



VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY  
**UNIVERSITY OF INFORMATION TECHNOLOGY**

---



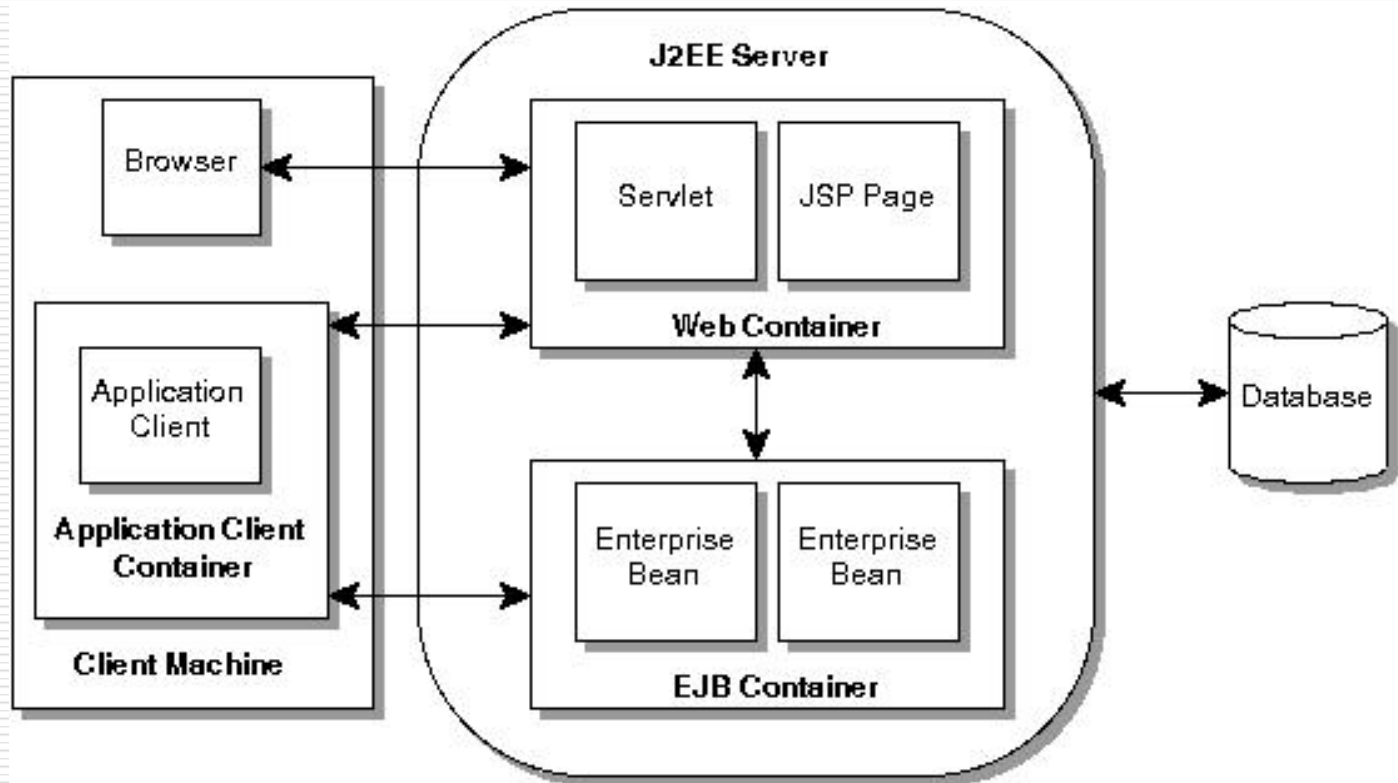
## Chapter 6

# **JAVA SERVLET**

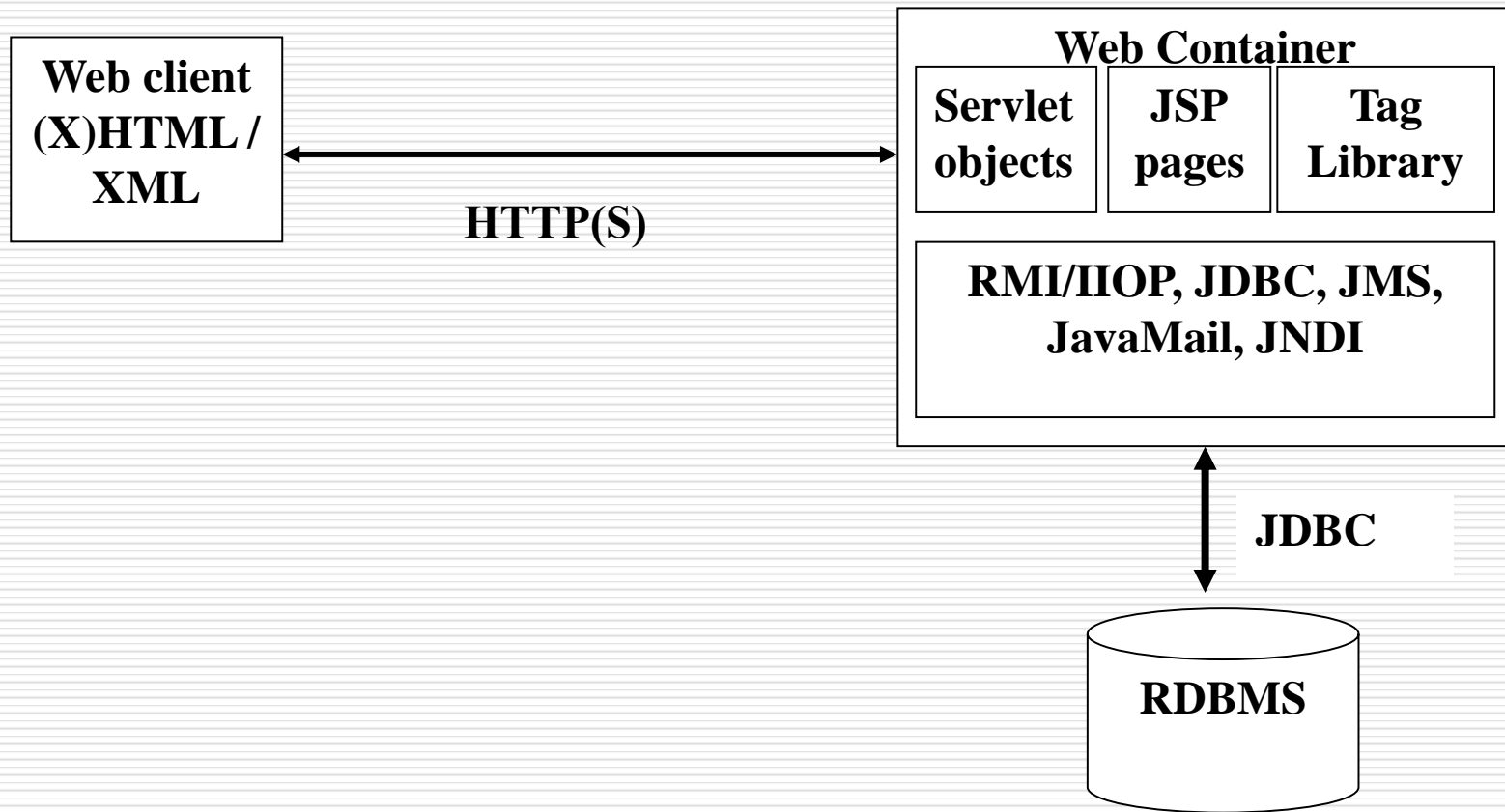
Lecturer: MSc. Kiet Van Nguyen  
Faculty of Information Science and Engineering  
University of Information Technology, VNU-HCM

