**Java实现ftp功能**

import sun.net.ftp.\*;   
import sun.net.\*;   
import java.awt.\*;   
import java.awt.event.\*;   
import java.applet.\*;   
import java.io.\*;   
  
public class FtpApplet extends Applet   
{   
FtpClient aftp;   
DataOutputStream outputs ;   
TelnetInputStream ins;   
TelnetOutputStream outs;   
TextArea lsArea;   
Label    LblPrompt;   
Button   BtnConn;   
Button   BtnClose;   
TextField  TxtUID;   
TextField  TxtPWD;   
TextField  TxtHost;   
int ch;   
public String a="没有连接主机";   
        String hostname="";   
public void init () {   
setBackground(Color.white);   
setLayout(new GridBagLayout());   
GridBagConstraints GBC = new GridBagConstraints();   
LblPrompt = new Label("没有连接主机");   
LblPrompt.setAlignment(Label.LEFT);   
  
BtnConn = new Button("连接");   
BtnClose = new Button("断开");   
BtnClose.enable(false);   
TxtUID = new TextField("",15);   
TxtPWD = new TextField("",15);   
TxtPWD.setEchoCharacter(’\*’);   
TxtHost = new TextField("",20);   
Label LblUID = new Label("User ID:");   
Label LblPWD = new Label("PWD:");   
Label LblHost = new Label("Host:");   
  
lsArea = new TextArea(30,80);   
lsArea.setEditable(false);   
GBC.gridwidth= GridBagConstraints.REMAINDER;   
GBC.fill     = GridBagConstraints.HORIZONTAL;   
((GridBagLayout)getLayout()).setConstraints(LblPrompt,GBC);   
add(LblPrompt);   
  
GBC.gridwidth=1;   
((GridBagLayout)getLayout()).setConstraints(LblHost,GBC);   
add(LblHost);   
GBC.gridwidth=GridBagConstraints.REMAINDER;   
((GridBagLayout)getLayout()).setConstraints(TxtHost,GBC);   
add(TxtHost);   
  
GBC.gridwidth=1;   
((GridBagLayout)getLayout()).setConstraints(LblUID,GBC);   
add(LblUID);   
GBC.gridwidth=1;   
((GridBagLayout)getLayout()).setConstraints(TxtUID,GBC);   
add(TxtUID);   
  
GBC.gridwidth=1;   
((GridBagLayout)getLayout()).setConstraints(LblPWD,GBC);   
add(LblPWD);   
GBC.gridwidth=1;   
((GridBagLayout)getLayout()).setConstraints(TxtPWD,GBC);   
add(TxtPWD);   
  
GBC.gridwidth=1;   
GBC.weightx=2;   
((GridBagLayout)getLayout()).setConstraints(BtnConn,GBC);   
add(BtnConn);   
GBC.gridwidth=GridBagConstraints.REMAINDER;   
  
((GridBagLayout)getLayout()).setConstraints(BtnClose,GBC);   
add(BtnClose);   
  
GBC.gridwidth=GridBagConstraints.REMAINDER;   
GBC.fill     = GridBagConstraints.HORIZONTAL;   
((GridBagLayout)getLayout()).setConstraints(lsArea,GBC);   
add(lsArea);   
        }   
  
public boolean connect(String hostname, String uid,String pwd)   
{   
                this.hostname = hostname;   
LblPrompt.setText("正在连接，请等待.....");   
try{   
  aftp =new FtpClient(hostname);   
  aftp.login(uid,pwd);   
  aftp.binary();   
  showFileContents();   
}   
catch(FtpLoginException e){   
a="无权限与主机:"+hostname+"连接!";   
LblPrompt.setText(a);   
return false;   
}   
catch (IOException e){   
a="连接主机:"+hostname+"失败!";   
LblPrompt.setText(a);   
return false;   
}   
catch(SecurityException e)   
{   
a="无权限与主机:"+hostname+"连接!";   
LblPrompt.setText(a);   
return false;   
}   
LblPrompt.setText("连接主机:"+hostname+"成功!");   
return true;   
}   
  
public void stop()   
{   
try   
{   
aftp.closeServer();   
}   
catch(IOException e)   
{   
}   
}   
  
public void paint(Graphics g){   
}   
  
public boolean action(Event evt,Object obj)   
{   
if (evt.target == BtnConn)   
{   
LblPrompt.setText("正在连接，请等待.....");   
if (connect(TxtHost.getText(),TxtUID.getText(),TxtPWD.getText()))   
{   
BtnConn.setEnabled(false);   
BtnClose.setEnabled(true);   
}   
return true;   
}   
if (evt.target == BtnClose)   
{   
stop();   
BtnConn.enable(true);   
BtnClose.enable(false);   
LblPrompt.setText("与主机"+hostname+"连接已断开!");   
return true;   
}   
return super.action(evt,obj);   
}   
public boolean sendFile(String filepathname){   
boolean result=true;   
if (aftp != null)   
{   
LblPrompt.setText("正在粘贴文件,请耐心等待....");   
  
String  contentperline;   
try{   
a="粘贴成功!";   
String fg =new  String("\\");   
int index = filepathname.lastIndexOf(fg);   
String filename = filepathname.substring(index+1);   
File localFile ;   
localFile = new File(filepathname) ;   
RandomAccessFile sendFile = new RandomAccessFile(filepathname,"r");   
//   
sendFile.seek(0);   
outs = aftp.put(filename);   
outputs = new DataOutputStream(outs);   
while (sendFile.getFilePointer() < sendFile.length() )   
{   
  ch = sendFile.read();   
  outputs.write(ch);   
}   
outs.close();   
sendFile.close();   
}   
catch(IOException e){   
  a = "粘贴失败!";   
  result = false ;   
  
