

## Chương 12

# Lập trình Web chạy ở server bằng Java

### 12.0 Dẫn nhập

### 12.1 Tổng quát về lập trình Web chạy ở server

### 12.2 Qui trình định nghĩa 1 DSN

### 12.3 Các điểm cơ bản về lập trình Java Servlet

### 12.4 Thí dụ xây dựng 1 website dùng Servlet

### 12.5 Các điểm cơ bản về lập trình JSP

### 12.6 Thí dụ xây dựng 1 website dùng JSP

### 12.7 Các điểm cơ bản về lập trình JavaBean

### 12.8 Thí dụ xây dựng 1 website dùng JavaBean

### 12.9 Kết chương



## 12.0 Dẫn nhập

- ❑ Chương này sẽ trình bày thông tin tổng quát về lập trình Web chạy ở máy server. Sau đó chúng tôi tập trung giới thiệu 3 công nghệ lập trình Web bằng Java chạy ở server : đó là Servlet, JSP và JavaBean.
- ❑ Ứng với mỗi công nghệ lập trình, chúng tôi sẽ trình bày các tính chất cơ bản của nó rồi giới thiệu 1 thí dụ đơn giản ứng dụng công nghệ đó.



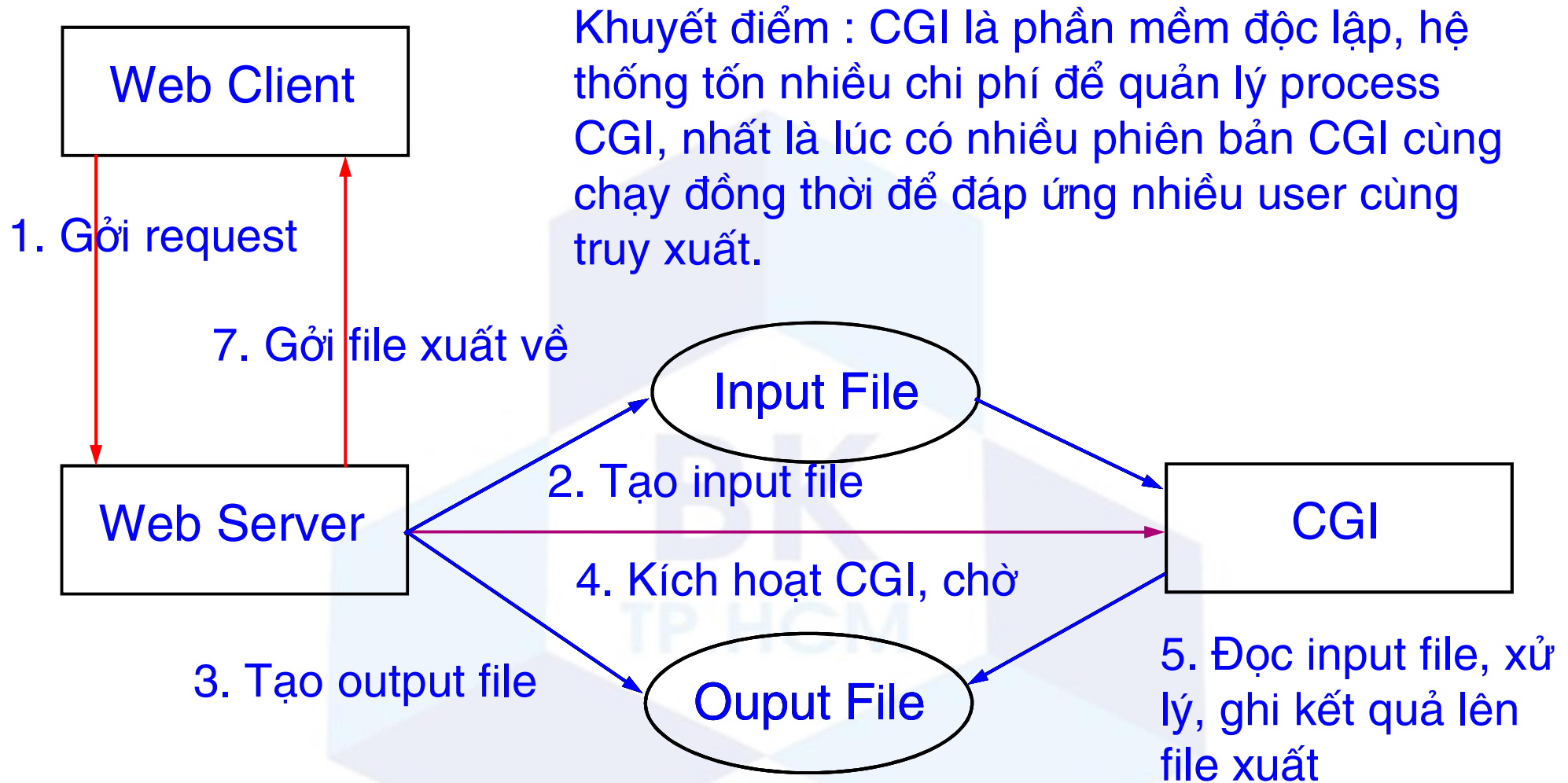
## 12.1 Tổng quát về lập trình Web chạy ở server

- ❑ Với các yếu điểm của lập trình Web chạy ở phía client được miêu tả trong chương 11, người ta thường chỉ dùng nó để xử lý sơ bộ quá trình tương tác và nhập liệu của người dùng, còn để xử lý các vấn đề còn lại, người ta dùng các công nghệ lập trình Web ở Server.
- ❑ Hiện có rất nhiều công nghệ lập trình Web chạy ở phía server khác nhau :

1. **CGI (Common Gateway Interface)** : là 1 phần mềm xử lý hàng lệnh truyền thống cho phép chuyển hướng (redirection) thiết bị nhập/xuất chuẩn. Mỗi lần Web server nhận yêu cầu từ client, nó chuẩn bị dữ liệu trong 1 file nhập, thiết lập file nhập và file xuất thành 2 thiết bị I/O chuẩn rồi kích hoạt CGI chạy. Trong quá trình chạy, CGI nhập dữ liệu từ file nhập và xuất dữ liệu ra file xuất do Web Server thiết lập. Sau khi CGI chạy xong, Web server sẽ lấy dữ liệu từ file xuất và gởi về client.



