ARS

Member of

TIMETOACT GROUP
SOFTWARE & CONSULTING

Servlets - Grundlagen

Java EE Grundlagen

ARS

Inhalte dieses Kapitels

Allgemeines

Zuordnung von Requests

Konfiguration und Umgebungsinformationen

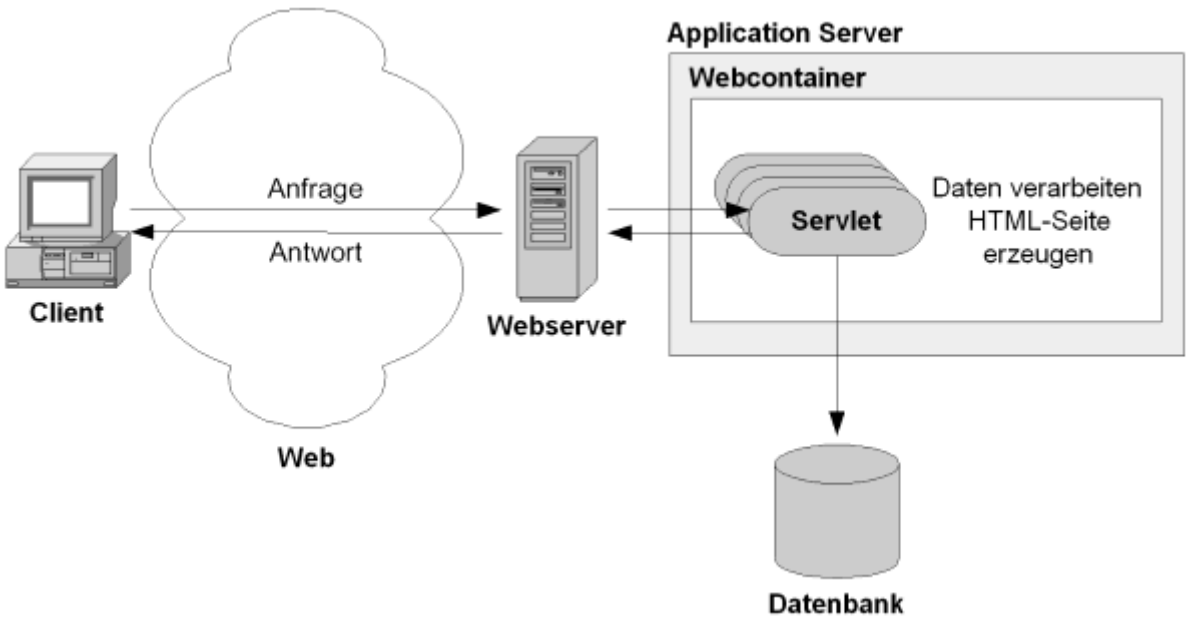
Zusammenfassung

ARS

Funktionsweise [1|2]

Was sind Servlets?

- Serverseitige, in Java geschriebene Komponenten
- Laufzeitumgebung: Webcontainer
- Implementierung des Request/Response-Ablaufmodells
  - Entgegennahme und Analyse eines Requests
  - Generierung einer Antwort



ARS

Funktionsweise [2|2]

Wie erstellt man Servlets?

- Implementierung des Interface `javax.servlet.Servlet`
- Registrierung beim Webcontainer per
  - Annotation oder
  - Eintrag in Web Deployment Descriptor
- Angabe von URL-Mapping(s) für Aufrufbarkeit per HTTP-Request

```
@WebServlet(
  value = "/control",
  name="ControllerServlet",
  displayName="Controller Servlet",
)
public class ControllerServlet
    implements Servlet {
  …
}
```