}   
LblPrompt.setText(a);   
showFileContents();   
}   
else{   
result = false;   
}   
return result;   
}   
  
public void showFileContents()   
{   
StringBuffer buf = new StringBuffer();   
lsArea.setText("");   
try   
{   
ins= aftp.list();   
while ((ch=ins.read())>=0){   
  buf.append((char)ch);   
}   
    lsArea.appendText(buf.toString());   
ins.close();   
        }   
catch(IOException e)   
{   
}   
}   
        public static void main(String args[]){   
             Frame f = new Frame("FTP Client");   
             f.addWindowListener(new WindowAdapter(){   
               public void windowClosing(WindowEvent e){   
                   System.exit(0);   
               }   
  
             });   
             FtpApplet ftp = new  FtpApplet();   
             ftp.init();   
             ftp.start();   
             f.add(ftp);   
             f.pack();   
             f.setVisible(true);   
        }   
}

**Java URL编程**

import java.io.\*;   
import java.net.\*;   
//   
//   
// GetHost.java   
//   
//   
public class GetHost    
{   
public static void main (String arg[]){   
if (arg.length>=1){   
InetAddress[] Inet;   
int i=1;   
try{   
for (i=1;i<=arg.length;i++){   
Inet = InetAddress.getAllByName(arg[i-1]);   
for (int j=1;j<=Inet.length;j++){   
System.out.print(Inet[j-1].toString());   
System.out.print("\n");   
}   
}   
}   
catch(UnknownHostException e){   
System.out.print("Unknown HostName!"+arg[i-1]);   
}   
}   
 else{   
System.out.print("Usage java/jview GetIp <hostname>");   
 }   
}   
}   
</pre>   
</p>   
<p>   
<pre>   
<font color=red>Example 2</font>   
<a href="GetHTML.java">download now</a>   
//GetHTML.java   
/\*\*   
 \* This is a program which can read information from a web server.   
 \* @version 1.0 2000/01/01   
 \* @author jdeveloper   
\*\*/   
import java.net.\*;   
import java.io.\*;   
  
public class GetHTML {   
public static void main(String args[]){   
if (args.length < 1){   
System.out.println("USAGE: java GetHTML httpaddress");   
System.exit(1);   
}   
String sURLAddress = new String(args[0]);   
URL    url = null;   
try{   
   url = new URL(sURLAddress);   
}catch(MalformedURLException e){   
   System.err.println(e.toString());   
                   System.exit(1);   
}   
try{   
                   InputStream ins = url.openStream();   
   BufferedReader breader = new BufferedReader(new InputStreamReader(ins));   
                   String info = breader.readLine();     
                   while(info != null){   
                        System.out.println(info);   
                        info  = breader.readLine();     
   }      
}   
                catch(IOException e){   
   System.err.println(e.toString());   
                   System.exit(1);   
}   
}   
}

**Java RMI编程**

<b>Step 1: Implements  the  interface of Remote Server as SimpleCounterServer.java</b>   
  
    public interface SimpleCounterServer extends java.rmi.Remote   
{   
    public int getCount() throws java.rmi.RemoteException;   
       
}   
Compile it with javac SimpleCounterServer.java   
  
<b>Step  2:      Produce the implement file SimpleCounterServerImpl.java as</b>   
import java.rmi.\*;   
import java.rmi.server.UnicastRemoteObject;   
//   
//   
// SimpleCounterServerImpl   
//   
//   
public class SimpleCounterServerImpl   
extends UnicastRemoteObject   
implements SimpleCounterServer   
{   
    private int iCount;   
    public SimpleCounterServerImpl() throws java.rmi.RemoteException   
    {   
        iCount = 0;   
    }   
    public int getCount() throws RemoteException   
    {   
        return ++iCount;   
    }   
  
    public static void main(String args[])   
    {   
        System.setSecurityManager(new RMISecurityManager());   
        try   
        {   
            SimpleCounterServerImpl server = new SimpleCounterServerImpl();   
            System.out.println("SimpleCounterServer created");   
            Naming.rebind("SimpleCounterServer",server);   
            System.out.println("SimpleCounterServer registered");   
        }   
        catch(RemoteException x)   
        {   
            x.printStackTrace();   
        }           
        catch(Exception x)   
        {   
            x.printStackTrace();   
        }   
    }   
}   
Complile it with javac SimpleCounterServerImpl.java   
  
<b>Step 3: Produce skeleton and stub file with rmic SimpleCounterServerImpl</b>   
            You will get two class files:   
                    1.SimpleCounterServerImpl\_Stub.class   
                 2.SimpleCounterServerImpl\_Skel.class   
  
<b>Step 4: start rmiregistry </b>   
  
<b>Step 5: java SimpleCounterServerImpl</b>    
<b>Step 6:  Implements the Client Applet to invoke the Remote method</b>   
    File :SimpleCounterApplet.java as   
    import java.applet.Applet;   
import java.rmi.\*;   
import java.awt.\*;   
//   
//   
// SimpleCounterApplet   
//   
//   
public class SimpleCounterApplet extends Applet   
{   
    String message="";   
    String message1 = "";   
    public void init()   
    {   
        setBackground(Color.white);   
        try   
        {   
            SimpleCounterServer sever = (SimpleCounterServer)   
                Naming.lookup("//"+getCodeBase().getHost()+"/SimpleCounterServer");   
            message1 = "//"+getCodeBase().getHost()+"/SimpleCounterServer";   
            message = String.valueOf(sever.getCount());   
        }   
        catch(Exception x)   
        {   
            x.printStackTrace();   
            message = x.toString();   
        }   
    }   
    public void paint(Graphics g)   
    {   
        g.drawString("Number is "+message,10,10);   
        g.drawString("Number is "+message1,10,30);   
    }   
}   
<b>step 7 create an Html File rmi.htm : </b>   
  