## 12.1 Tổng quát về lập trình Web chạy ở server

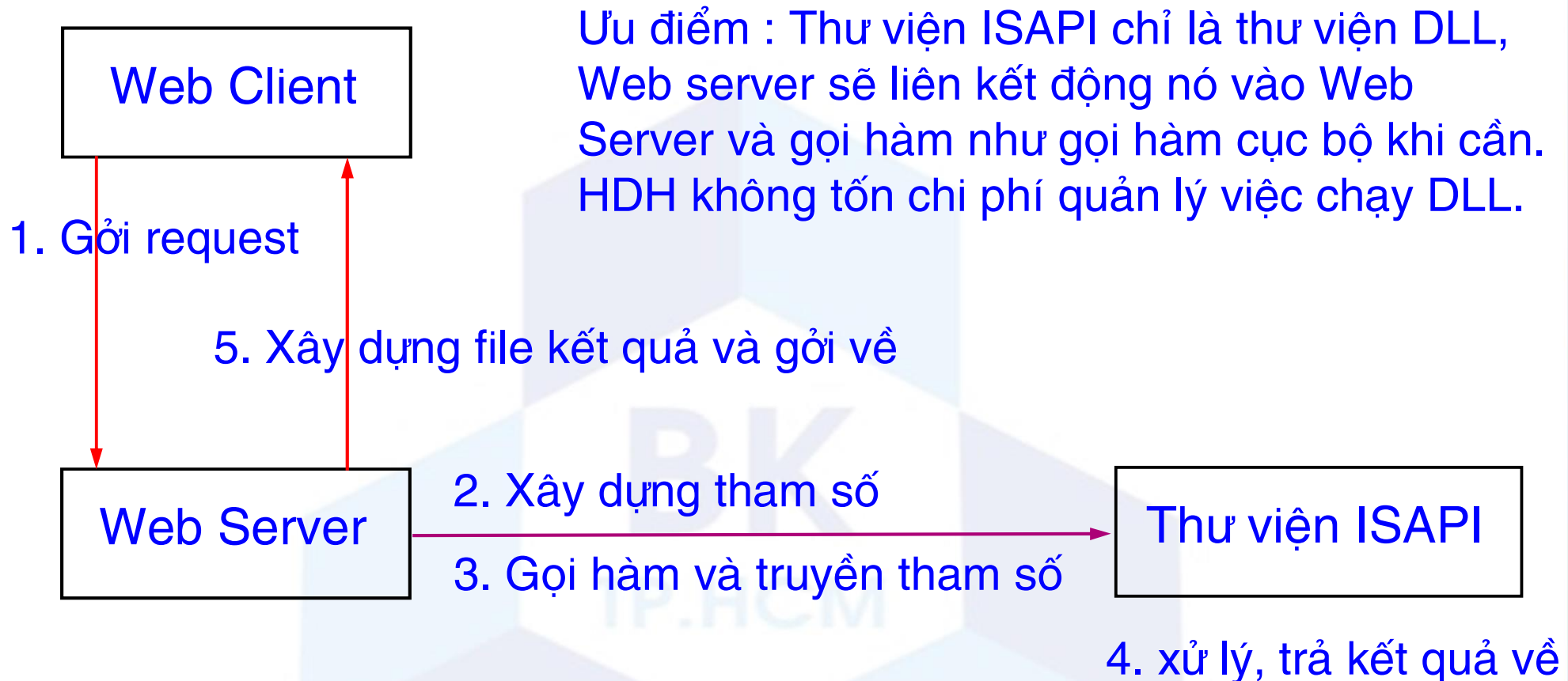


## 12.1 Tổng quát về lập trình Web chạy ở server

2. **ISAPI Extension** : mỗi thành phần xử lý được đóng gói thành 1 hàm và được để trong thư viện \*.dll. Mỗi lần Web server cần thành phần nào (do client yêu cầu), nó sẽ liên kết động với hàm đó và gọi hàm cục bộ  $\Rightarrow$  hệ điều hành ở máy server không tốn chi phí quản lý process như CGI.



## 12.1 Tổng quát về lập trình Web chạy ở server

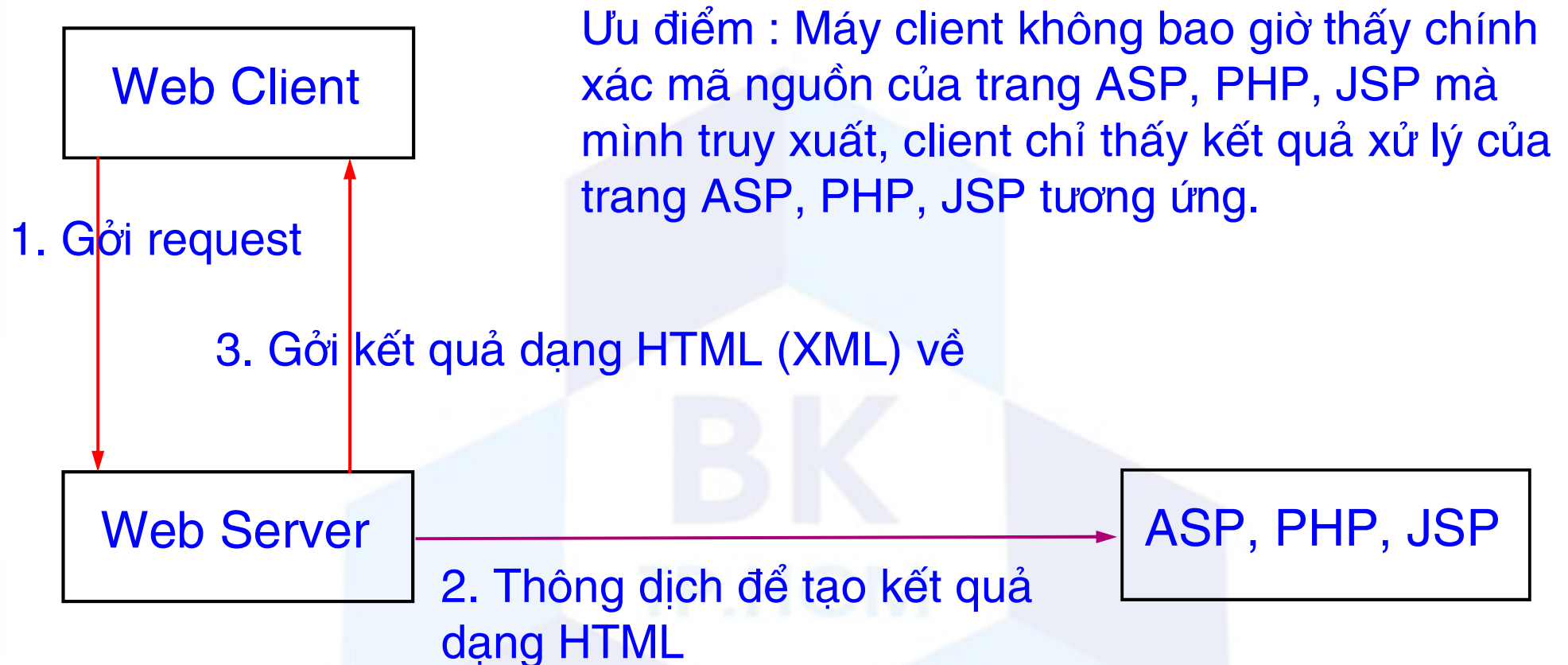


## 12.1 Tổng quát về lập trình Web chạy ở server

3. ASP, JSP, PHP,... : thành phần phần mềm là những đoạn script được nhúng thẳng vào trang Web nhưng sẽ được server chạy trước, kết quả thường được thay thế vào vị trí của thành phần phần mềm trước khi gửi về client.



## 12.1 Tổng quát về lập trình Web chạy ở server





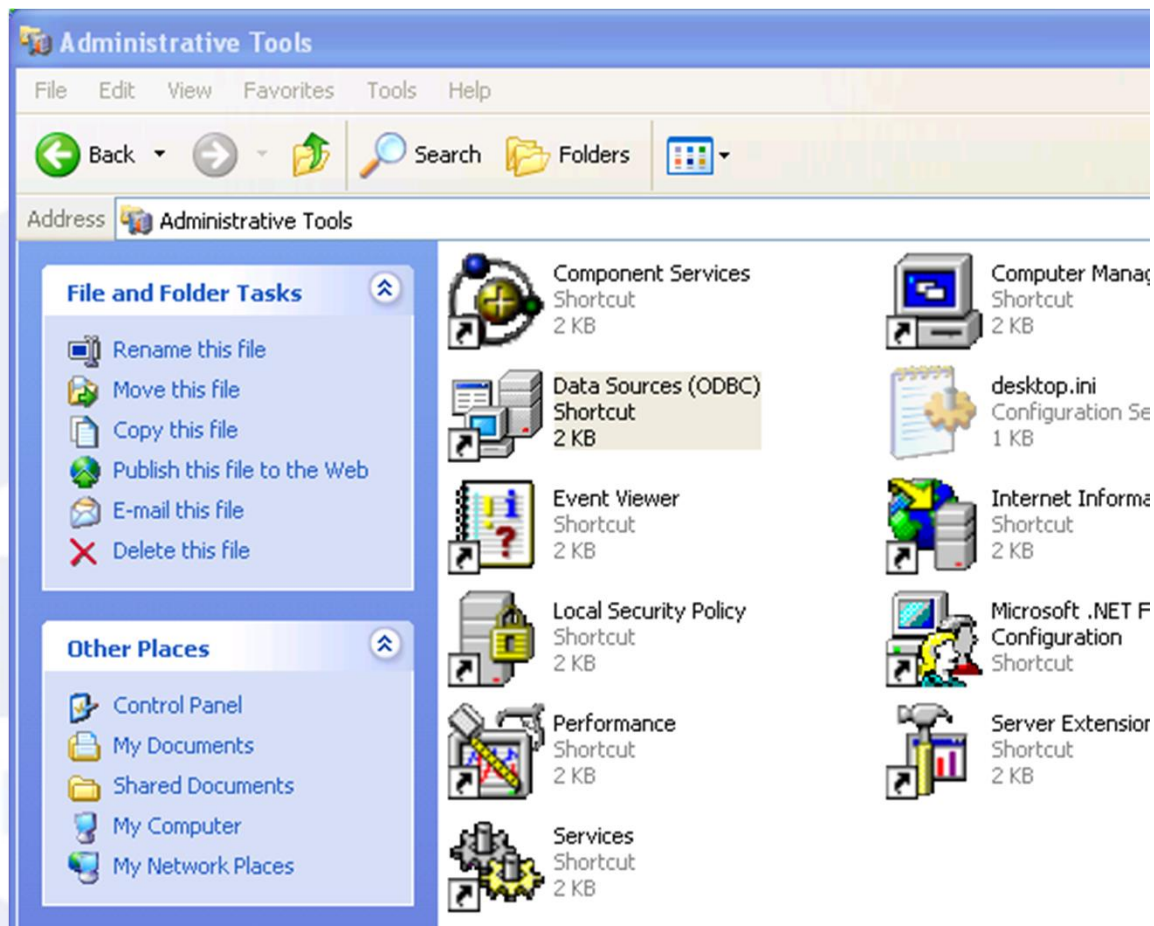
## 12.1 Tổng quát về lập trình Web chạy ở server