# Kiến trúc J2EE

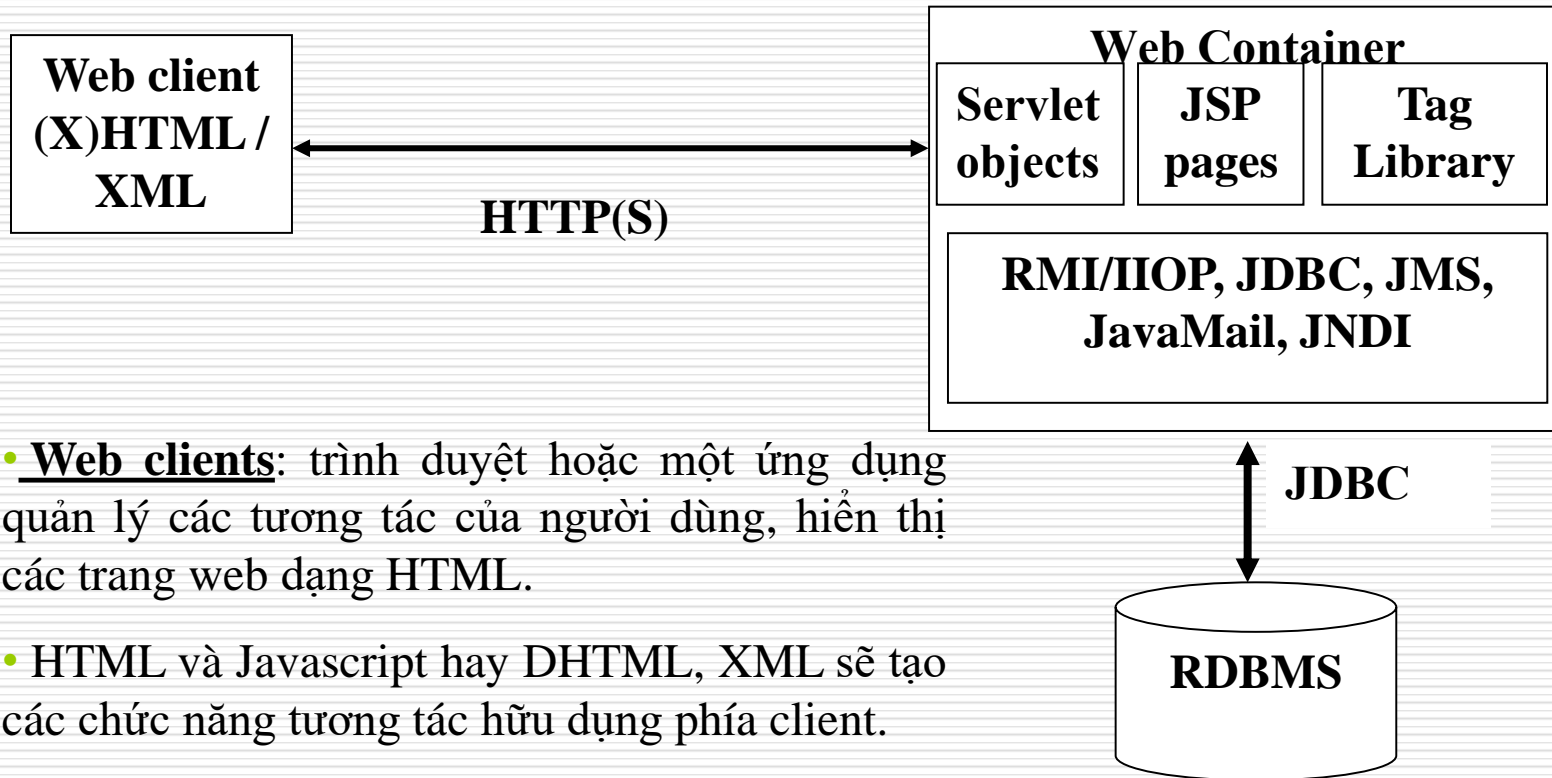
- Kiến trúc J2EE bao gồm nhiều kỹ thuật lập trình phía server



# Kiến trúc của J2EE

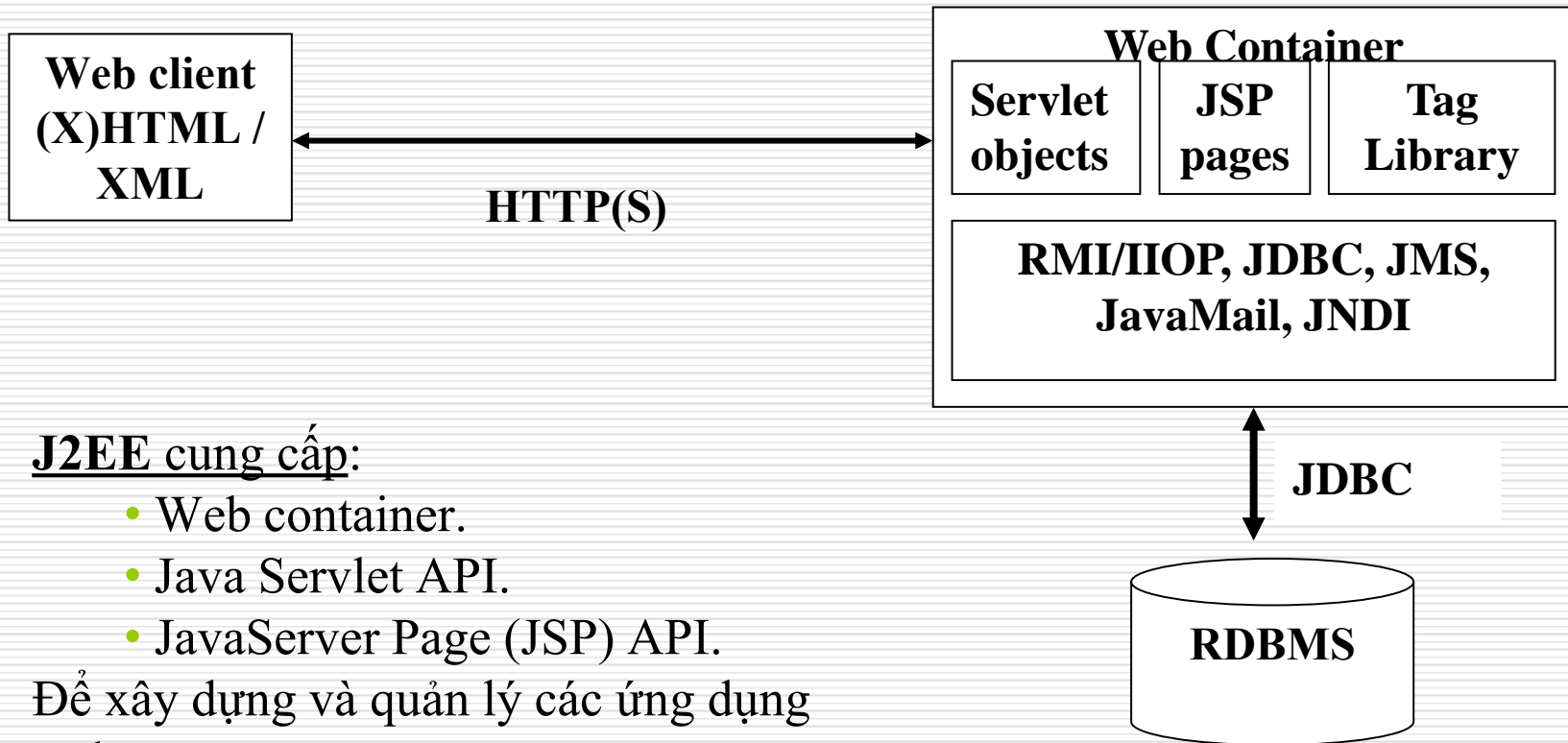


# Kiến trúc của J2EE



- **Web clients:** trình duyệt hoặc một ứng dụng quản lý các tương tác của người dùng, hiển thị các trang web dạng HTML.
- HTML và Javascript hay DHTML, XML sẽ tạo các chức năng tương tác hữu dụng phía client.
- HTTP hoặc HTTPS là giao thức chuyển đổi thông tin của web client và các ứng dụng web trên web server.