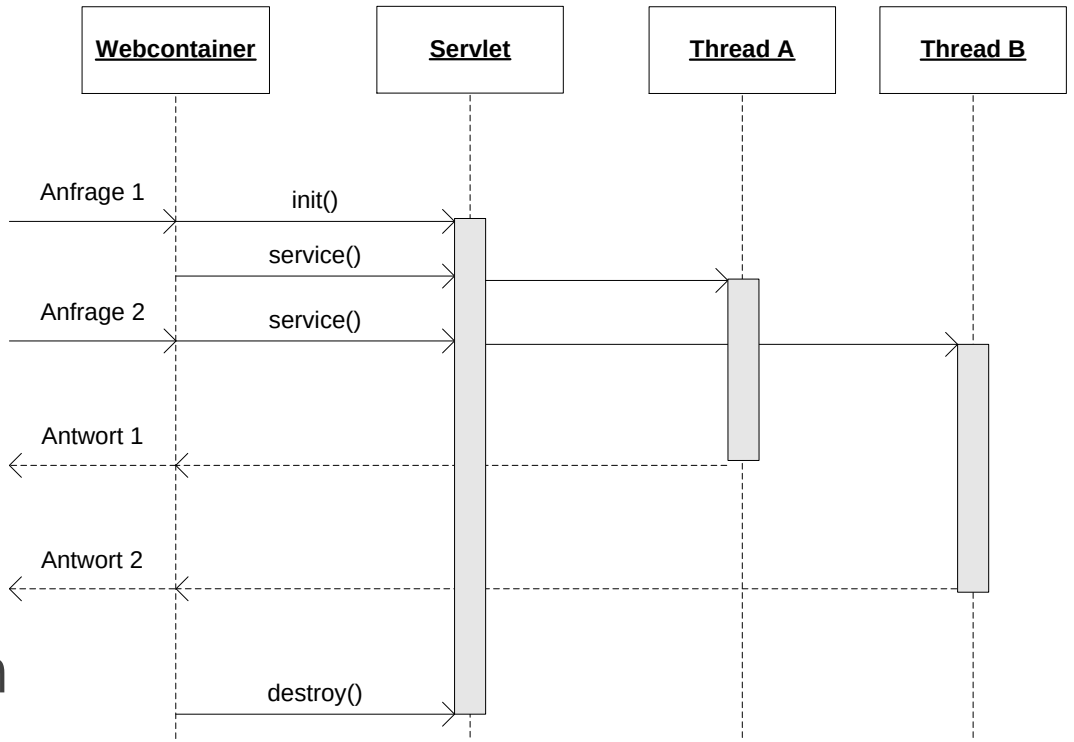
```
<servlet>
  <servlet-name>ControllerServlet</servlet-name>
  <display-name>Controller Servlet</display-name>
  <servlet-class>
    de.ars.library.ControllerServlet
  </servlet-class>
</servlet>

<servlet-mapping>
   <servlet-name>ControllerServlet</servlet-name>
   <url-pattern>/control</url-pattern>
</servlet-mapping>
```