< html>   
< body>   
< applet code="SimpleCounterApplet.class" width="500" height="150">   
< /applet>   
< /body>   
< /html>

**Java CORBA入门**

Below is a simple example of a CORBA program   
download the source file    
  
<b>1. produce a idl file like this</b>   
   hello.idl   
   module HelloApp {   
     interface Hello    {            
         string sayHello();   
    };   
  };   
  
<b>2. produce stub and skeleton files through idltojava.exe</b>   
   idltojava hello.idl   
   idltojava is now named as idlj.exe and is included in the JDK.    
  
<b>3. write a server program like this </b>   
  
// HelloServer.java

银杏树 www.516fc.net

苏州美肤 www.szpfmr.net

苏州祛斑 www.szquban.net

苏州除皱 www.szchuzhou.com

苏州脱毛 www.sztuomao.com.cn

苏州人才网 www.xinyie.com

南京调查公司 www.jslycs.com

南京调查公司 www.penqiang.net  
     
import HelloApp.\*;   
  
import org.omg.CosNaming.\*;   
import org.omg.CosNaming.NamingContextPackage.\*;   
import org.omg.CORBA.\*;   
  
import java.io.\*;   
class HelloServant extends \_HelloImplBase    
{   
    public String sayHello()   
    {   
       return "\nHello world !!\n";    
    }      
     
}   
  
public class HelloServer {   
  
    public static void main(String args[])   
    {   
try{   
    // create and initialize the ORB   
    ORB orb = ORB.init(args, null);   
  
    // create servant and register it with the ORB   
    HelloServant helloRef = new HelloServant();   
    orb.connect(helloRef);   
  
    // get the root naming context   
    org.omg.CORBA.Object objRef =    
orb.resolve\_initial\_references("NameService");   
    NamingContext ncRef = NamingContextHelper.narrow(objRef);   
  
    // bind the Object Reference in Naming   
    NameComponent nc = new NameComponent("Hello", "");   
    NameComponent path[] = {nc};   
    ncRef.rebind(path, helloRef);   
  
    // wait for invocations from clients   
            java.lang.Object sync = new java.lang.Object();   
            synchronized (sync) {   
                sync.wait();   
            }   
  
} catch (Exception e) {   
    System.err.println("ERROR: " + e);   
    e.printStackTrace(System.out);   
}   
    }   
}       
  
<b>4. write a client program like this</b>   
// HelloClient.java   
import HelloApp.\*;   
import org.omg.CosNaming.\*;   
import org.omg.CORBA.\*;   
  
public class HelloClient    
{   
    public static void main(String args[])   
    {   
try{   
    // create and initialize the ORB   
    ORB orb = ORB.init(args, null);   
  
            // get the root naming context   
            org.omg.CORBA.Object objRef =    
orb.resolve\_initial\_references("NameService");   
            NamingContext ncRef = NamingContextHelper.narrow(objRef);   
            // test   
            System.out.println("OK..");                   
            // resolve the Object Reference in Naming   
            NameComponent nc = new NameComponent("Hello", "");   
            NameComponent path[] = {nc};   
            Hello helloRef = HelloHelper.narrow(ncRef.resolve(path));   
  
    // call the Hello server object and print results   
           //String oldhello = helloRef.lastMessage();   
           //System.out.println(oldhello);   
    String Hello = helloRef.sayHello();   
    System.out.println(Hello);   
  
} catch (Exception e) {   
    System.out.println("ERROR : " + e) ;   
    e.printStackTrace(System.out);   
}   
    }   
}   
  
<b>5. complie these files</b>   
  
   javac \*.java HelloApp/\*.java   
  
<b>6. run the application</b>   
      
  a. first you’ve to run the Name Service prior to the others likethis   
     c:\>tnameserv   
  b. run server   
     c:\>java HelloServer   
  c. run client    
     c:\>java HelloClient

**利用RamdonAccessFile来实现文件的追加**

RamdonAccessFile 是个很好用的类，功能十分强大，可以利用它的   
length()和seek()方法来轻松实现文件的追加，相信我下面这个例子是   
很容易看懂的，先写入十行，用length()读出长度（以byte为单位），   
在用seek()移动到文件末尾，继续添加，最后显示记录。   
  
import java.io.\*;   
public class IOStreamDemo {   
　public static void main(String[] args) {   
　　try{   
　　　RandomAccessFile rf1 = new RandomAccessFile("d:\\jeru.txt","rw");   
　　　for (int i = 0; i &lt; 10; i ++ )　{   
　　　　rf1.writeBytes("xixi,this is line "+i+"\n");   
　　　}   
　　　rf1.close();   
　   
　　　int i = 0;   
　　　String record = new String();   
　　　RandomAccessFile rf2 = new RandomAccessFile("d:\\jeru.txt","rw");   
　　　rf2.seek(rf2.length());   
　　　rf2.writeBytes("lala,append line"+"\n");   
　　　rf2.close();   
　　   
　　　RandomAccessFile rf3 = new RandomAccessFile("d:\\jeru.txt","r");   
　　　while ((record = rf3.readLine()) != null) {   
　　　　i ++;   
　　　　System.out.println("Value "+i+":"+record);   
　　　}   
　　　rf3.close();   
　 }catch(Exception e){}   
}   
}