# Kiến trúc của J2EE

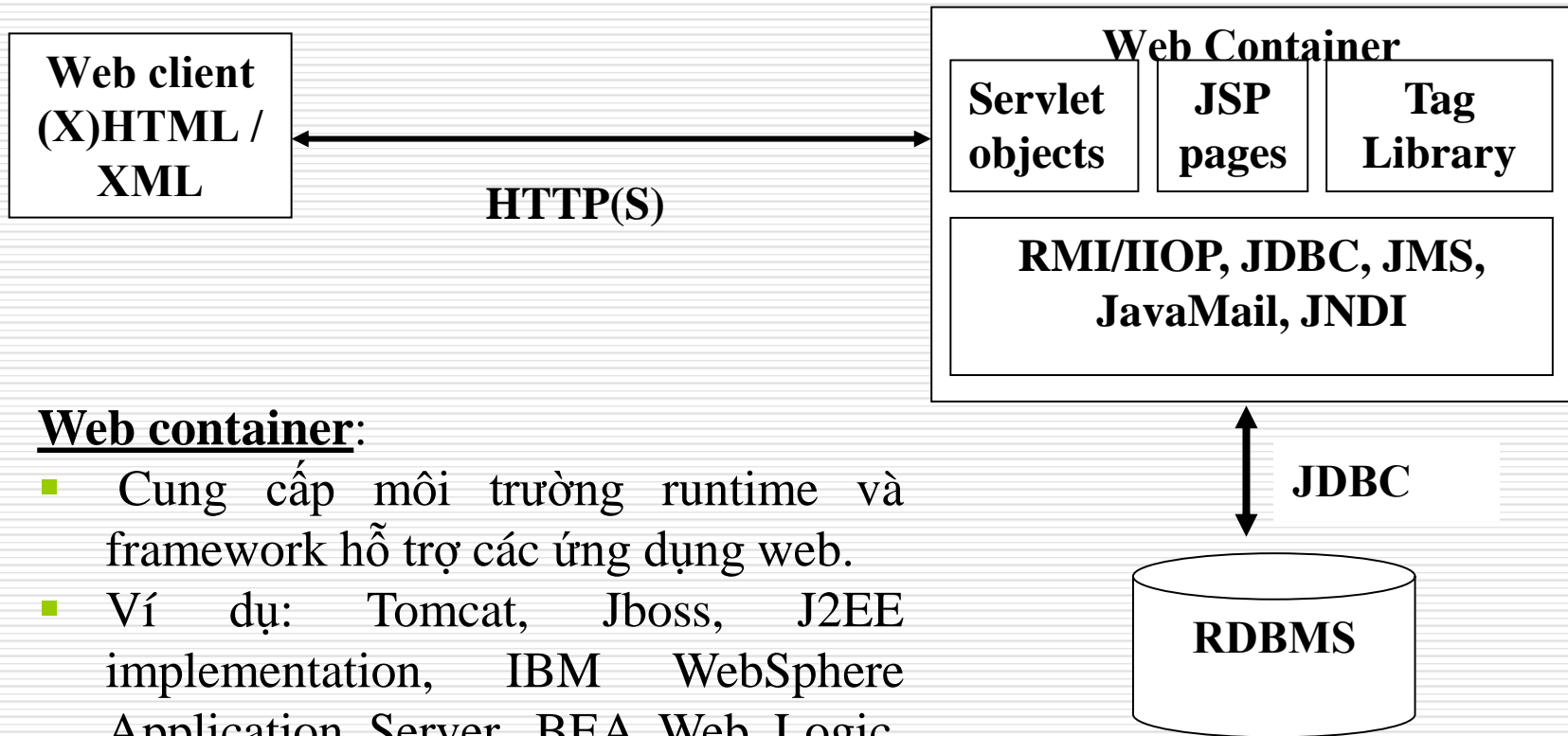


## J2EE cung cấp:

- Web container.
- Java Servlet API.
- JavaServer Page (JSP) API.

Để xây dựng và quản lý các ứng dụng web.

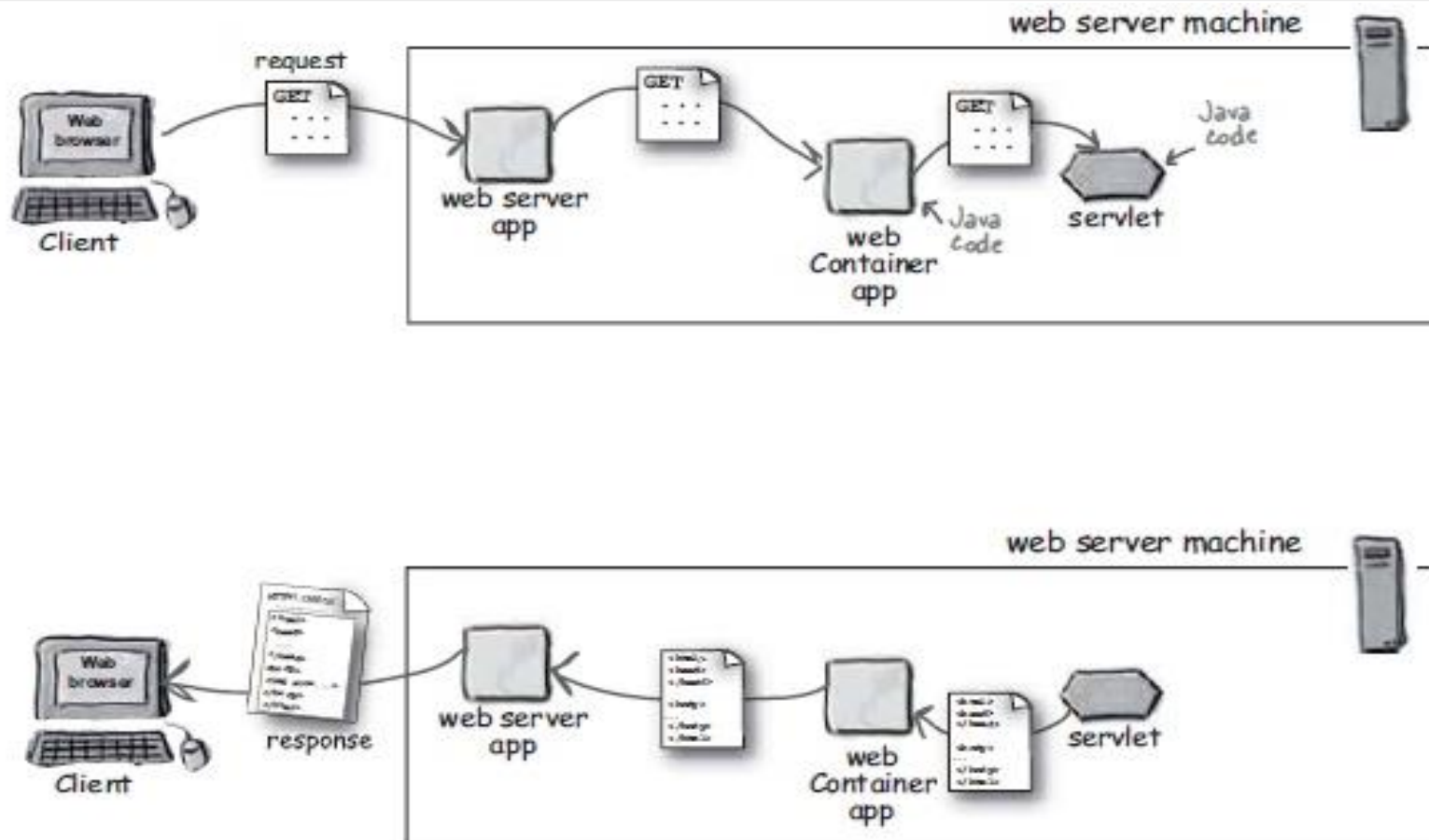
# Kiến trúc của J2EE



## Web container:

- Cung cấp môi trường runtime và framework hỗ trợ các ứng dụng web.
- Ví dụ: Tomcat, Jboss, J2EE implementation, IBM WebSphere Application Server, BEA Web Logic, Oracle Application Server...