ARS

Lebenszyklus

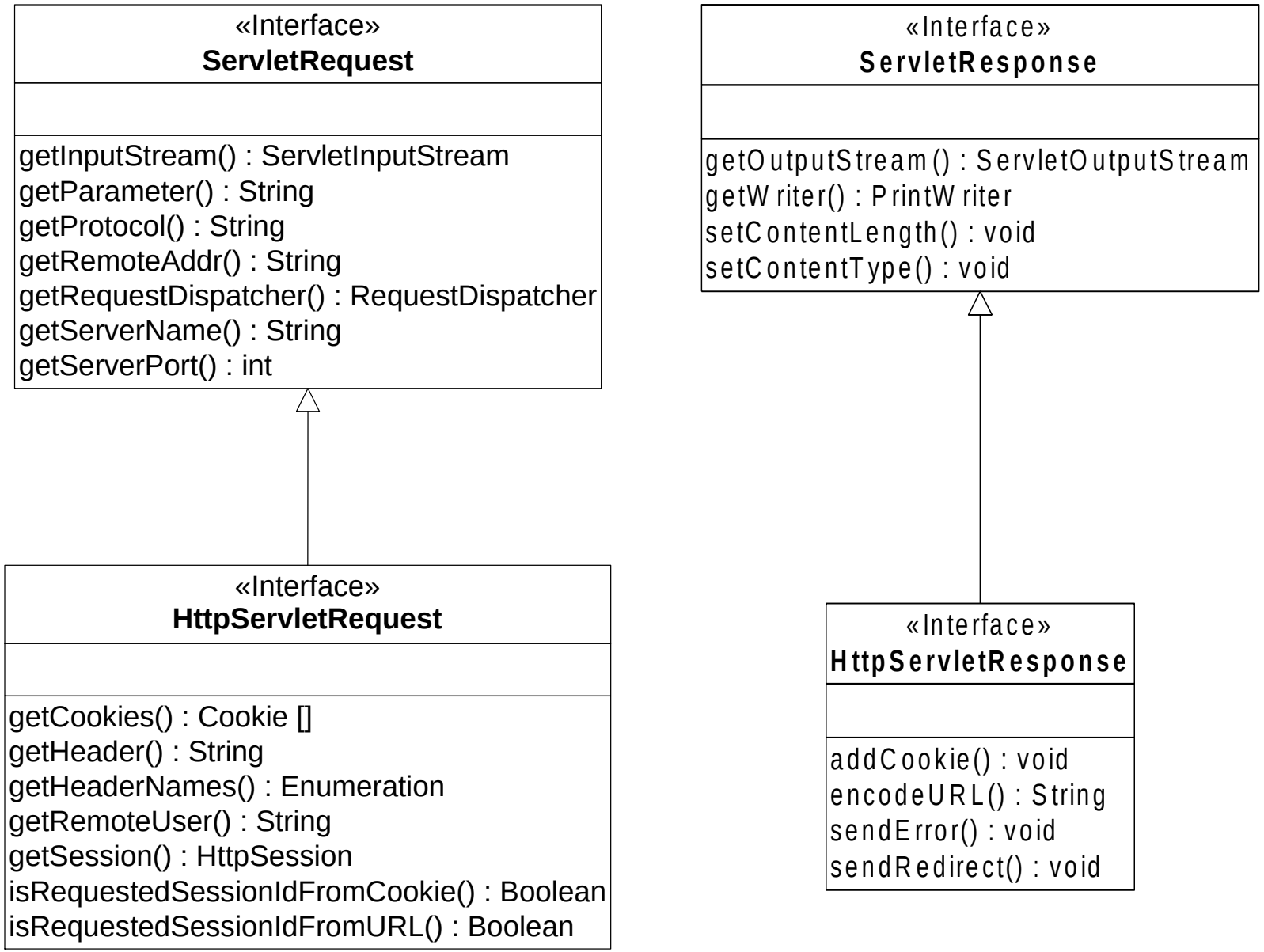
Es gibt im Webcontainer max. eine Instanz eines Servlets!

- `init()` / `@PostConstruct` - Methoden
  - Initialisierung (einmalig)
    - Auslesen von Init.-Parametern
    - Setzen von Instanzvariablen
- `service()`
  - Aufruf bei jedem Request in separatem Thread
  - Ausführung von Programmlogik
- `destroy()` / `@PreDestroy` - Methoden
  - Finalisierung (einmalig) beim Stoppen der Anwendung bzw. des Application Servers
    - Freigabe von Ressourcen



ARS

Servlet API



ARS

Beispiel [1|2]

Webseite mit HTML-Formular

```
<html>
  <head>
    <title>Anmeldung</title>
  </head>

  <body>
    <h1>Willkommen beim Newsletter-System</h1>
    Bitte geben sie Ihre Email-Adresse ein:
    <form method="post" action="register">
      <input type="text" size="30" name="email"/>
      <input type="submit" value="Registrieren"/>
    </form>
  </body>
</html>
```

ARS

Beispiel [2|2]

Servlet zur Auswertung der Anfrage (vereinfacht)

```
public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
  // Parameter vom HTML-Formular abfragen
  String email = request.getParameter("email");
  // Email speichern ...
  // Format der Antwort festlegen
  response.setContentType("text/html");
  // Ausgabe generieren
  try(PrintWriter out = response.getWriter()) {
    out.println("<h1>!! Danke für die Registrierung !!</h1>");
    out.println("<p>Ihre Daten lauten: " + email + "</p>");
  }
}
```

ARS

Schritte bei der Requestverarbeitung [1|2]

Terminanfrage

JavaServer Faces – Programmierung mit Eclipse

Bitte füllen Sie die notwendigen Felder aus. Geben Sie weitere Kontaktinformationen in das Nachrichtenfeld ein, falls erwünscht.