- ❑ Các công nghệ ASP, JSP, PHP,... dùng ngôn ngữ script để miêu tả chương trình  $\Rightarrow$  không đủ mạnh để giải quyết những vấn đề xử lý phức tạp  $\Rightarrow$  thường nhờ sự trợ giúp của các thành phần chuyên dụng khác :
  - ASP : dùng COM, \*.dll, Web Service.
  - JSP : dùng Servlet Java, JavaBean, Web Service.
  - PHP : dùng các thư viện hàm API viết sẵn.
- ❑ Tóm lại, thông qua dịch vụ Web, từ client, ta có thể kích hoạt chạy 1 ứng dụng hay 1 hàm trong thư viện nào đó ở máy server.
- ❑ Trong phần còn lại của chương này, chúng ta sẽ trình bày các công nghệ lập trình Web để chạy ở máy Server dùng ngôn ngữ Java, đó là công nghệ Servlet Java, JSP và JavaBean.



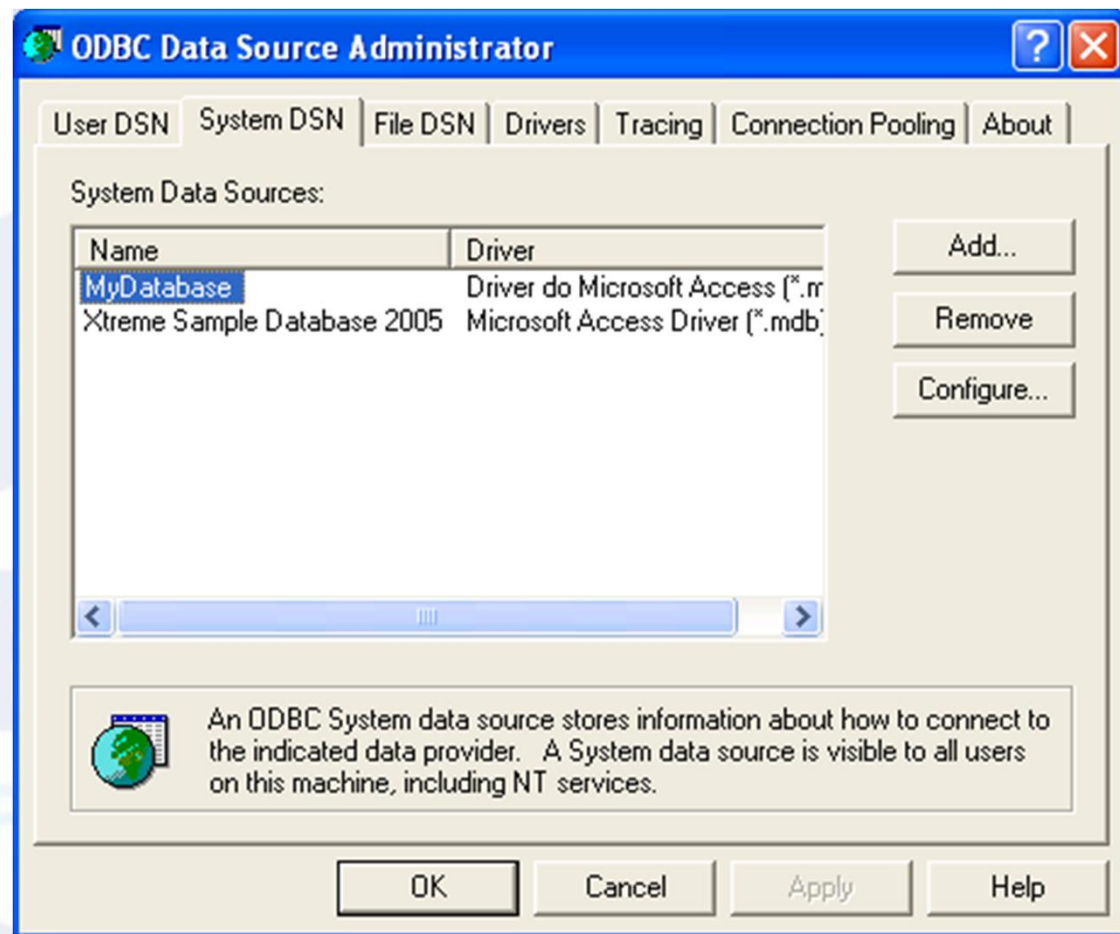
## 12.2 Qui trình định nghĩa 1 DSN

Các website mà chúng ta sẽ xây dựng đều có truy xuất database. Để code truy xuất database có tính tổng quát cao, không phụ thuộc vào công nghệ quản lý database và vị trí database, chúng ta sẽ dùng DSN (Data Source Name) để nhận dạng database cần truy xuất. Để định nghĩa 1 DSN trên Windows, ta vào Control Panel.Administrative Tools, cửa sổ bên sẽ hiển thị :



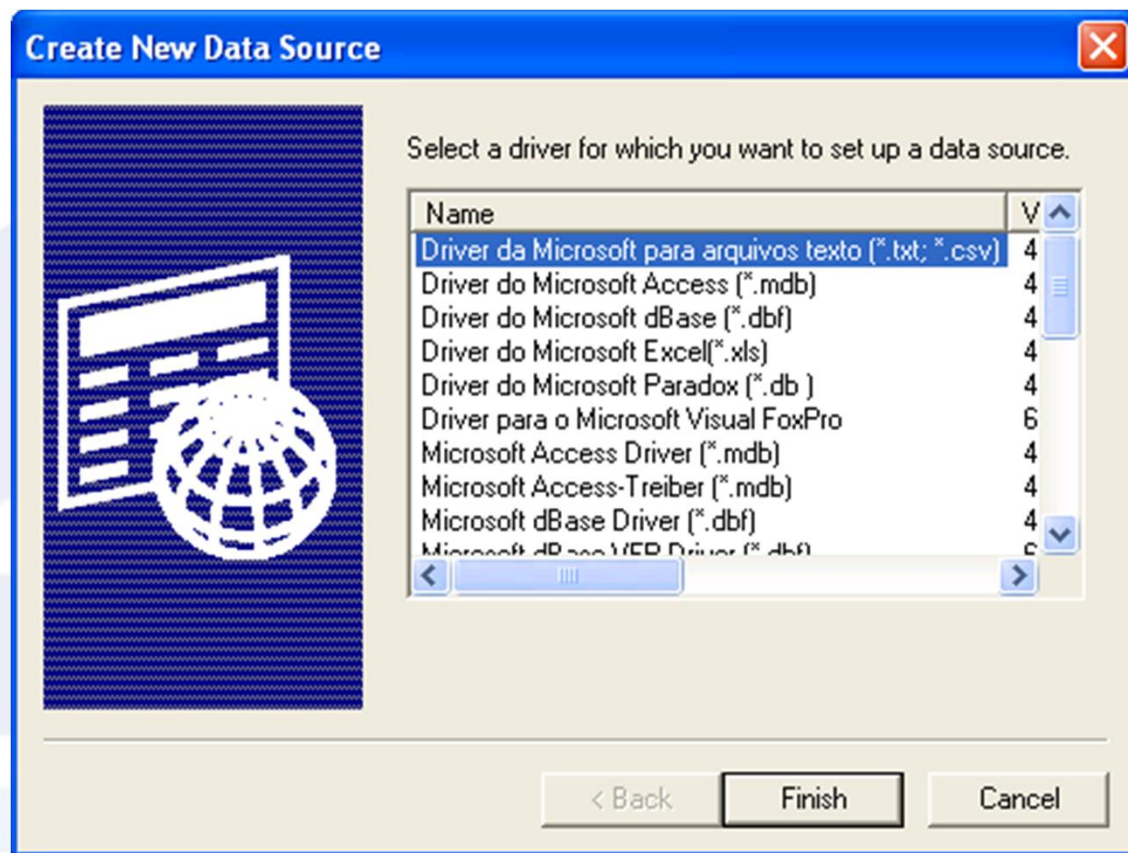
## 12.2 Qui trình định nghĩa 1 DSN

Ấn kép chuột vào mục Data Source (ODBC) để mở cửa sổ của tiện ích tương ứng như hình bên. Chọn tab “System DSN” để định nghĩa DSN có tầm vực truy xuất cấp hệ thống (bất kỳ ứng dụng nào cũng truy xuất được). Chọn button Add để thêm 1 DSN mới, cửa sổ Add sẽ hiển thị như slide kế.