**把JavaMail包装成简单的Sendmail**

package steeven;   
  
/\*   
Calss:  Sendmail   
Author: steeven@kali.com.cn   
QQ:     7461522   
Date:   11/13/2001   
Notes:   
        Find samples in main();   
        If you send mail only from/to intranet, change domain as u need.   
Setup:   
        1. change host, mailFooter   
\*/   
  
  
import java.util.\*;   
import java.sql.\*;   
import javax.mail.\*;   
import javax.mail.internet.\*;   
import javax.activation.\*;   
  
public class Sendmail extends Common{   
  
  public String host = "smtpServer";              //smtp server   
  static String mailFooter = "\n\n\n此邮件由WebMail发送 http://www/mail.jsp\n\n";   
  String domain = "company";   
  boolean sessionDebug;   
  
  Message msg;   
  Multipart multipart;   
  
  public Sendmail() throws Exception{   
    Properties props = System.getProperties();   
    props.put("mail.host", host);   
    props.put("mail.transport.protocol", "smtp");   
    Session session = Session.getDefaultInstance(props, null);   
    // Set debug on the Session so we can see what is going on   
    // Passing false will not echo debug info, and passing true   
    // will.   
    session.setDebug(sessionDebug);   
    // Instantiate a new MimeMessage and fill it with the   
    // required information.   
    msg = new MimeMessage(session);   
    msg.setSentDate(new java.util.Date());   
    multipart = new MimeMultipart();   
    msg.setContent(multipart);   
  }   
  public Sendmail(String to, String from, String subject, String body) throws Exception{   
    this();   
    this.setSendTo(to);   
    this.setFrom(from);   
    this.setSubject(subject);   
    this.setBody(body);   
  }   
  public Sendmail(String[] to, String from, String subject, String body) throws Exception{   
    this();   
    this.setSendTo(to);   
    this.setFrom(from);   
    this.setSubject(subject);   
    this.setBody(body);   
  }   
  public void setHost(String host) throws Exception{   
    this.host=host;   
  }   
  void setFrom(String from) throws Exception{   
    msg.setFrom(new InternetAddress(from+"@"+domain));   
  }   
  void setSendTo(String[] to) throws Exception{   
    for (int i=0;i&lt;to.length;i++)   
      setSendTo(to[i]);   
  }   
  void setSendTo(String to) throws Exception{   
//        System.out.println("sending mail to :"+to[i]);   
    InternetAddress[] address = {new InternetAddress(to + "@"+domain)};   
    msg.setRecipients(Message.RecipientType.TO, address);   
  }   
  void setCopyTo(String[] to) throws Exception{   
    for (int i=0;to!=null && i&lt;to.length;i++)   
      setCopyTo(to[i]);   
  }   
  void setCopyTo(String to) throws Exception{   
//        System.out.println("Copying mail to :"+to[i]);   
    InternetAddress[] address = {new InternetAddress(to +"@"+ domain)};   
    msg.setRecipients(Message.RecipientType.CC, address);   
  }   
  void setSubject(String subject) throws Exception{   
    sun.misc.BASE64Encoder enc = new sun.misc.BASE64Encoder();   
    msg.setSubject("=?GB2312?B?"+enc.encode(subject.getBytes())+"?=");   
  }   
  void setBody(String body) throws Exception{   
      MimeBodyPart messageBodyPart = new MimeBodyPart();   
      messageBodyPart.setText(body+mailFooter);   
      multipart.addBodyPart(messageBodyPart);   
  }   
  void setAttachFile(String fileName) throws Exception{   
    setAttachFile(fileName,fileName);   
  }   
  void setAttachFile(String fileName,String name) throws Exception{   
        MimeBodyPart messageBodyPart = new MimeBodyPart();   
        DataSource source = new FileDataSource(fileName);   
        messageBodyPart.setDataHandler(new DataHandler(source));   
        messageBodyPart.setFileName(name);   
        multipart.addBodyPart(messageBodyPart);   
  }   
  void setAttachFile(String[][] fileName) throws Exception{   
    for (int i=0;i&lt;fileName.length;i++)   
      setAttachFile(fileName[i][0],fileName[i][1]);   
  }   
  public void send() throws Exception{   
      Transport.send(msg);   
  }   
  public static void main(String[] args) throws Exception{   
    Sendmail s= new Sendmail("steeven","steeven","测试附件","This is text body");   
    s.setAttachFile("index.jsp");   
    s.send();   
    // or simply:   
    new Sendmail("steeven","steeven","mail subject","mail text body").send();   
  }   
}

