Angabe des Drehsinns über die Vertex-Reihenfolge die Richtung der Normalen eindeutig definiert werden. (++++) In beiden Fällen erübrigt sich die Angabe von Normalenvektoren. (++)

4. Erläutern Sie die Metapher einer „Bühne mit Darstellern“ für interaktive Multimedia-Anwendungen aus der Sicht Objekt-orientierter Programmierung. Wie wird der lineare Ansatz der Metapher durchbrochen? (20)

- importierte Medien-Objekte (Darsteller) (++) (z.B. Grafiken, Animationen, etc.) als "Klassen" (++)
- Instanzen der Medien-Objekte (Sprites) (++) als "Objekte" (++)
- Verhalten (++) der Sprites als "Methoden" (++)
- Eigenschaften (++) von Sprites (z.B. Farbe) und Objekten (z.B. Variablen-Werte) (++)
- Nicht-Linearität durch Stop-/Warte-Punkte (++) an Szenenenden sowie Ein- und Aussprungsmarken (++) für Querbezüge.

5. Wie kommt die Plattform-Unabhängigkeit von Java-Programmen zustande? (10)

- vor-kompilierter Byte-Code (++) als plattform-unabhängige (+) Zwischenform, die von plattform-abhängiger Version (+) der Java Virtual Machine (++) (auf dem ausführenden Rechner installiert (+)) interpretiert wird (++)

6. 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 (+)

7. Wozu dient „double buffering“? Wie unterscheiden sich diesbezüglich die Java-Grafikumgebungen AWT und Swing? (7)

- double buffering: Bildaufbau im Hintergrund (Bildschirmpuffer) und Austausch gegen aktive Bilddarstellung nach Fertigstellung (++++)
- In AWT muss *d.b.* extra programmiert werden, in Swing ist es als Standard enthalten (+++)

8. Beschreiben Sie die „Multithreading“-Fähigkeit von Java-Programmen. Gehen Sie dabei besonders auf kritische Zustände (potentielle Konflikte) beim Zusammenwirken verschiedener Threads ein. (8)

- Multithreading: gleichzeitiger Ablauf verschiedener separater Programmteile. (++)
- Priorität: CPU-Zuteilung an gleichzeitig ablaufende Threads. (+++)
- Synchronisation: gleichzeitiger Zugriff auf gemeinsame (veränderliche) Speicherbereiche/Variablen. (+++)



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

– II. Prüfungsfragen zur Veranstaltung im SS 2000 (HMS-2) –

1. Wie kann eine objekt-orientierte Betrachtung von HTML-Dokumenten für dynamische Erweiterungen genutzt werden? (9)

- DOM: individuell adressierbare Tags (+++)
- Veränderung von Tag-Attributen und/oder -Inhalten zur Laufzeit (+++)
- Zuordnung von Eigenschaften, Ereignissen und Methoden an einzelne Tags (+++)

2. Weisen Sie dem folgenden HTML-Tag unter Verwendung eines einzeiligen DHTML-Konstrukts einen neuen Inhalt zu, so dass der Text „neu“ anstelle des Textes „alt“ und fett statt schräg gedruckt erscheint. Zeigen Sie zwei verschiedene Möglichkeiten unter Verwendung von „test1“ bzw. „test2“ auf. (10)

```
<div id="test1"><i id="test2">alt</i></div>
```

- test1.innerHTML = "<b id="test2">neu</i>" (+++++)
- test2.outerHTML = "<b id="test2">neu</i>" (+++++)

3. Formulieren Sie die hierarchische Struktur
- (a) des dritten <APPLET>-Tags eines HTML-Dokuments und
 - (b) der Inline-Grafik mit ID="graf" im sechsten Frame im übergeordneten Frameset der aktuellen HTML-Seite in JavaScript. (6)

- (a) window.document.applets[3] (+++)
- (b) parent.frames[6].graf (+++)



4. Beschreiben (und begründen) Sie ein typisches Anwendungsszenario für die Verwendung von HTML Style Sheets. Gehen Sie dabei besonders auf den Aspekt des Kaskadierens ein. Unter welchen Umständen empfiehlt es sich nicht, Style Sheets einzusetzen? (20)