Name* Titel **Vorname*** **Nachname***

Frau

Firma* **Abteilung / Beruf***

Straße Land PLZ Stadt

D

Email* **Telefon*** Fax

+49-

Wunschtermin* **Anzahl der Teilnehmer*** **Lokation***

...

1

Wir stellen selbst einen Raum bereit.

Ihre Nachricht

Berechnen Sie bitte die Differenz von eins und vier:*

Absenden

***Für fett markierte Felder ist die Eingabe erforderlich.**

ARS

Schritte bei der Requestverarbeitung [2|2]

1. Anfrage-Parameter bzw. -Header

Reaktionen auf Fehler:

a) Auslesen

b) Auf Nullwerte prüfen `!! HTTP 400 !!`

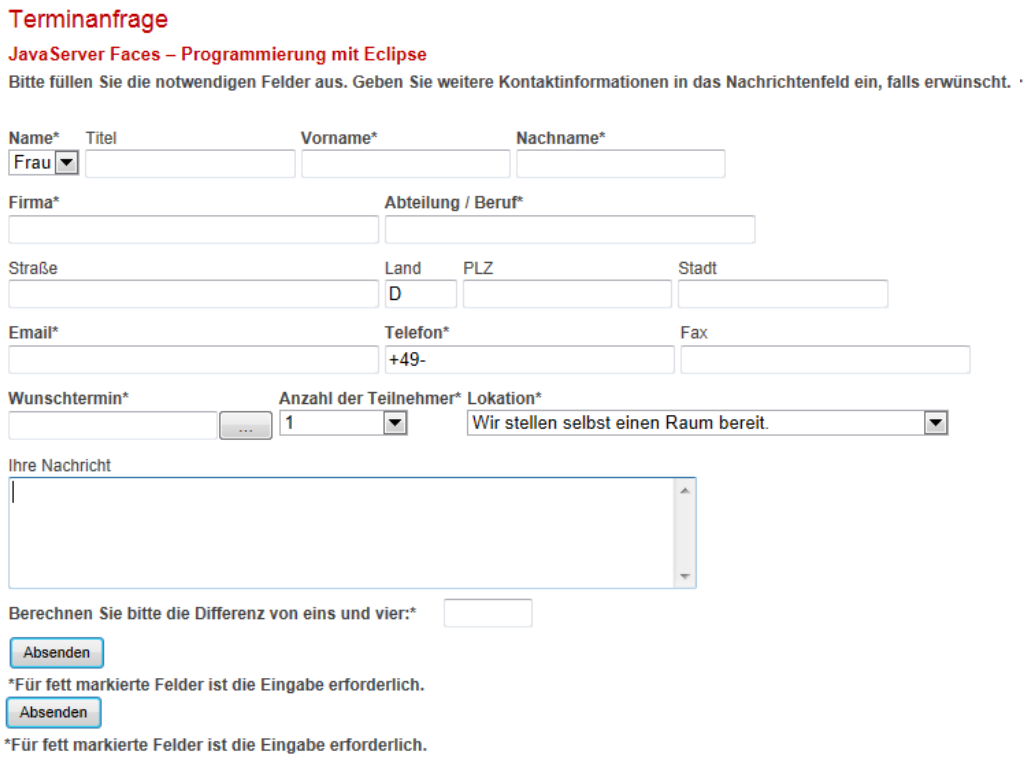
c) Konvertieren

d) Validieren

} `HTTP 200`

2. Aktion ausführen `!! HTTP 500 !!`

3. Antwort generieren `??`



ARS

Begriffe

Context Path

- Teil der URL zur Adressierung eines Webmoduls

Servlet Path

- Pfad zur Adressierung eines Servlet
- Direkt nach dem Context Path

Context Path **Servlet Path**

```
http://myserver.com/tours/travelagency/OrderServlet
```