**Hashtable简易数据库程序**

import java.io.\*;    
import java.util.\*;    
class str14    
{    
InputStreamReader stdin=new InputStreamReader(System.in);    
BufferedReader bufin=new BufferedReader(stdin);    
  
public static void main(String args[]) throws IOException    
{    
new str14();//run the application    
}    
  
public str14() throws IOException//constructor    
{    
Hashtable htable=new Hashtable(20,0.75F);    
dataBase(htable);    
}    
  
public str14(Hashtable htable) throws IOException //override the constructor    
{    
dataBase(htable);    
}    
  
public void dataBase(Hashtable htable) throws IOException    
{    
int count=htable.size();//get the ammount of the data in htable    
int value,id,num;    
String key,enter,data;    
Enumeration keys=htable.keys();//get all the key in htable    
Enumeration elements=htable.elements();//get all the elements in htable    
System.out.println("  
Hashtable 简易数据库程序  
");    
System.out.println("(1)输入数据");    
System.out.println("(2)请除所有数据");    
System.out.println("(3)显示单笔数据");    
System.out.println("(4)删除单笔数据");    
System.out.println("(5)显示所有数据");    
System.out.println("(6)结束程序");    
System.out.print("请输入您的选择:");    
value=select();//call select(),return the function number    
switch(value)//the function    
{    
case 1:    
System.out.print("  
请输入一笔数据:");//need data input    
data=bufin.readLine();    
count++;    
key=String.valueOf(count);    
htable.put(key,data);//store it to the htable    
System.out.print("  
输入完成，按任意键继续...");    
enter=bufin.readLine();    
new str14(htable);//reenter    
break;    
case 2://clear all data from the htable    
htable.clear();    
System.out.print("  
已删除了所有数据... 按任意键继续...");    
enter=bufin.readLine();    
new str14(htable);//reenter    
break;    
case 3:    
System.out.print("  
请输入要显示的数据编号:");    
id=getid(count);//call getid()    
key=String.valueOf(id);    
Object select=htable.get(key);//fetch the data from the htable    
data=select.toString();    
System.out.print("  
编号"+" "+"内容");    
System.out.println("  
 "+key+" "+data);//display the data    
System.out.print("  
按任意键继续...");    
enter=bufin.readLine();    
new str14(htable);//reenter    
break;    
case 4:    
System.out.print("请输入要删除的数据编号:");    
id=getid(count);    
key=String.valueOf(id);    
htable.remove(key);//remove data    
count--;    
num=count;    
System.out.print("已删除了所选择的数据...按任意见继续...");    
Hashtable htable1=new Hashtable(20,0.75F);//create new htable named htable1    
elements=htable.elements();//fetch all the data from htable    
while(elements.hasMoreElements())    
{    
key=String.valueOf(num);//a new value    
data=(String)elements.nextElement();//fetch content of the data    
htable1.put(key,data);//store it to htable1    
num--;    
}    
htable.clear();    
enter=bufin.readLine();    
new str14(htable1);    
break;    
case 5:    
String[] sortkey=new String[count];//create a new sort array    
String[] sortdata=new String[count];    
num=count;    
elements=htable.elements();    
keys=htable.keys();    
while(elements.hasMoreElements())    
{    
key=(String)keys.nextElement();//fetch value of key    
data=(String)elements.nextElement();//fetch value of data    
sortkey[num-1]=key;//store the value of key to sortkey array    
sortdata[num-1]=data;//store the value of data to sortdata array    
num--;    
}    
System.out.println("  
编号"+" "+"内容");    
for(int i=0;i<count;i++)//display all content after sorting    
System.out.println(" "+sortkey[i]+" "+sortdata[i]);    
System.out.print("  
目前共有"+count+"笔数据");    
System.out.print("  
  
按任意键继续...");    
enter=bufin.readLine();    
new str14(htable);    
break;    
default:    
}    
}    
  
public int select() throws IOException//method of getting a function selector    
{    
String input;    
int value=0;    
input=bufin.readLine();//read a input from keyboard    
try    
{    
value=Integer.parseInt(input);//convert a string to a int value    
}catch(NumberFormatException e)//can’t vonverted    
{    
System.out.print("请输入选项1~6:");    
value=select();    
}    
if(value>6||value<1)//if exceed then print a message and reenter    
{    
System.out.print("请输入选项1~6:");    
value=select();    
}    
return value;//return a value    
}    
  
public int getid(int count)throws IOException//a method of return the number of data    
{    
String input;    
int value=0;    
input=bufin.readLine();//read a user input string from keyboard    
try    
{    
value=Integer.parseInt(input);//convert the string to a int    
}catch(NumberFormatException e)//if can’t convert to a integer then reenter    
{    
System.out.print("请输入数据编号:");    
value=getid(count);    
}    
if(value>count)//the input value is out of bound    
{    
System.out.print("无此编号的数据,请重新输入:");    
getid(count);    
}    
return value;//return a value    
}    
}

**Java核心代码例程之：HelloWorld.java**

// The most basic Java program   
public class HelloWorld   
{   
   public static void main(String args[]) throws Exception   
   {   
       System.out.println("Hello World!");   
   }   
}

**Java核心代码例程之：StringBufferDemo.java**

/\*\*   
 \* Demo use of a StringBuffer   
 \*\*\*/   
public class StringBufferDemo   
{   
    public static void main(String args[]) throws Exception   
    {   
        StringBuffer sb = new StringBuffer();   
        sb.append("Hello, World!  
");   
        sb.append(123);   
        sb.append(", ");   
        sb.append(2.5);   
           
        System.out.println(sb.toString());   
    }   
}

**Java核心代码例程之：PropertiesDemo.java**

import java.util.\*;   
import java.io.\*;   
  
  
/\*\*   
 \* Demonstrates how to sort strings.   
 \*\*\*/   
public class PropertiesDemo   
{   
    public static void main(String args[]) throws Exception   
    {   
        Properties props = new Properties();   
        props.put("db.user", "jack");   
        props.put("db.password", "opendoor");   
           