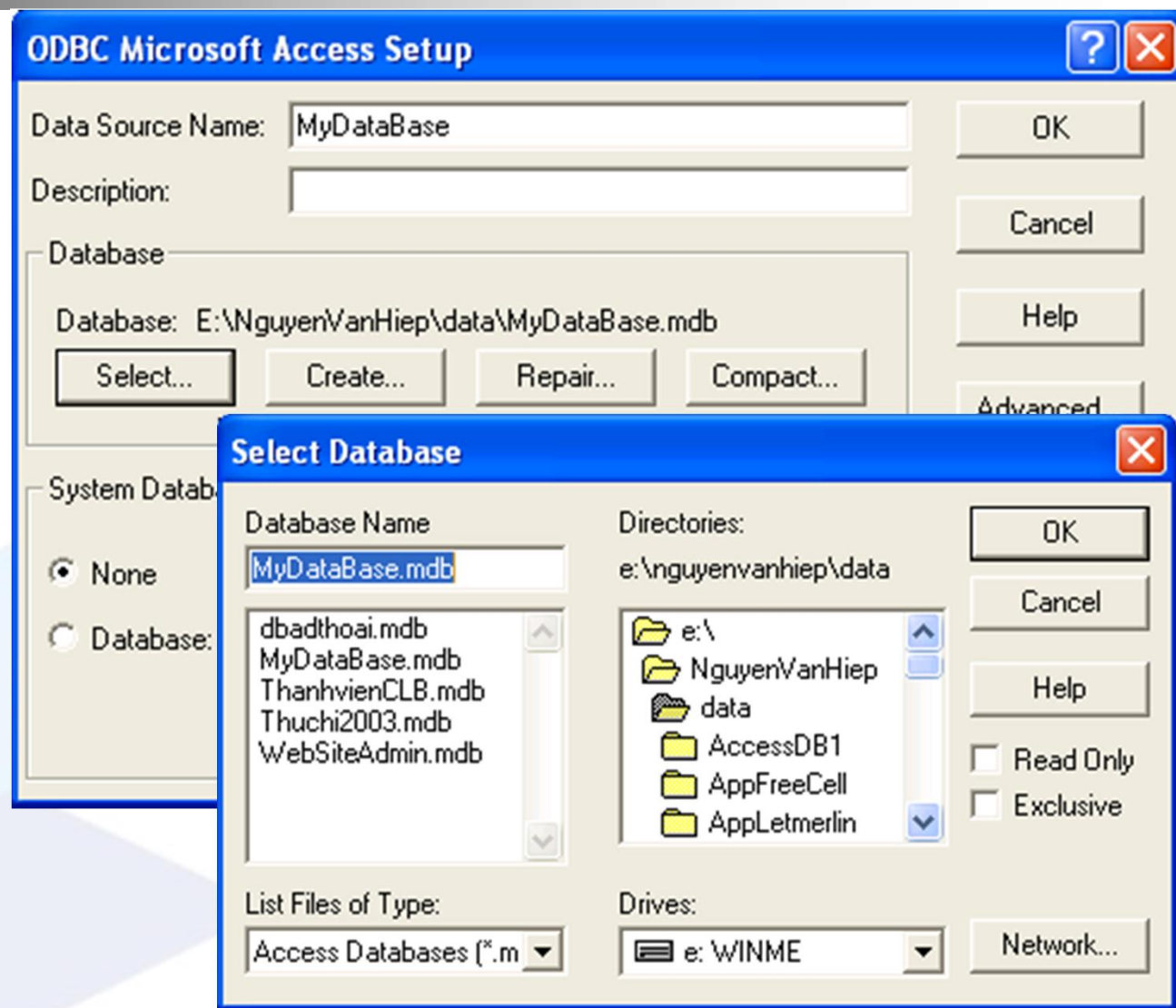
## 12.2 Qui trình định nghĩa 1 DSN

Duyệt tìm driver tương thích với hệ quản trị database và chọn nó. Cuối cùng chọn button Finish, cửa sổ của slide kế sẽ hiển thị.



## 12.2 Qui trình định nghĩa 1 DSN

Nhập tên nhận dạng DSN, chọn button Select để hiển thị cửa sổ Select Database, trên cửa sổ Select, duyệt tìm file database, ấn button OK để kết thúc qui trình Add DSN. Từ đây, bất kỳ ứng dụng nào cũng có thể truy xuất database với tên nhận dạng là MyDatabase.





## 12.3 Các điểm cơ bản về lập trình Java Servlet

- ❑ Servlet là đối tượng Java (giống như Applet), nhưng không có giao diện người dùng vì nó chạy trên máy server, nó giao tiếp với client thông qua việc gửi nhận request/response.
- ❑ Servlet có thể làm việc trên web server hỗ trợ các công nghệ Java như Tomcat, Apache, GlassFish...
- ❑ Viết 1 servlet cũng giống như 1 class Java bình thường khác, nhưng Servlet phải là class con của class **HttpServlet** và thường phải override 1 hay cả 2 method **doGet()** và **doPost()**.
- ❑ Có 2 khả năng khác nhau mà web browser hay 1 form giao diện trong trang Web ở máy client có thể dùng để gửi dữ liệu về Web server (tức về Servlet) đó là **GET** và **POST**. Tác vụ **doGet()** của Servlet sẽ được kích hoạt chạy khi client gửi yêu cầu theo cách GET, còn tác vụ **doPost()** sẽ được kích hoạt chạy khi client gửi yêu cầu theo cách POST.



## 12.3 Các điểm cơ bản về lập trình Java Servlet

### Template của 1 class servlet :

```
public class MyServlet extends HttpServlet {  
    ...  
    //override tác vụ xử lý HTTP Get request  
    public void doGet(HttpServletRequest request,  
        HttpServletResponse response)  
        throws ServletException, IOException { ... }  
    //override tác vụ xử lý HTTP Post request  
    public void doPost(HttpServletRequest request,  
        HttpServletResponse response)  
        throws ServletException, IOException { ... }  
    ...  
}
```



## 12.3 Các điểm cơ bản về lập trình Java Servlet

### Nhập/xuất thông tin của Java Servlet

- ❑ tham số request của 2 tác vụ doGet() & doPost() cho phép Servlet truy xuất thông tin được gửi về từ URL yêu cầu hay từ Form nhập liệu. Cách thức truy xuất tham số như sau :

```
String tentbao = request.getParameter("txtTenthuebao");
```

- ❑ tham số response của 2 tác vụ doGet() & doPost() cho phép Servlet xuất thông tin gửi về ứng dụng client. Cách thức xuất kết quả như sau :

```
//thiết lập định dạng kết quả là HTML
```

```
response.setContentType("text/html");
```

```
//tìm đối tượng phục vụ xuất kết quả
```

```
PrintWriter out = response.getWriter();
```

```
//xuất kết quả theo định dạng HTML
```

```
(xem tiếp trang kế)
```





## 12.4 Nhập/xuất thông tin của Java Servlet

### Nhập/xuất thông tin của Java Servlet

//tạo các tag HTML ở đầu trang web kết quả

```
out.println("<html>");
```

```
out.println("<head><title>Kết quả xử lý</title></head>");
```

//tạo tag BODY chứa nội dung cụ thể của trang Web kết quả

```
out.println("<body>");
```

//xử lý và xuất các kết quả

```
out.println("chuỗi kết quả");
```

...

