課題実験報告書

４年　情報工学科　６番

奥村　るい

【目的】

Java Mail を用いて、インターネットのホームページ上でメール受信をするサーブレットを作成する。

【概要】

・ Java

Java とは、サンマイクロシステムズが開発したプログラミング言語。

Java 特徴を以下に示す。

① オブジェクト指向型のプログラミング言語である。

② インタプリタ言語であり、マシンや OS を選ばない。

③ マルチスレッドで動作可能。

・ サーブレット

サーブレットとはサーバで動く Java のプログラムのこと。

クライアントの WEB ブラウザから要求があると、サーブレットのプログラムが HTML やその他のリソースを動的に生成して結果を WEB ブラウザに返す。

・ JSP

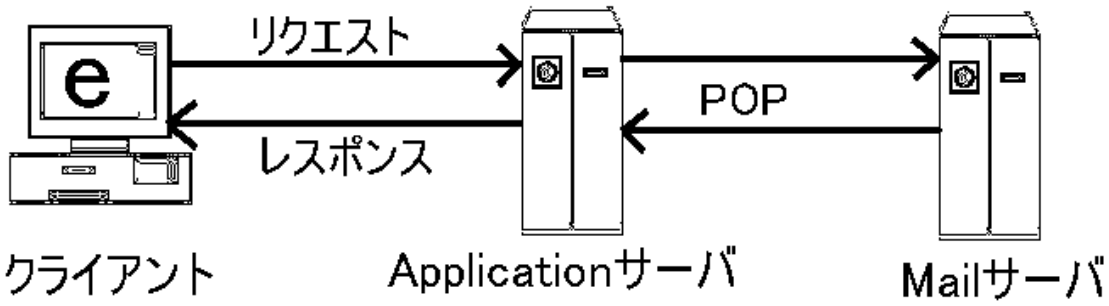
JSP(JavaServer Pages)とは HTML に Java コードを埋め込み Web サーバ側で動的なコンテンツを作成する仕組み。

ブラウザから要求された JSP ページは解釈・実行され、結果を HTML として出力する。メリットはコンパイルが不要なこと。

・ オブジェクト指向

オブジェクト指向とは「全ての事象や概念をオブジェクトととらえ、オブジェクトとオブジェクトとの間のメッセージのやり取りで全ての現象は進んでいく」という考え方。

【Mail 受信の流れ】



（1）クライアントは Mail 受信したいと Applicattion サーバにリクエストする。
（2）Applicattion サーバはプログラムを実行して、Mail サーバに Mail の転送を
要求する。ここでの通信プロトコルは POP3 を使用。
（3）Mail サーバは、Applicattion サーバに Mail を転送する。
ここでも通信プロトコルは POP3 を使用。
（4）Applicattion サーバは Mail を受け取り、HTML ファイルを作成して、クライアントに
レスポンスとして返す。
（5）クライアントのブラウザには、受信 Mail 一覧が表示される。
（6）一覧の中から１つ選んでクリックする。
（7）（1）～（4）を繰り返す。
（8）クライアントのブラウザには、選択された Mail の詳細（送信者、受信日時、
題名、etc…）が表示される。

【開発環境・動作環境】

・Tomcat　　4.1.12
・JavaMail　1.2
・JAF　　1.0.1
・JDK　　1.4.1
・RedHatLinux　7.2

・WEB サーバのスペック

CPU：300MHz
Memory：128MB
HDD：6GB

【必要なもの】

- Java 開発環境
- メールサーバ（POP3 サーバ）
- Java Server API に対応した Web サーバ
- mail.jar（JavaMail 本体）
- activation.jar（JAF:JavaBeans Activation Framework）
- pop3.jar（POP3 プロバイダー）