        // getProperty will return the value if found, otherwise null   
        System.out.println(props.getProperty("db.user"));   
           
        // getProperty will return the value if found, otherwise "none"   
        System.out.println(props.getProperty("db.password", "none"));   
           
        /\*   
        // You can also load properties from a file   
        FileInputStream fis = new FileInputStream("my.properties");   
        props.load(fis);   
        fis.close();   
        \*\*/   
    }   
}

**Java核心代码例程之：VectorDemo.java**

import java.util.\*;   
  
  
/\*\*   
 \* Demonstrates how to use a Vector (essentially a growable array).   
 \*\*\*/   
public class VectorDemo   
{   
    public static void main(String args[]) throws Exception   
    {   
        Vector v = new Vector();   
           
        v.addElement("Mary");   
        v.addElement("John");   
        v.addElement("David");   
        v.addElement("Pam");   
        v.addElement("Joe");   
           
        for (int i=0; i < v.size(); i++)   
            System.out.println("Hello " + (String)v.elementAt(i));   
    }   
}

**Java核心代码例程之：DateFormat.java**

import java.text.\*;   
  
/\*\*   
 \* Demo date formatting.   
 \*\*\*/   
public class DateFormat   
{   
    public static void main(String args[]) throws Exception   
    {   
        SimpleDateFormat sdf = new SimpleDateFormat("MM.dd.yy kk.mm.ss.S");   
        System.out.println(sdf.format(new java.util.Date()));   
    }   
}

**代码例程之：RMIExampleServer.java**

import java.rmi.\*;   
import java.rmi.server.\*;   
  
/\*\*   
 \* This class is a simple example of a RMI Server   
 \* @author Renga   
 \*\*/   
public class RMIExampleServer extends UnicastRemoteObject    
implements RMIExample {   
  
/\*\*   
 \* Do nothing constructor   
 \* @throws RemoteException   
 \*\*/   
public RMIExampleServer() throws RemoteException {   
System.out.println( "RMIExampleServer::cntr" );   
}   
  
/\*\*   
 \* Returns "Hello World from RMI!"   
 \* @return A string   
 \* @throws RemoteException   
 \*\*/   
public String sayHello() throws RemoteException {   
return "Hello World from RMI!";   
}   
}

**Java核心代码例程之：CalendarDemo.java**

import java.util.\*;   
  
/\*\*   
 \* Demo Calendar class.   
 \*\*\*/   
public class CalendarDemo   
{   
    public static void main(String args[]) throws Exception   
    {   
        GregorianCalendar cal = new GregorianCalendar();   
        cal.setTime(new Date());   
           
        System.out.println((cal.get(Calendar.MONTH)+1) + "/" +    
                           cal.get(Calendar.DAY\_OF\_MONTH) + "/" +   
                           cal.get(Calendar.YEAR));   
    }   
}

**Java核心代码例程之：SimpleApplet.java**

import java.applet.\*;   
import java.awt.\*;   
  
  
/\*\* This demo uses JDK 1.0 event model for browser compatibility   
    Here is the supporting HTML code:   
    <applet code=SimpleApplet width=200 height=100></applet>   
 \*\*/   
public class SimpleApplet   
             extends Applet   
{   
    Label l  = new Label("Please click the button", Label.CENTER);   
    Button b = new Button("Click it!");   
       
    public void init()   
    {   
        setLayout(new BorderLayout());   
           
        add("Center", b);   
        add("South",  l);   
    }   
       
       
    public boolean action(Event event, Object what)   
    {   
        if (event.target == b)   
        {   
            l.setText("Hello World!");   
            return true;   
        }   
           
        return false;   
    }   
}

**Java核心代码例程之：StringToInt.java**

/\*\*   
 \* Demo how to convert int to String and back.  Similar   
 \* stuff can be done with the Float, Double, etc. classes   
 \* in java.lang   
 \*\*\*/   
public class StringToInt   
{   
    public static void main(String args[]) throws Exception   
    {   
        // From String to int   
        String sA="123";   
        int iA=Integer.parseInt(sA);   
           
        // From int to String   
        int iB=23;   
        String sB = String.valueOf(iB);   
    }   
}

**Java核心代码例程之：ThreadDemo.java**

/\*\*   
 \* Demo how threads in Java work   
 \*\*\*/   
public class ThreadDemo   
             extends Thread   
{   
    private int startingNumber=0;   
      
    public ThreadDemo(int startingNumber)   
    {   
        this.startingNumber=startingNumber;   
    }   
       
       
    // this method runs in a separate thread   
    public void run()   
    {   
        //loop 10 times   
        for (int i=startingNumber; i < (startingNumber+100); i++)   
            System.out.println(i);   
    }   
       
       
    // test driver   
    public static void main(String args[]) throws Exception   
    {   
        ThreadDemo t1=new ThreadDemo(100),   
                   t2=new ThreadDemo(200);   
                      
        t1.start();   
        t2.start();   
    }   
}

**Java核心代码例程之：SortDemo.java**

import java.util.\*;   
  
  
/\*\*   
 \* Demonstrates how to sort strings.   
 \*\*\*/   
public class SortDemo   
{   
    public static void main(String args[]) throws Exception   
    {   
        String names[] = new String[5];   
        int i=0;   
        names[i++] = "Mary";   
        names[i++] = "John";   
        names[i++] = "David";   
        names[i++] = "Pam";   
        names[i++] = "Joe";   
           
        Arrays.sort(names);   
               
        for (int x=0; x < names.length; x++)   
            System.out.println("Hello " + names[x]);   
    }   
}