ARS

URL Mapping [1|2]

Reihenfolge bei der Zuordnung

1. Exakte Übereinstimmung
2. Längstes Präfix, z.B. virtuelle Verzeichnisse (`/subdirectory/*`)
3. Längstes Suffix, z.B. Dateiendungen (`*.do`)
4. DefaultServlet, falls vorhanden (`*` oder `/`)

Einschränkungen

- Groß- und Kleinschreibung relevant
- KEINE beliebigen regulären Ausdrücke
- maximal EINE Wildcard
- Wildcard NUR am Anfang/Ende eines Mappings

ARS

URL Mapping [2|2]

Beispiel

```
@WebServlet( value = {"/control.html", "/ctrl/*", "*.ctrl"} )
public class ControllerServlet implements Servlet { … }
```

http://localhost:9080/MyModule/control.html

http://localhost:9080/MyModule/test.html

http://localhost:9080/MyModule/ctrl/execute

http://localhost:9080/MyModule/ctrl/test/execute

http://localhost:9080/MyModule/temp/execute.ctrl

http://localhost:9080/MyModule/temp/execute.test

?? http://localhost:9080/MyModule/ctrl/test/execute.ctrl ??

```
@WebServlet( value = {"/test.html", "/ctrl/test/*", "*.test"} )
public class TestControllerServlet implements Servlet { … }
```

ARS

Konfiguration [1|2]

Festlegung von Servlet-Parametern

- Key-Value-Paare (String)
- Teil der Servlet-Definition

```
@WebServlet(
  initParams = {
    @WebInitParam(
      name = "mailTo",
      value = "java@ars.de"
    )
  }
)
```

```
<servlet>
  [...]
  <init-param>
    <param-name>mailTo</param-name>
    <param-value>java@ars.de</param-value>
  </init-param>
</servlet>
```

ARS

Konfiguration [2|2]

Abfrage von Servlet-Parametern

- Objekt der Klasse `ServletConfig`
  - Pro Servlet eigene Instanz
  - Konfigurationseinstellungen zum Servlet
    - Name
    - Initialisierungsparameter
  - Injection durch Webcontainer während der Initialisierung eines Servlets

«Interface» **ServletConfig**
getInitParameter() : String getInitParameterNames() : Enumeration getServletContext() : ServletContext getServletName() : String

```
private String mailToAddress;

// wird aufgerufen durch Webcontainer
@Override
public void init(ServletConfig config) {
  mailToAddress = config.getInitParameter("mailTo");
}
```

ARS

Umgebungsinformationen [1|2]