# WEB CONTAINER



---

# Giới thiệu Java Servlet

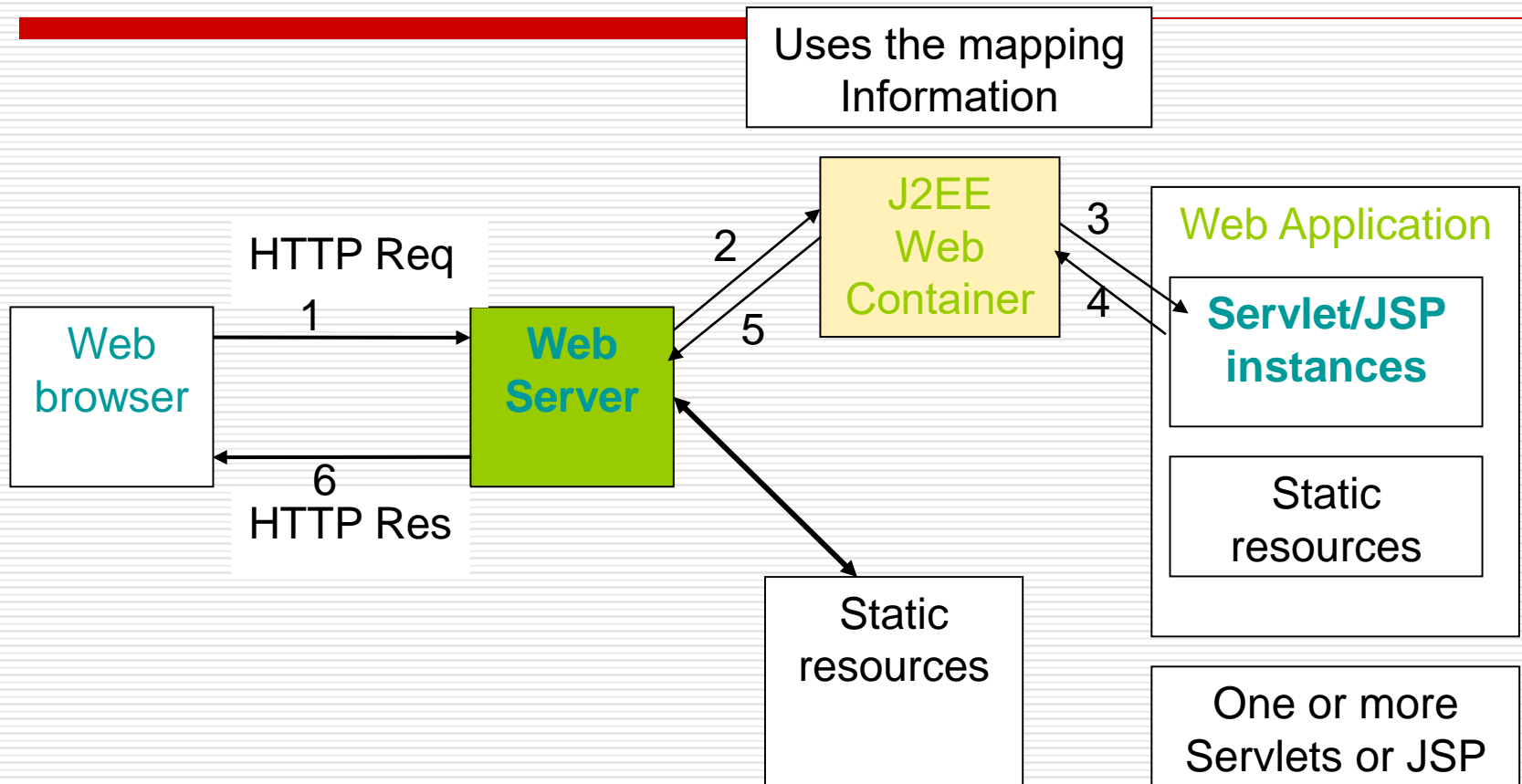


# Java Servlet

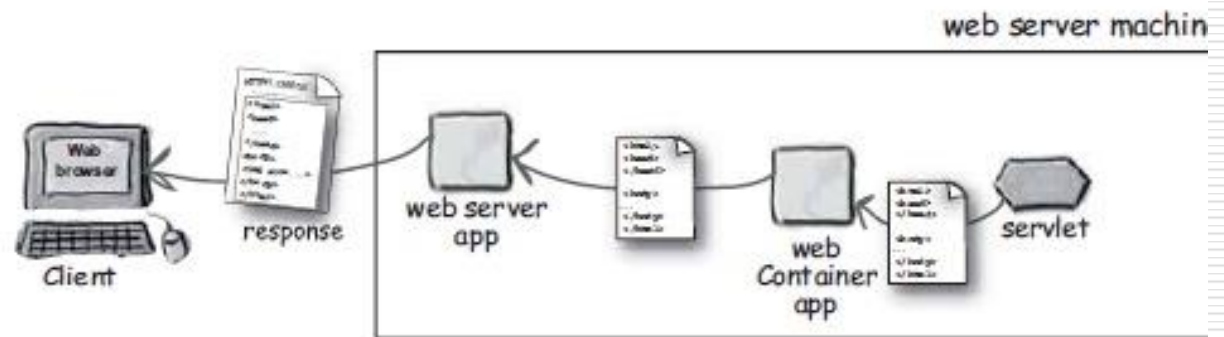
---

- Java servlet là những chương trình Java chạy ở phía server được lập trình để mở rộng chức năng của web server. Nghĩa là bạn có thể nhúng logic của ứng dụng riêng vào việc xử lý **các HTTP request và tạo HTTP response**.
- Java servlet không thể gọi từ client mà web container chứa ứng dụng web có server sẽ bị gọi.