**Java核心代码例程之：ShowFileContent.java**

import java.io.\*;   
  
// Read a file using BufferedReader.readLine()   
public class ShowFileContent   
{   
   public static void main(String args[]) throws Exception   
   {   
       FileReader fr = new FileReader("c:/autoexec.bat");   
       BufferedReader br = new BufferedReader(fr);   
       String line = null;   
          
       while ((line = br.readLine()) != null)   
          System.out.println(line);   
             
       br.close();   
   }   
}

**Java核心代码例程之：JDBC.java**

import java.sql.\*;   
  
/\*\*   
 \* Simple demo of how to perform a SQL query using JDBC.   
 \*\*/   
public class JDBC   
{   
   public static void main(String args[]) throws Exception   
   {   
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");   
         
      Connection conn = DriverManager.getConnection("jdbc:odbc:EmpPrjDB", "admin", "letmein");   
      Statement  stmt = conn.createStatement();   
      ResultSet  rslt = stmt.executeQuery("SELECT name, city FROM employee");   
  
      while (rslt.next())   
      {   
          System.out.println("Hello World from "   
                            + rslt.getString(1) // You can use getString(columnNumber)   
                            + " in "    
                            + rslt.getString("city")); // Or, you can use getString(columnName)   
      }   
  
      rslt.close();   
      stmt.close();   
      conn.close();   
   }   
}

**Java核心代码例程之：ProcessDemo.java**

import java.io.\*;   
  
  
/\*\*   
 \* Demonstrates how to run a program and read it"s output   
 \*\*\*/   
public class ProcessDemo   
{   
    public static void main(String args[]) throws Exception   
    {   
        Process process = Runtime.getRuntime().exec("ls");   
        InputStream standardOutput = process.getInputStream(),   
                    standardError  = process.getErrorStream();   
                      
        int c;   
        System.out.println("  
[ Standard Output ]");   
        while ((c = standardOutput.read()) != -1)   
           System.out.print((char)c);   
              
        System.out.println("  
[ Standard Error ]");   
        while ((c = standardError.read()) != -1)   
           System.out.print((char)c);   
              
        standardOutput.close();   
        standardError.close();   
    }   
}

**Java核心代码例程之：ServerSocketDemo.java**

import java.io.\*;   
import java.net.\*;   
  
  
/\*\*   
 \* Demonstrates how to write a Java Server   
 \*   
 \* @see ClientSocketDemo   
 \*\*\*/   
public class ServerSocketDemo   
{   
    public static void main(String args[]) throws Exception   
    {   
        // setup server on port 6000   
        ServerSocket ss = new ServerSocket(6000);   
        System.out.println(">> Waiting for client connection...");   
           
        // This can be placed in a while loop to serve multiple clients   
        Socket s = ss.accept();   
           
        // Print input   
        System.out.println(">> Connection received, reading data...");   
        BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream()));   
        PrintWriter pw = new PrintWriter(s.getOutputStream());   
           
        System.out.println(br.readLine());   
           
        System.out.println(">> Sending message back");   
        pw.println("Good Bye!");   
        pw.flush();   
           
        // close sockets   
        pw.close();   
        br.close();   
        ss.close();   
           
        System.out.println(">> Done!");   
    }   
}

**Java核心代码例程之：ClientSocketDemo.java**

import java.io.\*;   
import java.net.\*;   
  
  
/\*\*   
 \* Demonstrates how to write a Java client   
 \*   
 \* @see ServerSocketDemo   
 \*\*\*/   
public class ClientSocketDemo   
{   
    public static void main(String args[]) throws Exception   
    {   
        Socket s = new Socket("localhost", 6000);   
        PrintWriter pw = new PrintWriter(s.getOutputStream());   
        BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream()));   
           
        System.out.println(">> Connection with server established");   
        pw.println("Hello World!");   
        pw.flush();   
           
        // Print input   
        System.out.println(">> Reading output from server");   
        System.out.print(br.readLine());   
        System.out.println("  
>> Done!");   
           
        // close sockets   
        pw.close();   
        br.close();   
        s.close();   
    }   
}

**Java核心代码例程之：DatagramServerDemo.java**

import java.io.\*;   
import java.net.\*;   
  
  
/\*\*   
 \* Demonstrates how to write Datagram server   
 \*   
 \* @see DatagramClientDemo   
 \*\*\*/   
public class DatagramServerDemo   
{   
    public static void main(String args[]) throws Exception   
    {   
        byte Data[] = new byte[20];   
  
        // Client on port 4444, Server on 5555   
        DatagramSocket socket = new DatagramSocket(5555);   
        DatagramPacket packet = new DatagramPacket(Data, 20);   
  
        socket.receive(packet);   
        String received = new String(packet.getData(), 0);   
        System.out.println(received);   
  
        String sendMsg="Hello Back";   
        sendMsg.getBytes(0, sendMsg.length(), Data, 0);   
        packet  = new DatagramPacket(Data, 20,   
                                packet.getAddress(),   
                                packet.getPort());   
        socket.send(packet);   
    }   
}

**Java核心代码例程之：DatagramClientDemo.java**

import java.io.\*;   
import java.net.\*;   
  
  
/\*\*   
 \* Demonstrates how to Datagram client   
 \*   
 \* @see DatagramServerDemo   
 \*\*\*/   
public class DatagramClientDemo   
{   
    public static void main(String args[]) throws Exception   
    {   
        byte Data[] = new byte[20];   
        String sendMsg="Hello";   
        sendMsg.getBytes(0, sendMsg.length(), Data, 0);   
  