//xuất tag liên kết về trang chủ chứa form nhập liệu

```
out.println("<a href='InputForm.html'>Quay về trang chủ</a>");
```

//xuất tag đóng BODY và đóng trang Web

```
out.println("</body></html>");
```



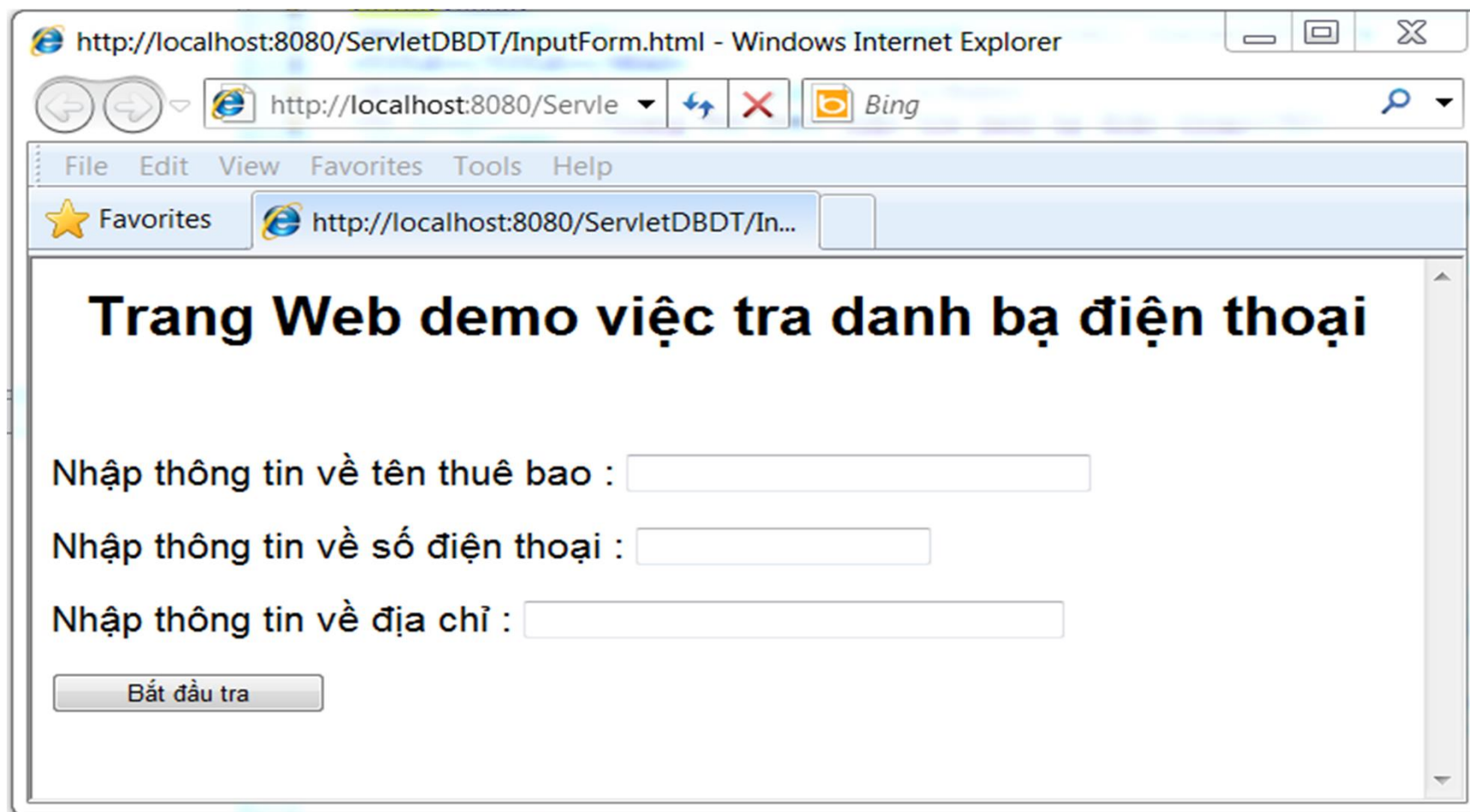
## 12.4 Thí dụ xây dựng 1 website dùng Servlet

- ❑ **Yêu cầu** : tạo 1 website dùng công nghệ Java Servlet cho phép người dùng tra danh bạ điện thoại bằng Web Browser.
- ❑ **Phân tích** : cần 2 module chức năng :
  - module chứa form nhập thông tin về các account cần tra. Module này có thể là 1 trang HTML thuần túy.
  - và module truy tìm trên database các account thỏa mãn tiêu chuẩn tìm kiếm, module này là 1 Servlet theo yêu cầu.
- ❑ **Qui trình thực hiện cụ thể** : ở đây ta dùng môi trường lập trình NetBeans 7.0.1. Qui trình điển hình cụ thể được liệt kê trong bài thực hành số 12.1



## 12.4 Thí dụ xây dựng 1 website dùng Servlet

- ❑ Cửa sổ trang Web giao tiếp người dùng :



## 12.4 Thí dụ xây dựng 1 website dùng Servlet

Xây dựng trang Web InputForm.html có nội dung sau :

```
<HTML><HEAD>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf8">
```

```
<TITLE></TITLE></HEAD>
```

```
<BODY><font face=Helvetica size="4">
```

```
<H2 align=center>Trang Web demo việc tra danh bạ điện thoại</H2>
```

```
<FORM id=FORM1 name=FORM1 action="ServletTraDBDT" method=post>
```

```
<P>&nbsp;</P>
```

```
<P>Nhập thông tin về tên thuê bao : <INPUT style="width: 244px; height: 22px" size=34 name=txtTenthuebao></P>
```

```
<P>Nhập thông tin về số điện thoại : <INPUT name=txtSodienthoai></P>
```

```
<P>Nhập thông tin về địa chỉ : <INPUT style="width: 284px; height: 22px" size=40 name=txtDiachi></P>
```

```
<P><INPUT style="width: 144px; height: 24px" type=submit size=45 value="Bắt đầu tra" name=submit1></P></FORM>
```

```
</font></BODY></HTML>
```



## 12.4 Thí dụ xây dựng 1 website dùng Servlet

Servlet tìm các account và tạo kết quả theo dạng sau :

**Kết quả tra điện thoại theo yêu cầu của bạn :**

Số thự tự	Tên thuê bao	Số điện thoại	Địa chỉ
1	Nguy?n V?n Hi?p	8644719	8A Lu Gia, P15, Q11, Tp.HCM
2	Hu?nh Tho?i Loan	9070438	245/4 Ly Thuong Kiet, P15, Q11, Tp.HCM
3	Nguy?n L??ng Hòa	9024437	240 Tran Binh Trong, P4, Q5, Tp.HCM
4	Nguy?n L?c Hân	2030478	157 Hoang Dieu, P1, Q1, Tp.HCM
5	Hu?nh Trác Luân	877056	Lu Gia
6	Nguy?n V?n M?t	877776	fggfgfgf
7	Nguy?n V?n Hai	0918729603	ghgfhfghfsdfdsfsdfsdfs
8	Nguy?n V?n Ba	0906999807	Xa l? quá

[Quay về trang chủ](#)



## 12.4 Thí dụ xây dựng 1 website dùng Servlet

Mã nguồn của Servlet ServletTraDBDT như sau :

```
package ServletDBDT; //định nghĩa package chứa servlet
import java.io.IOException; //import các package cần dùng
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import java.sql.*;
```

//định nghĩa chi tiết class Servlet

```
@WebServlet(name = "ServletTraDBDT", urlPatterns =
    {"/ServletTraDBDT"})
public class ServletTraDBDT extends HttpServlet {
    //định nghĩa các thuộc tính cần dùng cho Servlet
    private Connection con;
```





## 12.4 Thí dụ xây dựng 1 website dùng Servlet

//tác vụ xử lý yêu cầu HTTP Get

@Override

```
protected void doGet(HttpServletRequest request, HttpServletResponse  
response) throws ServletException, IOException {
```

//gọi tác vụ processRequest() thực hiện

```
processRequest(request, response);
```

```
}
```

//tác vụ xử lý yêu cầu HTTP Post

@Override

```
protected void doPost(HttpServletRequest request,  
HttpServletResponse response) throws ServletException, IOException {
```

//gọi tác vụ processRequest() thực hiện

```
processRequest(request, response);
```

```
}
```



## 12.4 Thí dụ xây dựng 1 website dùng Servlet

//tác vụ tìm các account theo yêu cầu

```
protected void processRequest(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    response.setContentType(CONTENT_TYPE);
    PrintWriter out = response.getWriter();
    //tạo các tag HTML ở đầu trang web kết quả
    out.println("<html>");
    out.println("<head><title>SearchAccount</title></head>");
    out.println("<body>");
    out.println("<h2>Kết quả tra điện thoại theo yêu cầu của bạn :</h2>");
    //tìm nội dung các tham số được truyền qua Servlet
    String tentbao = request.getParameter("txtTenthuebao");
    String sodthoai = request.getParameter("txtSodienthoai");
    String diachi = request.getParameter("txtDiachi");
```





## 12.4 Thí dụ xây dựng 1 website dùng Servlet

// xây dựng lệnh SQL tìm kiếm

```
boolean fsqlstart = true;
```

```
String newSQL = "SELECT * FROM danhbadienthoai";
```

```
if (tentbao != null && tentbao.length() != 0) {
```

```
    newSQL = newSQL + " where tenthuebao like '%" + tentbao + "%";
```

```
    fsqlstart = false;
```

```
}
```

```
if (sodthoai != null && sodthoai.length() != 0) {
```

```
    if (fsqlstart)
```

```
        newSQL = newSQL + " where sodienthoai like '%" + sodthoai + "%";
```

```
    else
```

```
        newSQL = newSQL + " and sodienthoai like '%" + sodthoai + "%";
```

```
    fsqlstart = false;
```

```
}
```