# Quy trình cơ bản khi web server nhận một HTTP request

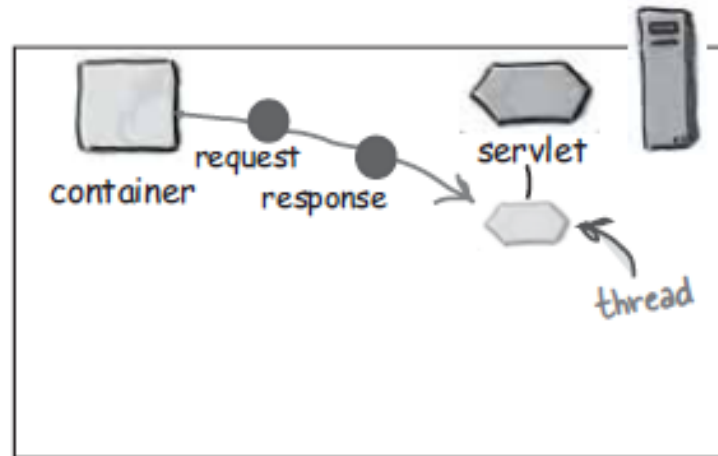


# Servlet Xử Lý HTTP Request



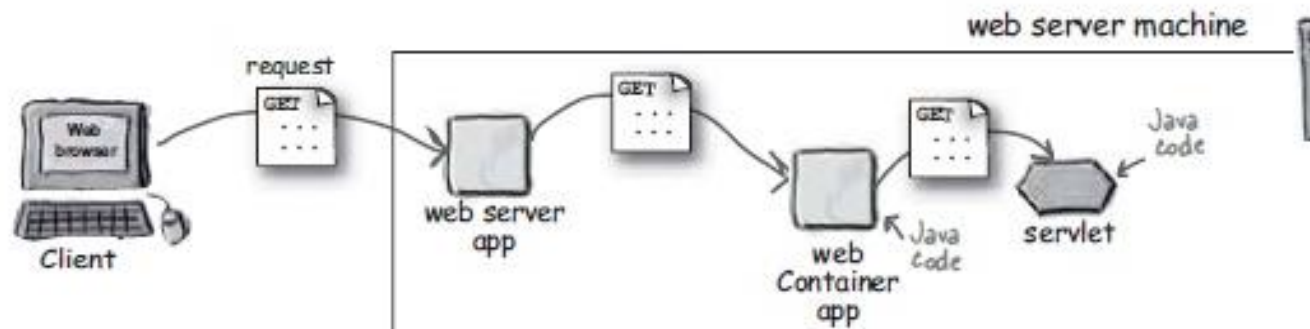
# Servlet Xử Lý HTTP Request

③



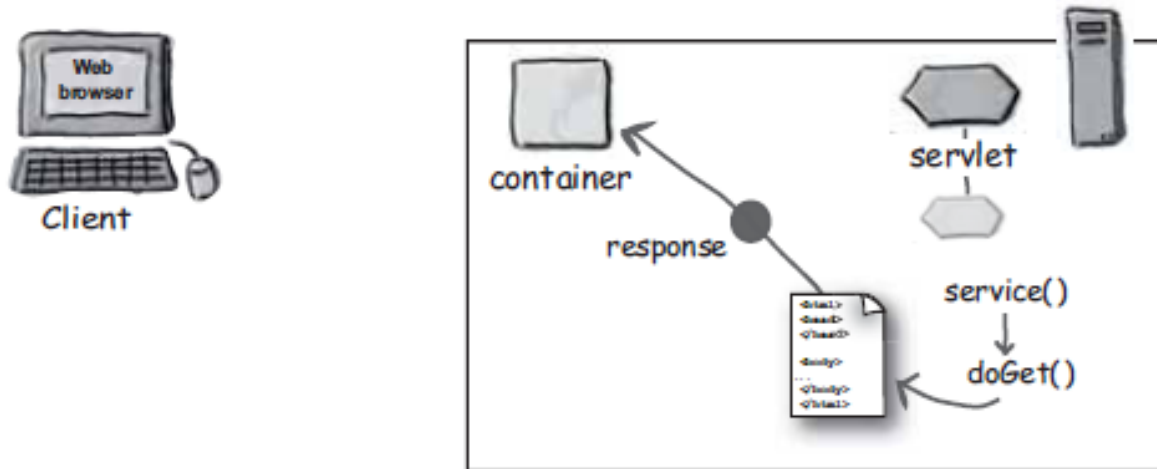
The container finds the correct servlet based on the URL in the request, creates or allocates a thread for that request, and passes the request and response objects to the servlet thread.

At this point, the servlet is responsible for the container, and gives the servlet the HTTP request and response, and it's the Container that calls the servlet's methods (like `doPost()` or `doGet()`).



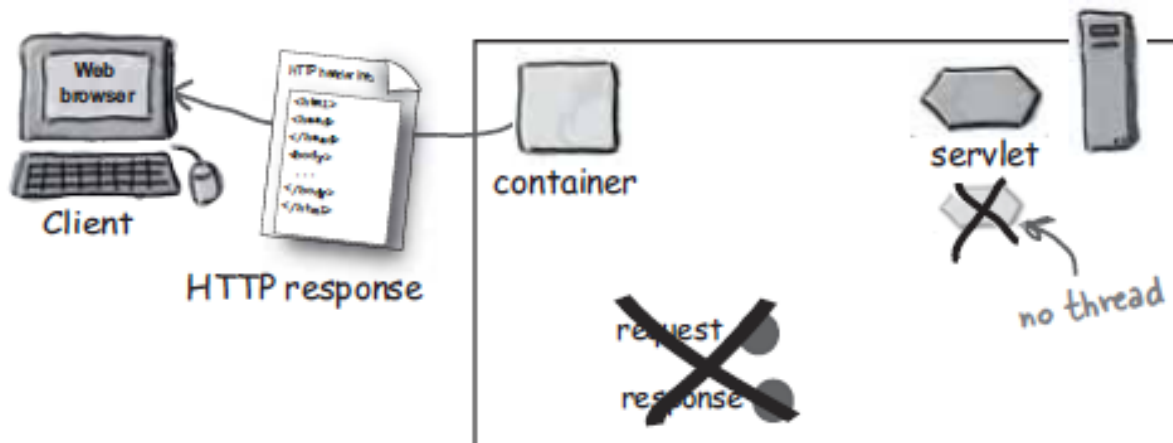
# Servlet Xử Lý HTTP Request

5



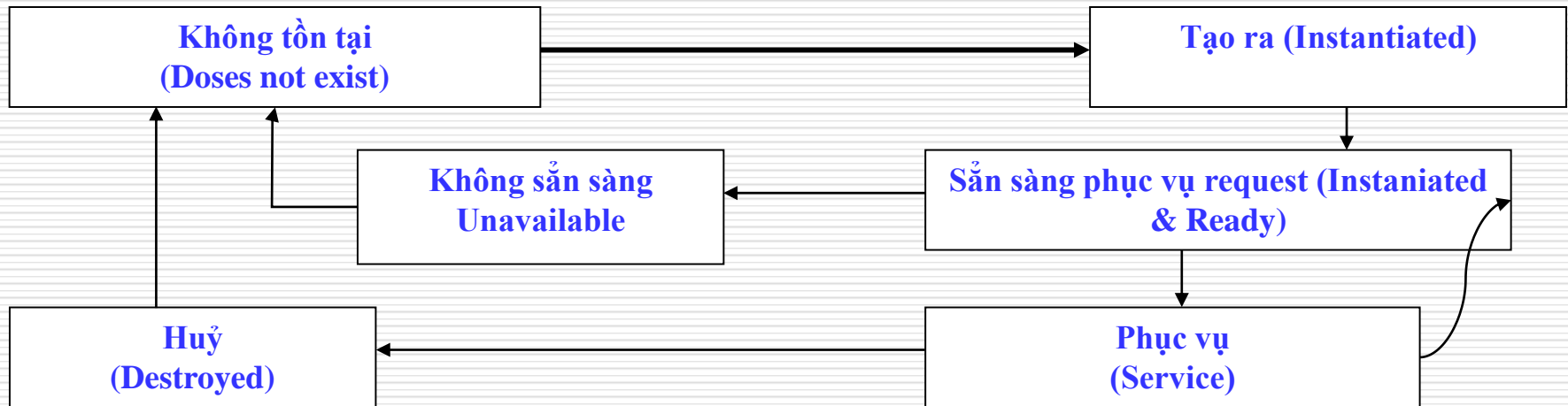
The doGet() method generates the dynamic page and stuffs the page into the response object. Remember, the container still has a reference to the response object!

