Klausur zur Vorlesung
„Einführung in Verteilte Systeme“ WS 05/06
Prof. Dr. Odej Kao
3. Februar 2006

Aufkleber

Name:	**Vorname:**
Matrikel:	**Studiengang:** inkl. DPO4, B/M, Schwerp., …

- *Schreiben Sie zunächst Ihren Namen <u>oder</u> Matrikelnummer auf **jedes Blatt** der Klausur!*
- *Lassen Sie die Klausur zusammengeheftet!*
- *Die Klausur dauert 75 Minuten und umfasst 6 Aufgaben auf 11 Seiten.*
- *Die Klausur ist bestanden, wenn mindestens 50% der Punkte erreicht wurden.*
- *Es sind keine Hilfsmittel zugelassen (insbesondere keine Taschenrechner und Handys)!*
- *Abschreiben und Abschreibenlassen führt zum Nichtbestehen der Klausur!*
- *Benutzen Sie kein eigenes Konzeptpapier bzw. Schmierpapier. Sie bekommen bei Bedarf Papier von der Klausuraufsicht!*
- *Wenn Sie Konzeptpapier abgeben wollen, dann nur mit Namen und Matrikelnummer!*
- *Kennzeichnen Sie Ihre Lösung eindeutig. Es wird keine Lösung gewertet, wenn Sie zu einer Aufgabe mehr als eine Lösung abgeben.*
- *Bei Multiple-Choice-Fragen führt eine falsche Antwort (d.h. falsches Ankreuzen) zu Punktabzug innerhalb der entsprechenden Teilaufgabe! (Minimal 0 Punkte pro Teilaufgabe)*
- *Benutzen Sie keine rotfarbigen Stifte oder Bleistifte.*

Aufgabe	1	2	3	4	5	6	Σ
Punkte	**12**	**14**	**8**	**8**	**19**	**14**	**75**
Erreicht							

Aufgabe 1: Grundlagen (4+4+4=12 Punkte)

a) Durch den Einsatz von Verteilten Systemen ergibt sich ein erhöhter Nutzwert in vier großen Bereichen der IT Infrastruktur und der Organisation von IT Prozessen. Benennen Sie diese Bereiche.

b) Geben Sie die grundlegenden Operationen bei der Fehlerverarbeitung in Verteilten Systemen an.

c) Erläutern Sie den Begriff „Transparenz“ und geben Sie zwei Beispiele (Nennung reicht aus) für Transparenz bei Verteilten Systemen an.

Aufgabe 2: Middleware und Architektur (6+4+4=14 Punkte)

a) Ein Unternehmen verfügt über sechs dedizierte Videoserver. Drei dieser Server liefern Videos in hoher Qualität, die restlichen drei Server liefern Videos in niedriger Qualität. Skizzieren Sie eine Kette von erweiterten Client/Server Elementen, so dass die Nutzer mit hoher bzw. niedriger Bandbreite zu der entsprechenden Servergruppe geleitet werden und die Last auf die Server innerhalb der Gruppe gleichmäßig verteilt wird.

b) Im Auftrag einer Firma sollen Sie ein Buchungssystem für Konzertkarten entwickeln. Skizzieren Sie die Komponenten der entsprechenden 3-Tier-Architektur und tragen Sie in jede Komponente ein, welche Daten sich darin befinden und welche Operationen von dieser Komponente ausgeführt werden.

c) Beschreiben Sie die Kernaufgaben einer Middleware für Verteilte Systeme.

Aufgabe 3: R/R-Protokolle und Web Services (4+4=8 Punkte)

a) Nennen Sie die aktuellen Standards, die zur Auffindung, Beschreibung, zum Datenaustausch und zum Aufruf von Web Services benutzt werden. Beschreiben Sie die Funktion der entsprechenden Standards jeweils in einem Satz.

b) Zeichnen Sie den Ablauf einer auf Datagram-Sockets basierenden Kommunikation.