## 12.4 Thí dụ xây dựng 1 website dùng Servlet

```
if (diachi != null && diachi.length() != 0) {  
    if (fsqlstart)  
        newSQL = newSQL + " where diachi like '%" + diachi + "%'";  
    else  
        newSQL = newSQL + " and diachi like '%" + diachi + "%'";  
    fsqlstart = false;  
}  
//xây dựng chuỗi ConnectionString truy xuất database dùng DSN  
String conStr = "jdbc:odbc:MyDatabase";  
//định nghĩa các biến cần dùng  
java.sql.Statement stmt = null;  
ResultSet rs = null;
```



## 12.4 Thí dụ xây dựng 1 website dùng Servlet

```
try { //kết nối database, thực thi lệnh SQL, kết quả chứa trong rs
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    con = DriverManager.getConnection(conStr, "", "");
    stmt = con.createStatement();
    rs = stmt.executeQuery(newSQL);
    int i = 1;
    if (rs != null && rs.next()) {
        //nếu có ít nhất 1 account,
        //tạo bảng dữ liệu, hàng tiêu đề và hàng dữ liệu đầu tiên
        out.println("<table border=1 cellPadding=1 cellSpacing=1>");
        out.println("<tr><th>Số thứ tự</th><th>Tên thuê bao</th><th>Số  
điện thoại</th><th>Địa chỉ</th></tr>\n");
        out.println("<tr><td>" + i++ + "</td><td>" + rs.getString(2) +  
                    "</td><td>" + rs.getString(3) + "</td><td>" +  
                    rs.getString("diachi") + "</td></tr>\n");
    }
}
```



## 12.4 Thí dụ xây dựng 1 website dùng Servlet

```
while (rs.next()) { //lặp xuất từng hàng kết quả còn lại
    out.println("<tr><td>" + i++ + "</td><td>" + rs.getString(2) +
        "</td><td>" + rs.getString(3) + "</td><td>" +
        rs.getString("diachi") + "</td></tr>\n");
}
out.println("</table>");
} else out.println("<p>Không tìm được account nào thỏa yêu cầu
của bạn<br>");
}
catch (Exception e) {
    System.out.println("Error : " + e);
}
```



## 12.4 Thí dụ xây dựng 1 website dùng Servlet

```
//đóng các đối tượng đã dùng lại
finally {
    if (rs != null) {
        try { rs.close(); }
        catch (Exception _ex) { System.out.println("Exception: " + _ex); }
    }
    if (stmt != null) {
        try { stmt.close(); }
        catch (Exception _ex) { System.out.println("Exception: " + _ex); }
    }
}
//xuất tag liên kết về trang chủ chứa form nhập liệu
out.println("<a href='InputForm.html'>Quay về trang chủ</a>");
out.println("</body></html>");
}
```



## 12.5 Các điểm cơ bản về lập trình JSP

- ❑ JSP là công nghệ script chạy ở máy server của SUN tương tự với ASP của Microsoft hay PHP.
- ❑ Hiện JSP có thể chạy trên nhiều Web server khác nhau từ Tomcat, Apache, Jrun, JavaWebServer,...
- ❑ Một trang JSP là trang HTML có tăng cường 1 số đoạn script viết bằng Java.
- ❑ Để server nhận biết được file nào là JSP, ta luôn dùng extension đã qui định theo cấu hình (mặc định là \*.jsp).
- ❑ Đoạn script JSP được đóng trong tag `<% .... %>`.
- ❑ JSP thường dùng các đối tượng Java chính quy để xử lý các giải thuật phức tạp. Các đối tượng java này sẽ được đóng gói trong module JavaBean.



## 12.5 Các điểm cơ bản về lập trình JSP

- ❑ Trang JSP là trang HTML có tăng cường 1 số đoạn mã Java, mỗi đoạn mã java thuộc 1 trong 3 loại sau :
  1. Biểu thức có dạng `<%= expression %>`, khi chạy biểu thức được tính và kết quả sẽ được thay thế vào vị trí của mã nguồn.
  2. Đoạn script có dạng `<% code %>`, sẽ được thêm vào tác vụ `_jspService` của Servlet kết quả, tác vụ này sẽ được gọi bởi tác vụ service (là tác vụ đóng vai trò như tác vụ `doGet` hay `doPut`).
  3. Khai báo có dạng `<%! code %>`, sẽ được thêm vào thân của Servlet nhưng nằm ngoài bất kỳ tác vụ nào.
- ❑ Mỗi lần client truy xuất, trang JSP sẽ được dịch ra thành 1 Servlet Java trước khi được thực thi, kết quả thực thi của Servlet tương ứng sẽ được gửi về client.



## 12.5 Các điểm cơ bản về lập trình JSP

### Nhập/xuất dữ liệu trong trang JSP

- ❑ tham số request của 2 tác vụ doGet() & doPost() cho phép Servlet truy xuất thông tin được gửi về từ URL yêu cầu hay từ Form nhập liệu. Cách thức truy xuất tham số như sau :

```
String tentbao = request.getParameter("txtTenthuebao");
```

- ❑ tham số response của 2 tác vụ doGet() & doPost() cho phép Servlet xuất thông tin gửi về ứng dụng client. Cách thức xuất kết quả như sau :

'tìm đối tượng xuất

```
PrintWriter out = response.getWriter();
```

'xuất kết quả

```
out.println("chuỗi kết quả");
```

...





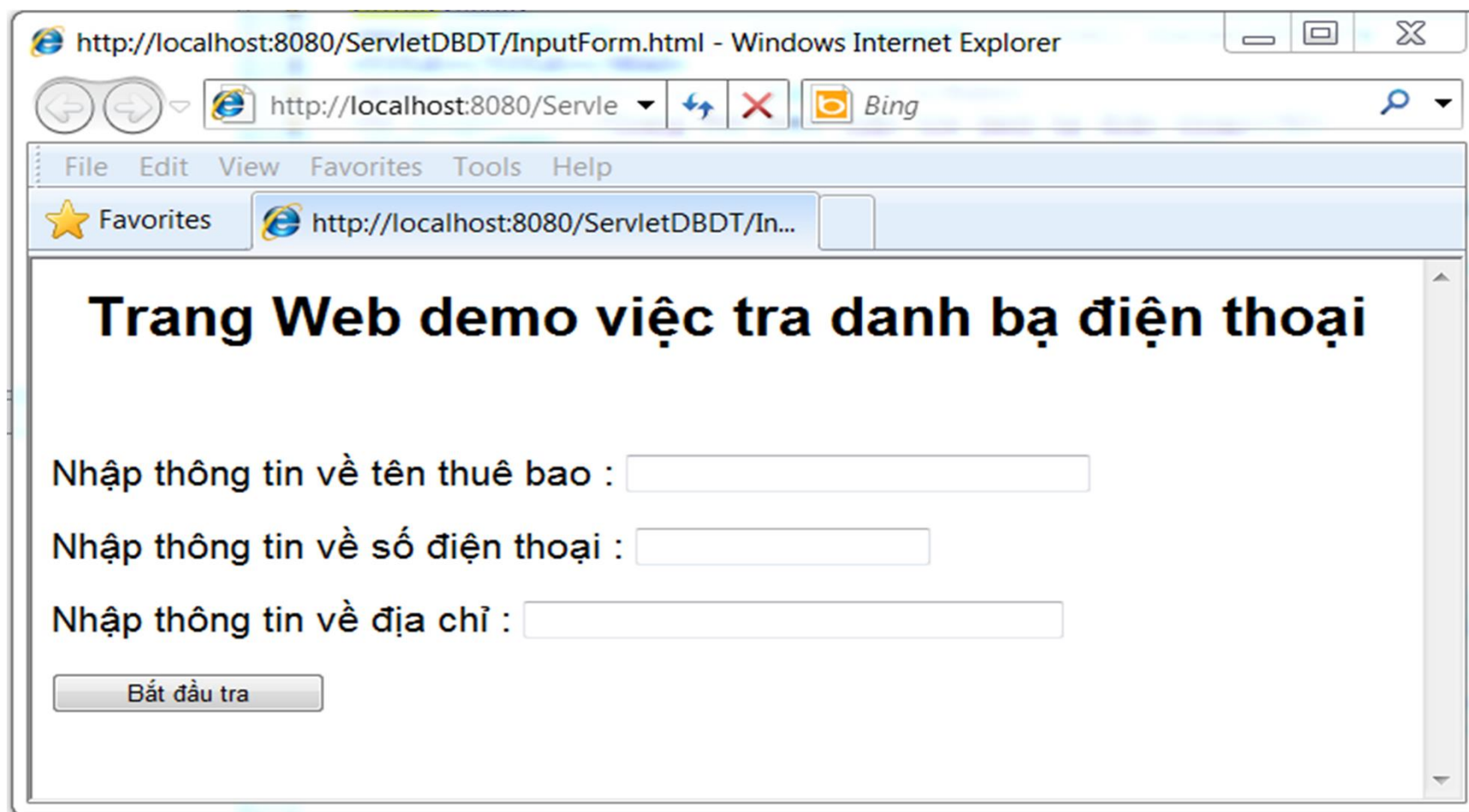
## 12.6 Thí dụ xây dựng 1 website dùng JSP