6



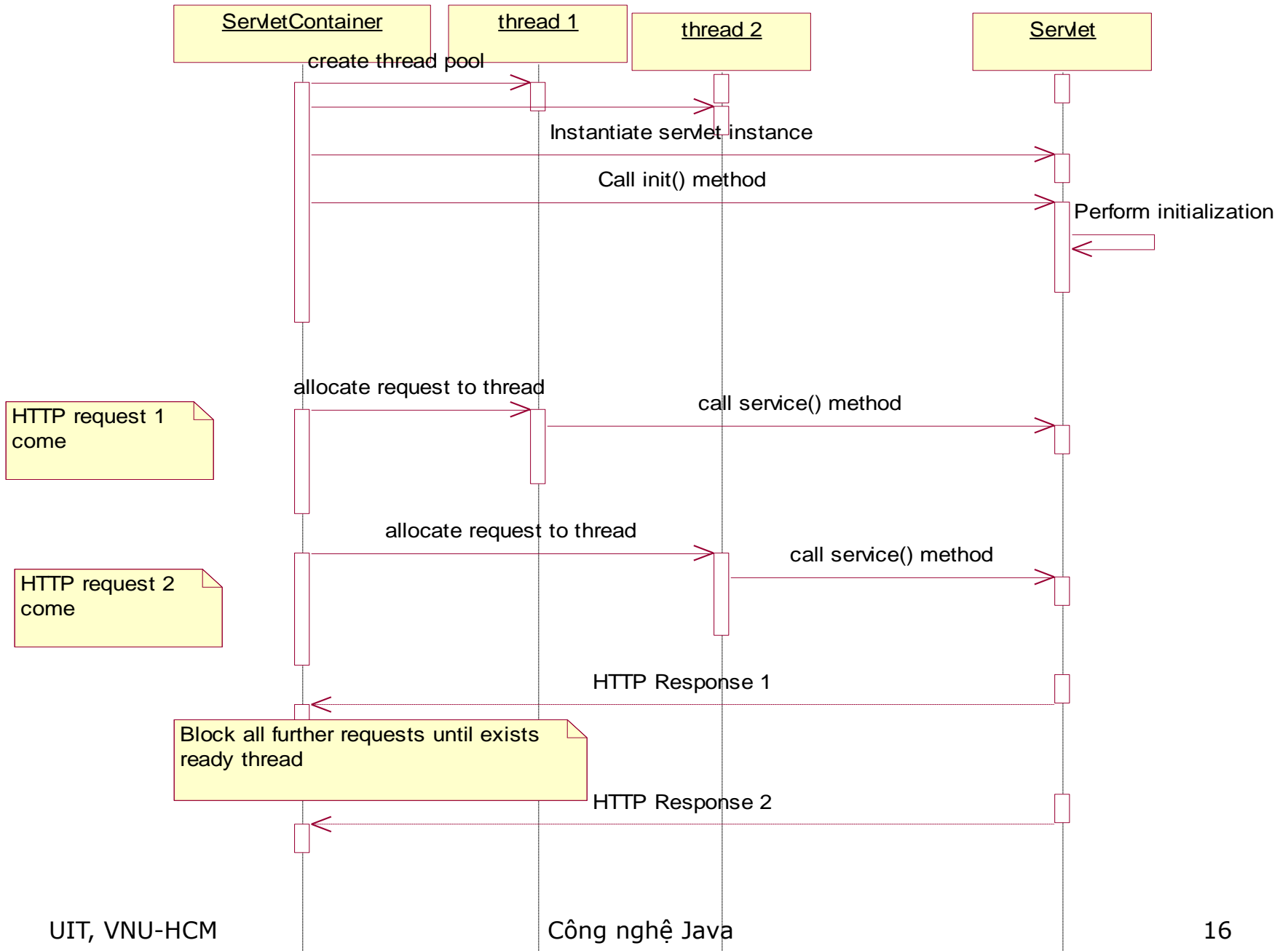
The thread completes, the container converts the response object into an HTTP response, sends it back to the client, then deletes the request and response objects.

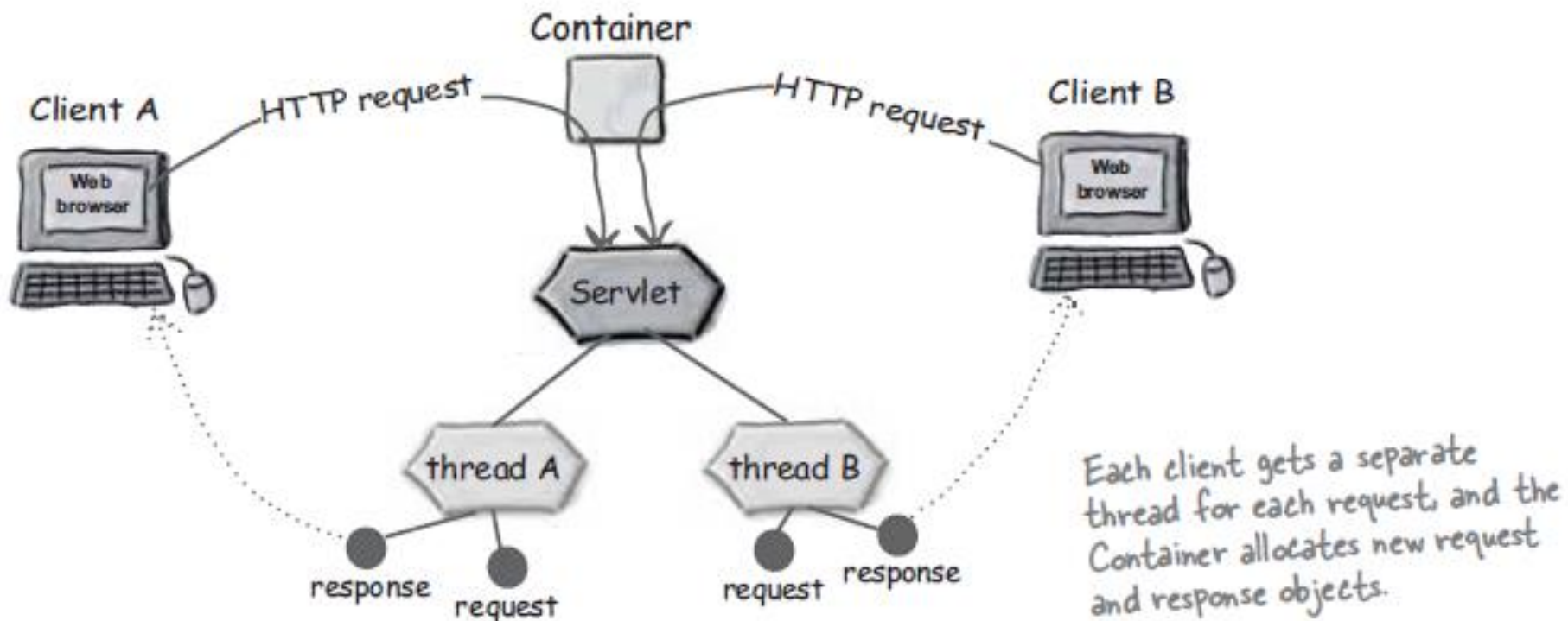
# Vòng đời của servlet



Vòng đời của servlet gồm 6 trạng thái:

- Tạo ra: web container tạo một thực thể servlet
- Sẵn sàng phục vụ: gọi method `init()` của servlet
- Phục vụ: nếu có một request thì container sẽ gọi method `service()` của thực thể servlet.
- Huỷ: container gọi method `destroy()` của thực thể servlet.
- Không sẵn sàng: thực thể được đánh dấu cho garbage collection.







# Cấu trúc cơ bản của Servlet

---

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class ServletTemplate extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
        // Use "req" to read incoming HTTP headers

        // Use "res" to specify the HTTP response status
        // code and headers (e.g. the content type, cookies).