- Anwendungsszenario: Ein Dokument wird in mehreren Varianten angeboten, die sich nicht inhaltlich unterscheiden, aber für verschiedene Ausgabemedien vorgesehen sind. (+++++) Die HTML-Datei enthält nur Stildefinitionen, die für alle Varianten gelten sollen. (++) Differierende Stildefinitionen sind in (ggf. mehrere verschachtelte) externe Dateien ausgelagert. Aufgerufen werden diese im HTML-Header. Verzweigung erfolgt durch parallele Verzeichnisse, welche die HTML-Datei idealerweise durch symbolische Links aus einer einzigen Originalquelle holen. (+++)
- Begründung durch Vorteile: Trennung von Form und Inhalt; erleichterte Pflege des Dokuments durch einmalige Änderungen in HTML-Datei. (++++)
- Kaskadierung: Layout-Definitionen an verschiedenen Stellen, auf verschiedenen Ebenen; keine Gefahr undefinierter Layout-Elemente; Änderungsmöglichkeiten auf mehreren Ebenen. (+++)
- keine Style Sheets wenn mit älteren (od. allg. nicht CSS-fähigen) Browsern gerechnet werden und ein bestimmtes Erscheinungsbild sichergestellt werden muss. (+++)

5. Warum ist Server-seitiges Scripting Browser-unabhängig? Welche weiteren Argumente begünstigen den Einsatz von Server-seitigem Scripting? (10)

- Auf Client-Seite kommt nur reiner HTML-Code an. (++++)
- Performance ist Client-unabhängig (++)
- Verbergen von Algorithmen (++)
- I/O auf Server-Seite kontrolliert möglich (++)



6. Sie sollen auf einem Unix-Rechner einen Web-Server mit Server-seitigen Skripten einrichten. Verwenden Sie CGI oder ASP? Begründen Sie Ihre Entscheidung. (4)

- CGI (+), weil ASP nur für Microsoft-Server möglich ist. (+++)

7. Wann ist ein XML-Dokument „wellformed“, wann „valid“? Was ist zusätzlich erforderlich, um ein XML-Dokument zu rendern? (13)

- wellformed: (max. 5)
 - Markup (+)
 - Anfang/Ende Tags (+)
 - Bezeichner (+)
 - Attribute mit Values (+)
 - Inhalt: parsable data (PCDATA) (+), non-parsable data (CDATA) (+)
 - Reguläres Nesting (+)
- valid:
 - wellformed (++)
 - DTD oder Scheme (++)
- render-fähig:
 - XML-fähiger Browser, ggf. spez. JavaScript zur XML-Darstellung (+)
 - Stil-Definition mit Style Sheet oder XSL (+++)



8. Formulieren Sie eine XML-DTD für den HTML-Tag `<TABLE>`. Berücksichtigen Sie auch die in die `<TABLE>`-Umgebung einschließbaren Tags `<TR>` und `<TD>`, die beide mindestens einmal vorkommen sollen. Im `<TD>`-Tag soll das Attribut "align" mit den Werten "left", "center" oder "right" möglich sein, wobei "center" als default-Wert festgelegt wird. Definieren Sie außerdem den HTML-Ersatzvorschrift für das Zeichen "ß". (14)

- `<!ELEMENT TABLE (TR+)>` (++)
`<!ELEMENT TR (TD+)>` (++)
`<!ELEMENT TD (#PCDATA)>` (++)
`<!ATTLIST TD`
`align (left|center|right) "center" >` (++++)
- `<!ENTITY (++) szlig "ß">` (++)

9. Nennen Sie die grundsätzlichen Unterschiede zwischen XHTML und HTML 4.0. Welche davon führen nach HTML 4.0-Standard zu fehlerhaftem Rendering oder Fehlermeldungen? – Nennen Sie dafür ein konkretes Beispiel. (14)

- Kleinschreibung der Tags (++)
- Schließen jedes geöffneten Tags (++)
- Anführungszeichen bei Attributzuweisungen zwingend (++)
- korrektes Nesting verpflichtend (++)
- XHTML erlaubt bei einzeln vorkommenden Tags (+) Öffnen und Schließen innerhalb einer Klammer (++); Bsp.: `
` (++); dies entspricht nicht dem HTML 4.0-Standard. (+)

Punkte Teil 2

/ 100

Viel Erfolg!



Punkte Teil 1	
Punkte Teil 2	
Punkte gesamt	
Note	