- ❑ **Yêu cầu** : tạo 1 website dùng công nghệ JSP cho phép người dùng tra danh bạ điện thoại bằng Web Browser.
- ❑ **Phân tích** : cần 2 module chức năng :
  - module chứa form nhập thông tin về các account cần tra. Module này có thể là 1 trang HTML thuần túy.
  - và module truy tìm trên database các account thỏa mãn tiêu chuẩn tìm kiếm, module này là 1 trang JSP theo yêu cầu.
- ❑ **Qui trình thực hiện cụ thể** : ở đây ta dùng môi trường lập trình NetBeans 7.0.1. Qui trình điển hình cụ thể được liệt kê trong bài thực hành số 12.2



## 12.6 Thí dụ xây dựng 1 website dùng JSP

- ❑ Cửa sổ trang Web giao tiếp người dùng (giống như mục 12.4) :



The screenshot shows a Windows Internet Explorer browser window. The address bar displays the URL `http://localhost:8080/ServletDBDT/InputForm.html`. The browser's menu bar includes File, Edit, View, Favorites, Tools, and Help. Below the menu bar is a Favorites bar with a star icon and the text 'Favorites'. The main content area of the browser displays the following HTML form:

**Trang Web demo việc tra danh bạ điện thoại**

Nhập thông tin về tên thuê bao :

Nhập thông tin về số điện thoại :

Nhập thông tin về địa chỉ :

## 12.6 Thí dụ xây dựng 1 website dùng JSP

Xây dựng trang Web InputForm.html có nội dung sau :

```
<HTML><HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=utf8">
<TITLE></TITLE></HEAD>
<BODY><font face=Helvetica size="4">
<H2 align=center>Trang Web demo việc tra danh bạ điện thoại</H2>
<FORM id=FORM1 name=FORM1 action="JSPTraDBDT.jsp" method=post>
<P>&nbsp;</P>
<P>Nhập thông tin về tên thuê bao : <INPUT style="width: 244px; height:
22px" size=34 name=txtTenthuebao></P>
<P>Nhập thông tin về số điện thoại : <INPUT name=txtSodienthoai></P>
<P>Nhập thông tin về địa chỉ : <INPUT style="width: 284px; height: 22px"
size=40 name=txtDiachi></P>
<P><INPUT style="width: 144px; height: 24px" type=submit size=45
value="Bắt đầu tra" name=submit1></P></FORM>
</font></BODY></HTML>
```



## 12.6 Thí dụ xây dựng 1 website dùng JSP

Servlet tìm các account và tạo kết quả theo dạng sau :

**Kết quả tra điện thoại theo yêu cầu của bạn :**

Số thự tự	Tên thuê bao	Số điện thoại	Địa chỉ
1	Nguy?n V?n Hi?p	8644719	8A Lu Gia, P15, Q11, Tp.HCM
2	Hu?nh Tho?i Loan	9070438	245/4 Ly Thuong Kiet, P15, Q11, Tp.HCM
3	Nguy?n L??ng Hòa	9024437	240 Tran Binh Trong, P4, Q5, Tp.HCM
4	Nguy?n L?c Hân	2030478	157 Hoang Dieu, P1, Q1, Tp.HCM
5	Hu?nh Trác Luân	877056	Lu Gia
6	Nguy?n V?n M?t	877776	fggfgfgf
7	Nguy?n V?n Hai	0918729603	ghgfhfghfsdfdsfsdfsd
8	Nguy?n V?n Ba	0906999807	Xa l? quá

[Quay về trang chủ](#)



## 12.6 Thí dụ xây dựng 1 website dùng JSP

Mã nguồn của trang JSP như sau :

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<%@ page import="java.sql.*" %>
```

```
<%!
```

```
public String SearchAccount (String tentbao, String sodthoai, String diachi)
```

```
{
```

```
    StringBuffer table = new StringBuffer();
```

```
    Connection con;
```

```
    // xây dựng lệnh SQL tìm kiếm theo yêu cầu người dùng
```

```
    boolean fsqlstart = true;
```

```
    String newSQL = "SELECT * FROM danhbadienthoai";
```

```
    if (tentbao != null)
```

```
        if (tentbao.length() != 0) {
```

```
            newSQL = newSQL + " where tenthuebao like '%" + tentbao + "%'";
```

```
            fsqlstart = false;
```

```
}
```



## 12.6 Thí dụ xây dựng 1 website dùng JSP

```
if (sodthoai != null)
    if (sodthoai.length() != 0) {
        if (fsqlstart) newSQL = newSQL + " where sodienthoai like '%" + sodthoai + "%'";
        else newSQL = newSQL + " and sodienthoai like '%" + sodthoai + "%'";
        fsqlstart = false;
    }
if (diachi != null)if (diachi.length() != 0) {
    if (fsqlstart) newSQL = newSQL + " where diachi like '%" + diachi + "%'";
    else newSQL = newSQL + " and diachi like '%" + diachi + "%'";
    fsqlstart = false;
}
//xây dựng chuỗi ConnectionString truy xuất database dùng DSN
String conStr = "jdbc:odbc:MyDatabase";
//định nghĩa các biến cần dùng
java.sql.Statement stmt = null;
ResultSet rs = null;
```





## 12.6 Thí dụ xây dựng 1 website dùng JSP

```
try {  
    // tạo connection đến database,  
    // thực hiện lệnh SQL, kết quả trả về trong recordset rs  
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  
    con = DriverManager.getConnection(conStr, "", "");  
    stmt = con.createStatement();  
    ResultSet rs = stmt.executeQuery(newSQL);  
    int i = 1;  
    if (rs != null && rs.next()) {  
        //nếu có account, tạo bảng, tạo tiêu đề và hàng miêu tả account đầu tiên  
        table.append("<table border=1 cellPadding=1 cellSpacing=1>");  
        table.append("<tr><th>Số tự tự</th><th>Tên thuê bao</th><th>Số điện thoại</th><th>Địa chỉ</th></tr>\n");  
        table.append("<tr><td>" + i++ + "</td><td>" + rs.getString(2) + "</td><td>" +  
rs.getString(3) + "</td><td>" + rs.getString("diachi") + "</td></tr>\n");  
    }  
}
```



## 12.6 Thí dụ xây dựng 1 website dùng JSP

```
//lặp xuất từng hàng thông tin account còn lại
while (rs.next()) {
    table.append("<tr><td>" + i++ + "</td><td>" + rs.getString(2) +
        "</td><td>" + rs.getString(3) + "</td><td>" +
        rs.getString("diachi") + "</td></tr>\n");
}
table.append("</table>");
} else
    table.append("<p>Không có account nào thỏa yêu cầu tìm
kiếm!</p>\n");
} catch (Exception e) {
    System.out.println("Error : " + e);
}
```





## 12.6 Thí dụ xây dựng 1 website dùng JSP

```
//đóng các đối tượng đã dùng lại
finally {
    if (rs != null) {
        try { rs.close(); }
        catch (Exception _ex) { System.out.println("Exception: " + _ex); }
    }
    if (stmt != null) {
        try { stmt.close(); }
        catch (Exception _ex) { System.out.println("Exception: " + _ex); }
    }
}
//trả kết quả về lệnh gọi
return table.toString();
}
%>
```



## 12.6 Thí dụ xây dựng 1 website dùng JSP