        PrintWriter out = res.getWriter();
        // Use "out" to send content to browser
    }
}
```

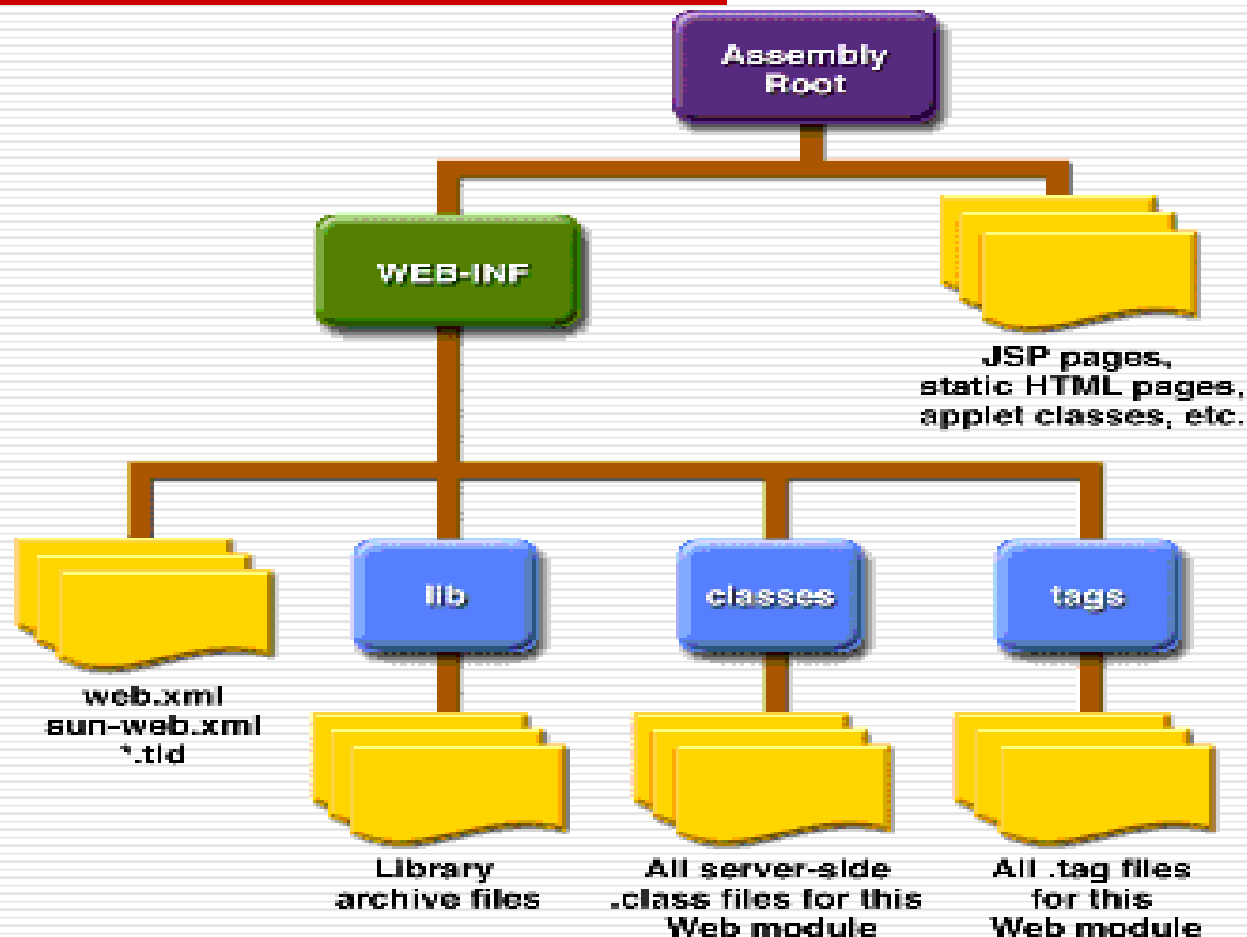
# The HelloWorld Servlet

---

```
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;

public class HelloServlet extends HttpServlet
{
    public void doGet(HttpServletRequest req,
        HttpServletResponse res) throws IOException,
        ServletException{
        res.setContentType("text/plain");
        PrintWriter out = res.getWriter();
        out.println("Hello, World!");
    }
}
```

# CẤU TRÚC CỦA WEB JAVA APPLICATION



UIT, VNU-HCM Công nghệ Java 21  
Cấu trúc này được đóng gói thành tập tin \*.war để deploy trên Web Server

# Cấu Hình Servlet trong web.xml

---

```
<web-app>  
  <servlet>  
    <servlet-name>Internal Name</servlet-name>  
    <servlet-class>HelloServlet</servlet-class>  
  </servlet>  
  <servlet-mapping>  
    <servlet-name>Internal Name</servlet-name>  
    <url-pattern>/PublicName</url-pattern>  
  </servlet-mapping>  
</web-app>
```

# Lập trình với Java servlet API

- Java Servlet API gồm:
  - `javax.servlet.*`
    - Chứa các class và interface độc lập với HTTP
  - `javax.servlet.http.*`
    - Chứa các class cụ thể cho HTTP
- Tất cả servlet đều phải hiện thực giao diện `javax.servlet.Servlet`. Tuy nhiên, do chúng ta dùng giao thức HTTP và `javax.servlet.http.HttpServlet` là lớp hiện thực giao diện Servlet, nên thường ta phải kế thừa từ lớp này và override `doGet()` và/hoặc `doPost()`.

# Servlet API

---

- Lớp GenericServlet
  - Triển khai kỹ thuật lập trình Servlet.
  - Ngoài ra còn implements Serializable, ServletConfig
- Lớp HttpServlet
  - Kế thừa lớp GenericServlet
  - Cung cấp thêm cơ chế xử lý giao thức HTTP

# Phương thức trong HttpServlet

---

- HttpServlet cung cấp các phương thức để xử lý các yêu cầu trong giao thức HTTP
  - doGet
  - doPost
  - doPut, doDelete (rare)
  - doTrace, doOptions (not overridden)
- Phương thức service() chuyển yêu cầu nhận được sang phương thức do\* thích hợp

# Lớp ServletRequest

---

- Đóng gói thông tin giao tiếp client - server
- Cho phép Servlet truy cập
  - Tên các tham số client gửi lên
  - Giao thức client đang sử dụng
  - Tên của máy client
  - Luồng nhập ServletInputStream để server lấy dữ liệu từ client
- Lớp con của ServletRequest cho phép servlet lấy thêm dữ liệu cho từng giao thức cụ thể
  - HttpServletRequest cho phép lấy thông tin HTTP-header



# ServletRequest - Client Info

---

- `getRemoteAddr()`
  - Trả về địa chỉ IP của máy gửi yêu cầu
- `getRemoteHost()`
  - Trả về tên của máy gửi yêu cầu
- `getProtocol()`
  - Trả về giao thức của yêu cầu đang sử dụng là một chuỗi có dạng  
`<protocol>/<major version>.<minor version>.`

# ServletRequest - URL Info

---

- `getScheme()`
  - Trả về tên giao thức của yêu cầu, ví dụ: "http", "https", or "ftp".
- `getServerName()`
  - Trả về tên server nhận yêu cầu
- `getServerPort()`
  - Trả về port mà server dùng để nhận yêu cầu
- `getServletPath()`
  - Trả về URL chỉ đến servlet, ví dụ: “/servlet/MyServlet”

# ServletRequest - Contents

---

- `getLength()`
  - Trả về kích thước của dữ liệu trong yêu cầu
- `getContentType()`
  - Trả về MIME type của dữ liệu trong yêu cầu
- `getInputStream()`
  - Trả về `InputStream` để đọc dữ liệu trong yêu cầu.
- `getReader()`
  - Trả về `BufferedReader` để đọc dữ liệu trong yêu cầu.

# ServletRequest - Parameters

---

- `String getParameter(String)`
  - Trả về chuỗi chứa giá trị của tham số, null nếu không tồn tại.
- `String[] getParameterValues(String)`
  - Tương tự `getParameter()` nhưng cho các tham số trả về nhiều giá trị
- `Enumeration getParameterNames()`
  - Trả về tên của các tham số.

# Lớp ServletResponse

---

- Đóng gói thông tin giao tiếp server - client
  - Cung cấp các phương thức để servlet trả lời cho client
  - Cho phép servlet xác định chiều dài và MIME type của dữ liệu gửi cho client
  - Cung cấp ServletOutputStream để server gửi dữ liệu cho client
- Các lớp con của ServletResponse cung cấp servlet khả năng thao tác cho giao thức cụ thể.
  - HttpServletResponse để thao tác trên thông tin của HTTP-header

# ServletResponse

---

- Đóng gói phản hồi của server
- Cách sử dụng:

