

Person:

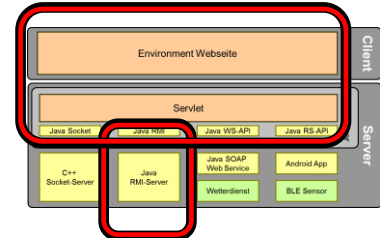
Name:

Matrikelnummer:



## Übungszettel 2

Kommunikation über RMI  
Servlets



### Aufwärmübung

„O“ (5%)

Implementieren Sie eine simple Client/Server-Kommunikation mittels RMI, in dem Sie auf dem Server eine Methode „saySomething()“ aufrufen, die auf der Server-Konsole „Kekse!“ ausgibt. Orientieren Sie sich an den Folien aus der Vorlesung!

### Aufgabe 1a

„M“ (10%)

Entwerfen Sie eine RMI-Schnittstelle (**IEnvService.java**) für die Kommunikation zwischen Client und Umweltdaten-Server. Zu unterstützen sind dabei folgende Methoden (oder solche mit äquivalenter Funktionsweise):

```
/**
 * Liefert die Typen der zur Verfügung stehenden Umweltsensoren
 *
 * @return Ein String-Array mit den Typen der Umweltsensoren
 * @throws RemoteException Ein Fehler trat bei der Kommunikation auf
 * @see java.lang.String
 * @see java.rmi.RemoteException
 */
public String[] requestEnvironmentDataTypes() throws RemoteException;

/**
 * Liefert die Messwerte für einen speziellen Sensor in Form eines
 * Environment Data (EnvData) Objektes zurück
 *
 * @param _type der betreffende Umweltsensor
 * @return EnvData Die aktuellen Messwerte des entsprechenden Sensors
 *         null, falls der Sensor nicht existiert
 * @throws RemoteException Ein Fehler trat bei der Kommunikation auf
 * @see at.fhooe.mc.vis.EnvData
 * @see java.lang.String
 * @see java.rmi.RemoteException
 */
public EnvData requestEnvironmentData(String _type) throws RemoteException;

/**
 * Liefert die Messwerte aller verfügbarer Sensoren zurück
 *
 * @return EnvData[] alle verfügbaren Messwerte
 * @throws RemoteException Ein Fehler trat bei der Kommunikation auf
 * @see at.fhooe.mc.vis.EnvData
 * @see java.lang.String
 * @see java.rmi.RemoteException
 */
public EnvData[] requestAll() throws RemoteException;
```

### Aufgabe 1b

„M“ (70%)

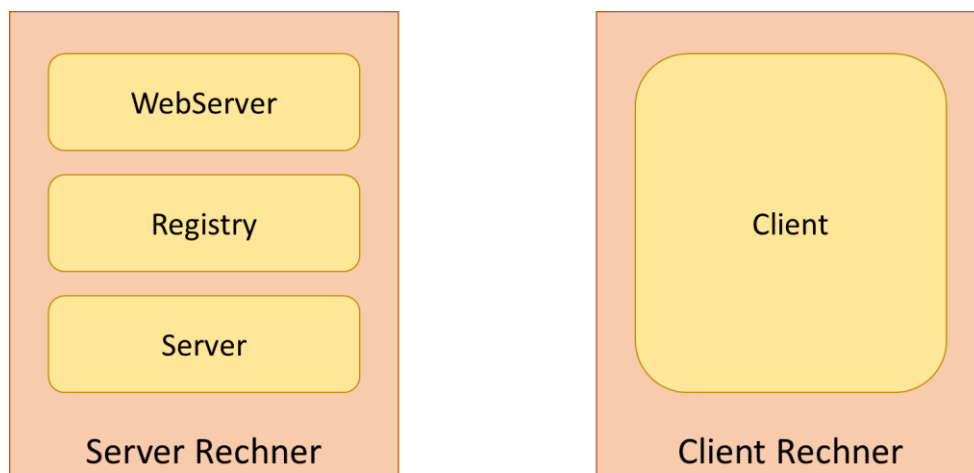
Implementieren Sie den Umweltdaten-Verwaltungsserver, in dem Sie die entsprechenden Sensoren (im Rahmen der Übung zunächst nur den Luftdruck) vorhalten/simulieren. Verändern Sie dabei über einen zufälligen, zeitgesteuerten Mechanismus die Werte der einzelnen Umweltsensoren. Melden Sie den Server an der Registry an.

Implementieren Sie einen kommandozeilenbasierten Java-Client, der sich am Server anmeldet, und analog zu Aufgabenzettel 1 alle verfügbaren Methoden kontinuierlich/periodisch verwendet und somit überprüft.

### Aufgabe 1c

„O“ (10%)

Ermöglichen Sie über die entsprechenden Mechanismen, dass der Servercode auf den Client dynamisch nachgeladen wird. Orientieren Sie sich dabei an folgender Komponenten-Verteilung bzw. Systemstruktur.



### Aufgabe 1d

„O“ (5%)

Schreiben Sie eine Klasse **ServiceMgmt**, die den Start und das Beenden des RMI-Dienstes über ein entsprechendes Menü anbietet und organisiert. Zu unterstützen sind dabei min. folgende Menüpunkte:

- Start → RMI-Dienst wird gestartet
- Stop → RMI-Dienst wird beendet
- Quit → ServiceMgmt wird beendet

## Aufgabe 2a

„O“ (5%)

Entwickeln Sie ein **HelloWorldServlet** welches die Anzahl der bisherigen Aufrufe des Servlets in einer sehr einfachen Webseite darstellt.

Versehen Sie die Klasse mit den notwendigen Annotationen und erstellen die dazugehörige Verzeichnisstruktur, zum „Ausliefern“ des Servlets. Das Servlet soll unter zwei verschiedenen Namen aufrufbar sein.

Erstellen Sie ein WAR-Archiv

```
jar -cvf ArchivName.war WEB-INF/* *.html
```

und deployen Sie es unter Tomcat.

## Aufgabe 2b

„O“ (5%)

Schreiben Sie ein Servlet (**InfoServlet**), welches die zur Verfügung stehenden Informationen über Client und Server ausliest und auf einer Webseite angeordnet zurückliefert. Zu den darzustellenden Informationen zählen: IP des Clients, Browsertyp des Clients, die akzeptierten MIME-Typen des Clients, das Client-Protokoll, der Port und der Name des Servers.

## Aufgabe 2c

„M“ (85%)

Erstellen Sie ein Servlet (**EnvironmentServiceServlet**), welches die Daten der unterschiedlichen Server tabellarisch auf einer Webseite darstellt.

### C++ Server Environment Data

Timestamp	Sensor	Value
123456	Light	750
123456	Sound	78

### RMI Server Environment Data

Timestamp	Sensor	Value
123457	Pressure	1080

Integrieren Sie hierzu die entsprechenden Client-Programme, die die Kommunikation zu den entsprechenden Servern ermöglichen. Sollte einer der Server beim Aufruf des Servlets gerade offline sein, so soll dies entsprechend in der Darstellung berücksichtigt werden

## Aufgabe 2d

„O“ (5%)

Peppen Sie die Webseite über Styling-Mechanismen auf und integrieren Sie einen „Reload“-Button bzw. einen zeitgesteuerten Mechanismus, der die Webseite alle 5 Sekunden neu lädt.