【実験内容】

（１） ＴＨＭＬのテストプログラムの作成。

（２） ＪＳＰを用いてメール受信プログラムの作成。

※ダウンロードや CLASSPATH の設定はすでにされていたので、今回は行っていない。

【Mail 受信プログラムの流れ】

ホストアドレス、アカウント、パスワードの設定

↓

接続

↓

フォルダを開く

↓

フォルダーにあるメッセージの数を取得

↓

メッセージを取得

↓

メッセージを表示

↓

フォルダーを閉じる

↓

メールの内容表示（送信者、受信日時、題名、本文 etc...）

【作成したプログラム】

```
<%@ page contentType="text/html; charset=Shift_JIS"
          import="java.util.*,java.io.*,javax.mail.*,javax.mail.internet.*"%>
<%@ page session="true"%>
<html>
<head>
<title>mail in</title>
</head>
<body>
<h1>mail in</h1>
<table border="1">
<tr>
<th>subject</th><th>from</th><th>date</th>
</tr>
<%
Properties prp=System.getProperties();
Session ses=Session.getDefaultInstance(prp);

String name =request.getParameter("name");          //パラメータの取得
String pass=request.getParameter("pass");
if(name.equals("")||pass.equals("")){
%>
<jsp:forward page="error.jsp" />            //エラーの表示

<%
}
Store str=ses.getStore("pop3");         //サーバへ接続するためのプロバイダの生成

str.connect("10.30.2.27",name,pass);

session.setAttribute("name",name);          //セッション変数の格納
session.setAttribute("pass",pass);

Folder fld=str.getDefaultFolder();          //フォルダの取得
```

```
fld=fld.getFolder("INBOX");                //INBOX フォルダの取得

fld.open(fld.READ_ONLY);

if(fld.getMessageCount() > 0){
   Message[] msg=fld.getMessages();        //メッセージの取得
   for(int i=0;i<msg.length;i++){
      Address[] from=msg[i].getFrom();
%>
  <tr>
   <td><a href="desc.jsp?num=<%=i+1 %>" target="_blank">
     <%=msg[i].getSubject() %></a></td>
   <td><%=MimeUtility.decodeText(from[0].toString())
        %></td>
   <td><%=msg[i].getSentDate() %></td>
   <td><%=msg[i].>
  </tr>

<%
  }

}else{
   out.println("<div style='color:Red;'> No message</div>");
}
fld.close(false);                  //フォルダのクローズ
str.close();                       //サーバへの接続を閉じる

%>
</table>
</body>
</html>
```

```
<%@ page contentType="text/html; charset=Shift_JIS"
        import="java.util.*,java.io.*,javax.mail.*,javax.mail.internet.*" %>

<%@ page session="true"%>
        <%
int num=Integer.parseInt(request.getParameter("num"));

Properties prp=System.getProperties();
Session ses=Session.getDefaultInstance(prp);
Store str=ses.getStore("pop3");
String name=(String)session.getAttribute("name");
String pass=(String)session.getAttribute("pass");

str.connect("10.30.2.27",name,pass);
Folder fld=str.getDefaultFolder();
fld=fld.getFolder("INBOX");
fld.open(fld.READ_ONLY);

Message msg=fld.getMessage(num);          //num 番目の Mail を取得

Address[] from=msg.getFrom();
%>

<html>
<head>
<title><%=msg.getSubject() %></title>
</head>
<body>
<table border="1">
<tr>
        <th>件名</th><td><%=msg.getSubject()%></td>
</tr><tr>
        <th>送信者</th><td><%=MimeUtility.decodeText(from[0].toString()) %></td>
</tr><tr>
```

```
        <th>送信日時</th><td><%=msg.getSentDate() %></td>
</tr><tr>

        <th valign="top">本文</th>
        <td><pre><%=msg.getContent() %></pre></td> // Mail の内容表示

</tr>
</table>
</body>
</html>
<%
fld.close(false);
str.close();
%>
```

・入力エラーを表示するプログラム

```
<%@ page contentType="text/html; charset=Shift_JIS" %>
<html>
<head>
<title>
error
</title>
</head>
<body>
<h1>
ユーザー名、またはパスワードが間違っています。
</h1>
</body>
</html>
```

【実験結果】

インターネットのホームページ上でメール受信をするサーブレットを作成できた。
また何も入力されなかったときに、エラー画面を表示することができた。

【考察】

今回の実験のメリットは、JSP を用いてプログラムを作成したので Applicattion サーバでサーブレットが自動的に生成される形となった。また、サーブレットでプログラムを作るより、JSP でプログラムを作成した方が見た目もわかりやすく、画面デザインの変更が容易になった。

前半の鈴木くんのプログラムとあわせると、Mail の送受信が可能になる。

今後の課題として、添付ファイルの送受信とエラー画面の修正が挙げられる。

【感想】

今回の実験の反省点は、まずテーマを決定するのに時間がかかってしまったことです。また Java についてあまりにも知識がなく、概要を理解するのにとても時間がかかりました。でも、HTML でレポートを作成したり JSP でプログラムを作成したことにより、普段授業では身につかない多くのことを学べたと思います。

【Q&A】

Q：JPS ってなんですか？

A： JSP は，Server Side Include（SSI）や Microsoft の Active Server Pages（ASP）に替わるテクノロジーとして開発されたもので，Java 言語を利用して Web サーバで動的に Web ページを生成し，クライアントに送信する技術である。
HTML タグに似た特別な JSP タグを使って Java コードを HTML 内に組み込むことができる。スクリプトとして Java を用い，サーバー上のクラスファイルを呼び出す形で複雑な Web アプリケーションを構築することができる。スクリプトが記述された JSP ページは，最初に呼び出された際にコンパイルされ，Servlet の形でメモリに常駐する。これにより，呼び出される度にプロセスが走り，システムのパフォーマンスが低下してしまうことを防ぐことができる。

Q：
Tomcat ってなんですか？

A：Tomcat とは、WEB サーバの機能を拡張する、サーブレット/JSP と呼ばれるプログラムの実行環境です。