Konfigurationsinformationen für die Webapplikation

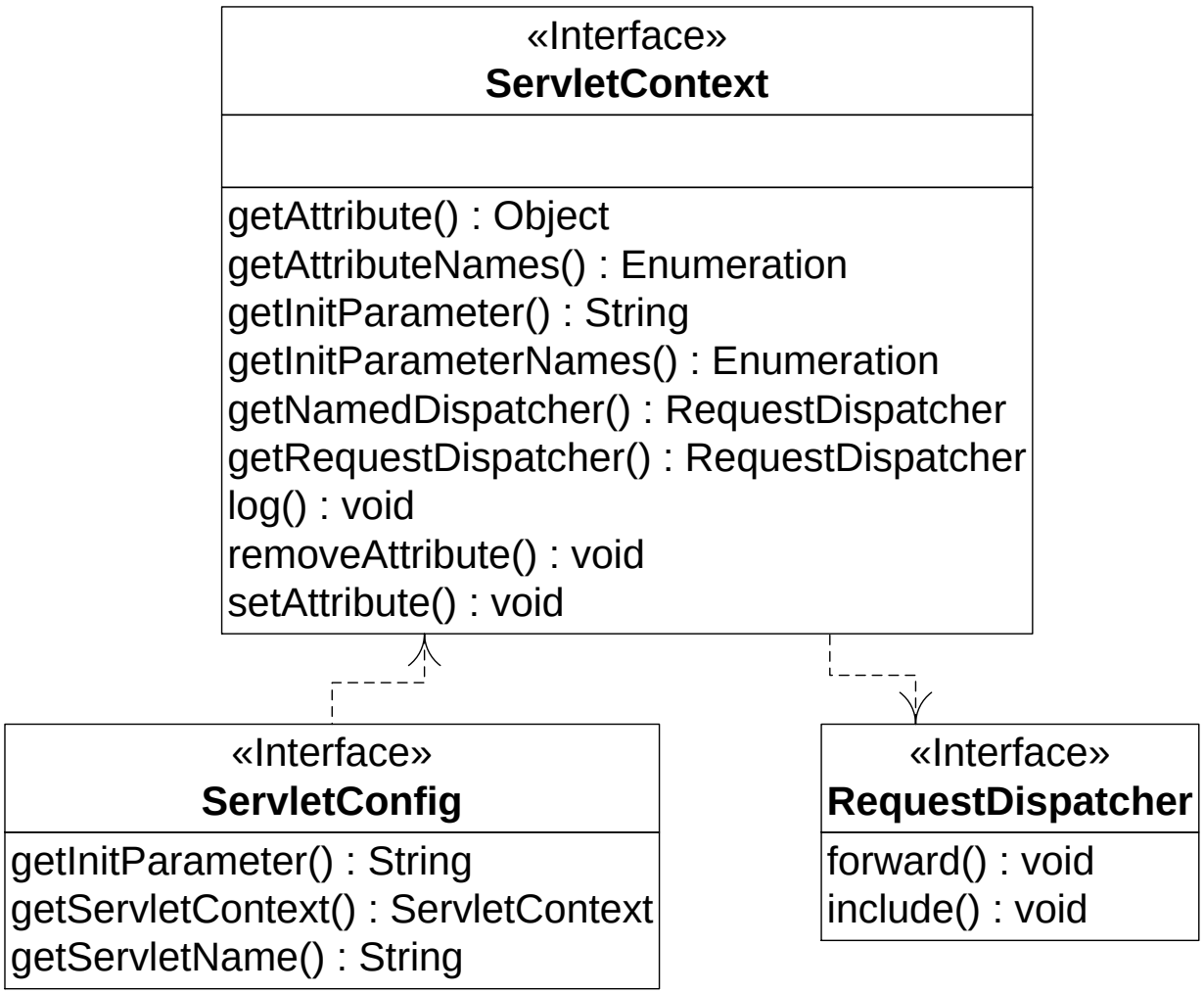
- Objekt der Klasse `ServletContext`
  - Instanz für gesamte Webapplikation
  - Kontext, in dem die Servlets ablaufen

Enthaltene Informationen

- Kontextparameter
  - Initialisierungsparameter der Webapplikation
- Attribute
  - Werden zur Laufzeit angelegt/gelöscht
  - Globaler/gemeinsamer Namensraum aller Servlets
- Allgemeine Informationen zum Webcontainer (z.B. API-Version)
- Zugriff auf Ablaufumgebung (z.B. absolute Pfade, Ressourcen)

ARS

Umgebungsinformationen [2|2]



ARS

Kontrollfragen

- Was sind Servlets?
- Wo und in welcher Form können Servlets konfiguriert werden?
- In welcher Beziehung (Multiplizität) stehen die folgenden Begriffe?
  - Servlet (-Definition)
  - Servlet-Klasse
  - URL-Mapping
- Wofür werden Initialisierungsparameter verwendet?
- Wieviele Instanzen eines Servlets gibt es zur Laufzeit im Webcontainer und wann werden diese erzeugt bzw. zerstört?