Klausur Web Engineering Test 1 (188.98 Gruppe A	51).	18. April 2016
MatrNr.:	Nachname: _	
Studienkennzahl:	Vorname:	

Erreichte Punkte: \_\_\_\_\_

## Hinweise zur Beantwortung:

- Die Klausur umfasst offene Fragen, die grafisch/textuell zu beantworten sind, und/oder Fragengruppen, die jeweils mit ja oder nein beantwortet werden müssen (das entsprechende Kästchen □ ankreuzen).
- Bewertung der Ja/Nein-Fragen:
  - korrekt beantwortet ⇒ 1 Punkt
  - -falsch beantwortet  $\Rightarrow$ 1 Punkt Abzug, aber pro Fragengruppe gibt es mind. 0 Punkte
  - nicht beantwortet ⇒ 0 Punkte
- Sie haben 30 Minuten Zeit.

Frage 1. Wofür steht WAI-ARIA und welchen Zweck hat es? Nennen Sie die drei der fünf Regeln, ARIA in HTML zu nutzen. (8 Punkte)

WAI-ARIA stands for Web Accessibility Initiative - Accessible Rich Internet Application. It provides additional semantic markup for dynamic web applications using roles for missing elements and providing custom states and properties.

The five rules of using are

Use native HTML element if possible

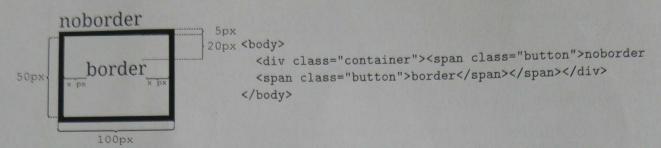
Do not change native semantics if possible

Interactive ARIA controls must be usable with keyboard

Do not use role presentation or aria-hidden on focusable element

All interactive elements must have an accessible name

Frage 2. Bitte definieren Sie ein CSS, das die angegebene Spezifikation bei gegebenem HTML-body erfüllt und maximal die Attribute display, border, padding-\*, margin-\*, width, height und text-align verwendet. (9 Punkte)



.button .button { display:block; border:5px solid black; padding-top:20px; width:90px; height:50px; text-align:center; }

Frage 3. Welche der folgenden Aussagen bezüglich HTTP sind korrekt? (5 Punkte)

Der Statuscode 200 zeigt die erfolgreiche Requestausführung an.	⊠ja
Bei HTTP handelt es sich um ein zustandsloses Protokoll.	⊠ja
In HTTP haben erfolgreiche GET requests den RESPONSE status.	⊠ nein
HTTP nutzt die TERMINATE-methode, um Websockets zu schließen.	⊠ nein
HTTP erlaubt, dass sich der Serverzustand durch GET-Anfragen ändert.	⊠ nein

Frage 4. Wofür steht die Abkürzung MVC? Beschreiben und skizzieren Sie das Zusammenspiel der verschiedenen Komponenten in MVC. (8 Punkte)

Model Hält Daten und eventuell Applikationslogik. Umsetzung in Java: Java Beans

View Zeigt Modelldaten und -zustand an. Kann durch Interaktionen Aktualisierung der Daten bewirken. Umsetzung in Java: JSP, JSF.

Controller Reagiert auf Requests und generiert Responses. Browser interagiert nur mit Controller. Umsetzung in Java: Servlet.

Frage 5. Welche der folgenden Aussagen sind korrekt? (5 Punkte)

HTTP-Request-Parameter können in Servlets mit der getAttribute-Methode ei-		⊠ nein
nes HTTPSession-Objekts abgerufen werden.		
Das flex-direction-Attribut in CSS definiert die Hauptachse des flex-containers.	⊠ ja	
Web Accessibility ist nur ein Rechtsthema.		⊠ nein
In HTML5 sollten die Elemente header, navigation und footer statt div ver-		⊠ nein
wendet werden, um mehr Bedeutung zu transportieren.		
JavaScript implementiert den ECMAScript-Standard.	⊠ ja	

Frage 6. Nummerieren Sie die input-Elemente in den [] aufsteigend nach dem ersten Mal, bei dem das checked-Attribut wahr wird, wenn der untenstehende JavaScript-Code ausgeführt wird. Nutzen Sie dieselbe Zahl für Elemente, deren checked-Attribut durch das gleiche Statement wahr wird und  $\neg$ , wenn es immer falsch bleibt. (10 Punkte)

## HTML-Code:

```
<html xmlns="http://www.w3.org/1999/xhtml">
 <script type="text/javascript" src="jquery.js" ></script>
 </head>
 <body>
  <form method="POST" action="bla" class="input" id="div" >
   <input type="checkbox" class="checked" id="input" /> [ ) ]
   <input type="checkbox" class="input form" id="checked" /> [ ~ ]
   <input type="checkbox" class="form" id="form" /> [ ] 
   <div class="form">
    <input type="checkbox" id="class" /> [ ( ]
    <input type="checkbox" class="checked" id="body" /> [ < ]</pre>
   </div>
  </form>
 </html>
JavaScript:
 $(document).ready(function() {
xmlHttp = new XMLHttpRequest();
xmlHttp.open("GET","http://example.com",true);
xmlHttp.onreadystatechange = function() {
 if (xmlHttp.readyState == 4) {
   $("form>input").prop("checked",true);
   $("form input").prop("checked",true);
 7
};
xmlHttp.send(null);
$(".input").prop("checked",true);
$("#div+input").prop("checked",true);
$("#checked+input").prop("checked",true);
});
```