Aufgabe 4: Entfernte Schnittstellen (4+4=8 Punkte)

a) Entfernte Schnittstellen und entferne Objektreferenzen sind die grundlegenden Konzepte des verteilten Objektmodells. Definieren Sie diese beiden Konzepte.

b) Erläutern Sie das Benachrichtigungskonzept, wenn sich das relevante Objekt außerhalb des Ereignisdienstes befindet.

Aufgabe 5: Programmiermodelle (5+5+4+5=19 Punkte)

a) Schreiben Sie ein Java-RMI-Interface zu dieser Interfaceimplementierung.

Datei: PersonenImpl.java

```
import java.rmi.*;
import java.rmi.server.*;
import java.io.*;

public class PersonenImpl extends UnicastRemoteObject
  implements PersonenInterface {

     String Vorname, Nachname;
     Integer Alter;

     public void setName(String Vorname, String Nachname)
        throws RemoteException {
          this.Vorname = Vorname;
          this.Nachname = Nachname;
     }
     public String getInitialien() throws RemoteException {
          return (Vorname.toCharArray())[0] + ". "
             + (Nachname.toCharArray())[0] + ".";
     }
     public NameUndAlter getNameUndAlter()
        throws RemoteException, IOException {
          return new NameUndAlter(this.Nachname);
     }
}
```

b) Mittels einer SOAP-Nachricht soll der folgende AXIS Web Service aufgerufen werden. Vervollständigen Sie die fehlenden Ausdrücke an den markierten Stellen.

Datei: Adder.jws

```
public class Adder {
     public int intsum (int adam, int eve) {
          return adam + eve;
     }
}
```

SOAP-Nachricht:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

     <soapenv:[          ]>
          <[          ] xmlns="http://localhost:8080/axis/[          ]>
               <[          ] xsi:type=[          ]> 4 <[          ]>
               <[          ] xsi:type=[          ]> 8 <[          ]>
          <[          ]>
     </soapenv:[          ]>

</soapenv:Envelope>
```

c) Welche Erweiterungen sind notwendig, um einen durch Socketprogrammierung realisierten Server auf maximal 3 gleichzeitig arbeitende (nicht nur existierende) Threads zu beschränken und weitere Anfragen abzulehnen?

d) Der folgenden AXIS Web Service soll innerhalb eines Java-Programms aufgerufen werden. Vervollständigen Sie die fehlenden Ausdrücke an den markierten Stellen.

Datei: Adder.jws

```
public class Adder {
     public int intsum (int adam, int eve) {
          return adam + eve;
     }
}
```

Java-Programm:

```
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import javax.xml.namespace.QName;

public class Adder {
   public static void main(String[] args) {
      try {
         String endpoint = "http://localhost:8080/axis/ [          ] ";

         [          ] MeinWS = new [          ] ();

         [          ] MeinWSAufruf = MyWebService.createCall();

         MeinWSAufruf.setTargetEndpointAddress(new java.net.URL(endpoint));
         MeinWSAufruf.setOperationName(new QName(

             "http://localhost:8080/axis/ [          ] "," [          ] "));

         int ret = ((Integer) MeinWSAufruf. [          ] (new Object[] {

             new Integer(4), new Integer(7)} )).intValue();

      } catch (Exception e) {
         System.err.println(e.toString());
      }
   }
}
```

Aufgabe 6: Fallstudien (4+4+6=14 Punkte)

a) Erläutern Sie das Zusammenspiel der MS Zwischensprache und des JIT Compilers unter .NET.

b) Die Grundlage von .NET sind CTS (Common Type System) und CLS (Common Language Specification). Welche Bedingungen müssen .NET Programmiersprachen hinsichtlich CTS und CLS erfüllen?

c) Skizzieren Sie die Entwicklung einer CORBA-Anwendung. Markieren Sie dabei die Schritte, die vom Programmierer und die Schritte, die automatisch durchgeführt werden.