```
response.setContentType("text/html");  
PrintWriter out = response.getWriter();  
out.println(  
    "<HTML><BODY>Hello</BODY></HTML>");
```

- `setContentType()` được dùng trước khi gọi `getWriter()` hay `getOutputStream()`.

# ServletResponse - Output

---

- `getWriter()`
  - Lấy về Writer để ghi dữ liệu
- `getOutputStream()`
  - Tương tự `getWriter` nhưng dùng cho dữ liệu nhị phân
- Và nhiều phương thức khác tương tự `ServletRequest`

# Servlet xuất web page

---

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HtmlServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Hello World!</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Hello World!</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```



## Bài tập: Ứng dụng tra cứu DBĐT bằng servlet

---

- Ứng dụng danh bạ điện thoại:
  - Viết một **InputForm servlet** để tạo một HTML form nhập để tra cứu danh bạ điện thoại.
  - Viết một **SearchAccount servlet** để tra cứu thông tin danh bạ.

# Bài tập: U.D DBĐT - Viết servlet InputForm

---

```
1. import javax.servlet.*; import javax.servlet.http.*;
2. import java.io.IOException;

3. public class InputForm extends HttpServlet {
4.     static final private String CONTENT_TYPE = "text/html";
5.     //Initialize global variables
6.     public void init() throws ServletException {
7.         // TO DO ...
8.     }
9.     public void doGet (HttpServletRequest request,
10.                        HttpServletResponse response)
11.        throws ServletException, IOException {
12.        // TO DO ...
13.    }
14. }
```

# Bài tập: Ứ.D DBĐT

## Viết servlet InputForm (tiếp)

```
1. //Process the HTTP Get request
2. public void doGet ( HttpServletRequest request, HttpServletResponse
   response)
3.             throws ServletException, IOException {
4.     response.setContentType(CONTENT_TYPE);
5.     PrintWriter out = response.getWriter(); // Write to client

6.     out.println("<html><head><title>InputForm</title></head>");
7.     out.println("<body>");
8.     out.println("<h1>Trang Web demo viec tra danh ba dien
   thoai.</h1>");
9.     out.println("<form action='searchaccount' method='post'>");
10.    out.println("Nhap ten thue bao: <INPUT NAME='ttbao'><br>");
11.    out.println("Nhap so dien thoai: <INPUT NAME='sodthoai'><br>");
12.    out.println("Nhap dia chi: <INPUT NAME='diachi'> <br>");
13.    out.println("<input type='submit' name='Submit' value='Bat dau
   tra'>");
14.    out.println("</form>");    out.println("</body></html>");
15. }
```

# Bài tập: Ứ.D DBĐT - Kết quả hiển thị trên trình duyệt

---



The screenshot shows a Microsoft Internet Explorer window titled "InputForm - Microsoft Internet Explorer". The address bar displays "http://localhost:8080/ServletDB/inputform". The main content area features a heading "Trang Web demo viec tra danh ba dien thoai." followed by a form with three input fields: "Nhap ten thue bao :", "Nhap so dien thoai : 848", and "Nhap dia chi :". A "Bat dau tra" button is located at the bottom of the form.

```
<html>
<head><title>InputForm</title></head>
<body> <h1>Trang Web demo viec tra danh ba dien thoai.</h1>
<form action=searchaccount method='post'>
    Nhap ten thue bao : <input name='ttbao' ><br>
    Nhap so dien thoai : <input name='sodthoai' ><br>
    Nhap dia chi      : <input name='diachi' ><br>
    <input type='submit' name='Submit' value='Bat dau tra'>
</form>
</body></html>
```

# Bài tập: Ứ.D DBĐT - Viết servlet SearchAccount

```
1. import javax.servlet.*; import javax.servlet.http.*;
2. import java.io.*; import java.util.*; import java.sql.*;

3. public class SearchAccount extends HttpServlet {

4.     static final private String CONTENT_TYPE = "text/html";
5.
6.     // Process the HTTP Post request
7.     public void doPost (HttpServletRequest request,
8.                         HttpServletResponse response)
9.         throws ServletException, IOException
10.    {
11.        // TODO...
12.        // Lay gia tri tu cac textbox tren InputForm
13.    }
14. }
```

# Bài tập: Ứ.D DBĐT

## Viết servlet SearchAccount (tiếp)

---

```
1. //Process the HTTP Post request
2. public void doPost (HttpServletRequest request, HttpServletResponse response)
3.     throws ServletException, IOException {
4.     response.setContentType(CONTENT_TYPE);
5.     PrintWriter out = response.getWriter();
6.     out.println("<html><head><title>SearchAccount</title></head>");
7.     out.println("<body><h1>Ket qua tra dien thoai theo yeu cau cua ban :</h1>");
8.     out.println("<table border=1 cellPadding=1 cellSpacing=1>");
9.     String tentbao = request.getParameter("ttbao");

10. // xay dung lenh SQL
11. String newSQL = "SELECT * FROM CUSTOMER";
12. if (tentbao != null && tentbao.length() != 0) {
13.     newSQL = newSQL + " where tentbao like '%" + tentbao + "%'";
14. }
15. String conStr = "jdbc:odbc:DanhBaDT";
16. Statement stmt = null;    ResultSet rs = null;
17. ....
```

# Bài tập: Ứ.D DBĐT

## Viết servlet SearchAccount (tiếp)

```
1.  try {    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
2.      con = DriverManager.getConnection(conStr, "", "");
3.      stmt = con.createStatement();
4.      rs = stmt.executeQuery(newSQL);
5.
6.      out.println("<tr><th>So thu tu</th><th>Ten thue bao</th><th>So dien" +
7.                  " thoai</th><th>Dia chi</th></tr>\n");
8.      if (rs != null) {
9.          for ( int i = 1; rs.next(); ) {
10.              out.println("<tr><td>" + i + "</td><td>" + rs.getString(2) +
11.                          "</td><td>" + rs.getString(3) + "</td><td>" +
12.                          rs.getString(4) + "</td></tr>\n");
13.          } // end for
14.      } // end if
15.      out.println("</table>");
16.      rs.close();      stmt.close();      con.close();
17.  } catch (Exception e) {      System.out.println("Error : " + e);      }
18.  ...
19. } // End doPost()
```

# Bài tập: Ứ.D DBĐT

## Kết quả chạy thử (tiếp)

**InputForm - Microsoft Internet Explorer**

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address http://localhost:8080/ServletDB/inputform

---

### Trang Web demo viec tra danh ba dien thoai.

Nhap ten thue bao :

Nhap so dien thoai :

Nhap dia chi :

**SearchAccount - Microsoft Internet Explorer**

File Edit View Favorites Tools Help

Address http://localhost:8080/ServletDB/searchaccount

---

### Ket qua tra dien thoai theo yeu cau cua ban :

So thu tu	Ten thue bao	So dien thoai	Dia chi
1	Xi Muoi	08-8481234	123 Ly Thuong Kiet
2	Nguyen Quang Hung	84-8481235	234 Lac Long Quan

[Quay ve trang chu](#)



# Bài tập

---

- ☐ Viết code các demo ví dụ (kết hợp tạo class java với package)
- ☐ Thực hiện bài tập viết ứng dụng Login dùng servlet.

# Q&A

# Thank you!