```
<title>
JSPTraDBDT
</title>
</head>
<body>
<h3>Kết quả tra danh bạ theo yêu cầu của bạn :</h3>
<%=SearchAccount(request.getParameter("txtTenthuebao"),request.getPa
    rameter("txtSodienthoai"),request.getParameter("txtDiachi"))%>
<p><a href="InputForm.html">Quay về trang chủ</a></p>
</body>
</html>
```



## 12.7 Các điểm cơ bản về lập trình JavaBean

- ❑ Như ta đã biết, trang JSP chỉ chứa các đoạn code được viết bằng ngôn ngữ javascript, do đó khó có thể diễn đạt tốt các thuật giải chức năng phức tạp trên nhiều kiểu dữ liệu khác nhau.
- ❑ Để tăng cường sức mạnh của các đoạn code javascript trong trang JSP, ta thường dùng các đối tượng Java chính quy để xử lý các giải thuật phức tạp. Các đối tượng java này sẽ được đóng gói trong module JavaBean.
- ❑ Mỗi đối tượng JavaBean được đặc tả bởi 1 class Java, nó cung cấp 1 số tác vụ chức năng mà bên ngoài nó (thường là trang JSP) sẽ gọi khi cần dùng.



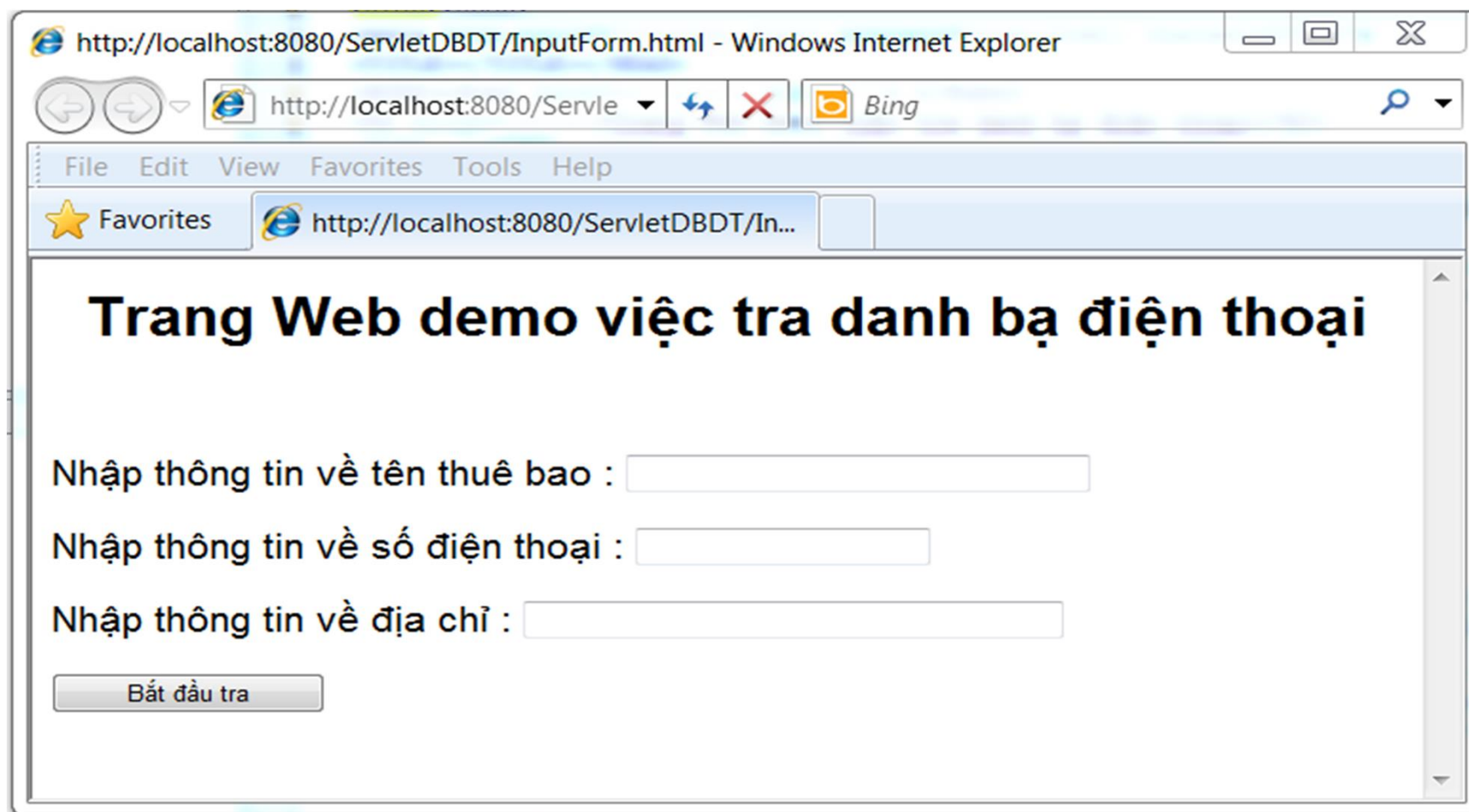
## 12.7 Thí dụ xây dựng 1 website dùng Javabeen

- ❑ **Yêu cầu** : tạo 1 website dùng công nghệ JavaBean cho phép người dùng tra danh bạ điện thoại bằng Web Browser.
- ❑ **Phân tích** : cần 3 module chức năng :
  - module chứa form nhập thông tin về các account cần tra. Module này có thể là 1 trang HTML thuần túy.
  - trang JSP nhận yêu cầu từ trang HTML.
  - và 1 JavaBean chứa các tác vụ truy tìm trên database các account thỏa mãn tiêu chuẩn tìm kiếm.
- ❑ **Qui trình thực hiện cụ thể** : ở đây ta dùng môi trường lập trình NetBeans 7.0.1. Qui trình điển hình cụ thể được liệt kê trong bài thực hành số 12.3



## 12.7 Thí dụ xây dựng 1 website dùng Javabeen

- ❑ Cửa sổ trang Web giao tiếp người dùng (giống như mục 12.4) :



## 12.7 Thí dụ xây dựng 1 website dùng Javabeen

Xây dựng trang Web InputForm.html có nội dung sau :

```
<HTML><HEAD>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf8">
```

```
<TITLE></TITLE></HEAD>
```

```
<BODY><font face=Helvetica size="4">
```

```
<H2 align=center>Trang Web demo việc tra danh bạ điện thoại</H2>
```

```
<FORM id=FORM1 name=FORM1 action="JSPTraDBDT.jsp" method=post>
```

```
<P>&nbsp;</P>
```

```
<P>Nhập thông tin về tên thuê bao : <INPUT style="width: 244px; height: 22px" size=34 name=txtTenthuebao></P>
```

```
<P>Nhập thông tin về số điện thoại : <INPUT name=txtSodienthoai></P>
```

```
<P>Nhập thông tin về địa chỉ : <INPUT style="width: 284px; height: 22px" size=40 name=txtDiachi></P>
```

```
<P><INPUT style="width: 144px; height: 24px" type=submit size=45 value="Bắt đầu tra" name=submit1></P></FORM>
```

```
</font></BODY></HTML>
```



## 12.7 Thí dụ xây dựng 1 website dùng Javabeans

Xây dựng trang JSPTraDBDT.jsp có nội dung sau :

```
<%@ page contentType="text/html; charset=UTF-8" %>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<html><head><title>JSPBeanDBDT</title></head>
<jsp:useBean id="JSPBeanDBDTId" scope="session"
    class="jspbeandbdt.JSPBeanDBDT" />
<jsp:setProperty name="JSPBeanDBDTId" property="*" />
<body bgcolor="#ffffff">
<h3>Kết quả tra danh bạ theo yêu cầu của bạn :</h3>
<%=JSPBeanDBDTId.SearchAccount(request.getParameter("txtTenth
    uebao"),request.getParameter("txtSodienthoai"),request.getParame
    ter("txtDiachi"))%>
<p><a href="InputForm.html">Quay về trang chủ</a></p>
</body>
</html>
```





## 12.8 Thí dụ xây dựng 1 website dùng JavaBean

Xây dựng class jspbeandbdt.JSPBeanDBDT có nội dung sau :

```
package jspbeandbdt;
import java.beans.*;
import java.io.Serializable;
import java.sql.*;
//class đặc tả JavaBean
public class JSPBeanDBDT implements Serializable {
    public String SearchAccount (String tentbao, String sodthoai, String
    diachi) {
        //hiện thực y như hàm này trong trang JSP của mục 12.7
    }
    //và hiện thực nhiều hàm chức năng khác nếu muốn
}
```



## 12.9 Kết chương

- ❑ Chương này đã trình bày thông tin tổng quát về lập trình Web chạy ở máy server. Chúng tôi đã tập trung giới thiệu 3 công nghệ lập trình Web bằng Java chạy ở server : đó là Servlet, JSP và JavaBean.
- ❑ Ứng với mỗi công nghệ lập trình, chúng tôi đã trình bày các tính chất cơ bản của nó rồi giới thiệu 1 thí dụ đơn giản ứng dụng công nghệ đó để giúp SV hiểu rõ hơn về cách sử dụng từng công nghệ.