        // Client on port 4444, Server on 5555   
        DatagramSocket socket = new DatagramSocket(4444);   
        DatagramPacket packet = new DatagramPacket(Data, 6,   
                                    InetAddress.getByName("localhost"),   
                                    5555);   
        socket.send(packet);   
  
        packet = new DatagramPacket(Data, 20);   
        socket.receive(packet);   
        String received = new String(packet.getData(), 0);   
        System.out.println(received);   
    }   
}

**Java核心代码例程之：HelloWorld.java(2)**

import java.lang.reflect.Method;   
import java.lang.reflect.InvocationTargetException;   
  
/\*\*   
 \* <code>main()</code> invokes "Hello world" reflectively.   
 \*\*/   
  
public class HelloWorld   
{   
  public static void main( String[] args )   
  {   
    Class outClass = System.out.getClass();   
    Class[] argTypes = new Class[] { String.class };   
  
    try   
    {   
      Method printMethod = outClass.getMethod( "println", argTypes );   
      printMethod.invoke( System.out, new Object[] { "Hello World!" } );   
    }   
    catch ( NoSuchMethodException e )   
    {   
      System.err.println( "Can"t find method: " + e.getMessage() );   
    }   
    catch ( SecurityException e )   
    {   
      System.err.println( "Can"t find method: " + e.getMessage() );   
    }   
    catch ( IllegalAccessException e )   
    {   
      System.err.println( "Can"t invoke method: " + e.getMessage() );   
    }   
    catch ( IllegalArgumentException e )   
    {   
      System.err.println( "Can"t invoke method: " + e.getMessage() );   
    }   
    catch ( InvocationTargetException e )   
    {   
      System.err.println( "Can"t invoke method, threw: "    
                          + e.getTargetException().getMessage() );   
    }   
  }   
}

**Java核心代码例程之：JNIExample.java**

/\*\*   
 \* This class is a simple example that uses a native function   
 \* @author Renga   
 \*\*/   
public class JNIExample {   
  
static {   
System.loadLibrary( "JNI" );   
}   
  
/\*\*   
 \* Native method declaration. Just returns the string "Hello World from JNI!"   
 \* @return A string   
 \*\*/   
public native String sayHello();   
  
/\*\*   
 \* Main method   
 \* @param args Command-line arguments   
 \*\*/   
public static void main( String[] args ) {   
  
// Create a new JNIExample object   
JNIExample temp = new JNIExample();   
  
// Invoke the native method   
System.out.println( temp.sayHello() );   
}   
  
}

**Java核心代码例程之：JNIExample.cpp**

#include "JNIExample.h"   
  
/\*\*   
\* This method just returns the string "Hello World from JNI!" to the caller   
\* Parameters:   
\* env - JNI Environment pointer   
\* obj - Invoking object   
\* Returns:   
\* A string   
\*\*/   
extern "C" JNIEXPORT jstring JNICALL Java\_JNIExample\_sayHello   
( JNIEnv \*env, jobject obj ) {   
return env->NewStringUTF( "Hello World from JNI!" );   
}

**Java核心代码例程之：JNIExample.h**

/\* DO NOT EDIT THIS FILE - it is machine generated \*\*/   
#include <jni.h>   
/\* Header for class JNIExample \*\*/   
  
#ifndef \_Included\_JNIExample   
#define \_Included\_JNIExample   
#ifdef \_\_cplusplus   
extern "C" {   
#endif   
/\*   
\* Class:     JNIExample   
\* Method:    sayHello   
\* Signature: ()Ljava/lang/String;   
\*\*/   
JNIEXPORT jstring JNICALL Java\_JNIExample\_sayHello   
  (JNIEnv \*, jobject);   
  
#ifdef \_\_cplusplus   
}   
#endif   
#endif

**Java核心代码例程之：RMIClient.java**

import java.rmi.\*;   
  
/\*\*   
 \* This class is the client for RMIExampleServer   
 \* @author Renga   
 \*\*/   
public class RMIClient {   
  
/\*\*   
 \* Main method   
 \* @param args Command-line arguments   
 \*\*/   
public static void main( String[] args ) {   
try {   
// Set the security manager   
System.setSecurityManager( new RMISecurityManager() );   
  
// Look up RMIExampleServer in the RMI registry   
RMIExample server = (RMIExample) Naming.lookup( "RMIExample" );   
  
// Invoke the method   
System.out.println( server.sayHello() );   
} catch( Exception e ) {   
e.printStackTrace();   
}   
}   
}

**Java核心代码例程之：RMIExample.java**

import java.rmi.\*;   
  
/\*\*   
 \* Remote interface for the RMIExampleServer   
 \* @author Renga   
 \*\*/   
public interface RMIExample extends Remote {   
/\*\*   
 \* Returns a string to the caller   
 \* @return A string   
 \* @throws RemoteException   
 \*\*/   
public String sayHello() throws RemoteException;   
}

**Java核心代码例程之：RMIExampleSetup.java**

import java.rmi.\*;   
import javax.naming.\*;   
  
/\*\*   
 \* This class registers RMIExampleServer with the RMI registry   
 \* @author Renga   
 \*\*/   
public class RMIExampleSetup {   
  
/\*\*   
 \* Main method   
 \* @param args Command-line arguments   
 \*\*/   
public static void main( String[] args ) {   
try {   
// Set the security manager   
System.setSecurityManager( new RMISecurityManager() );   
  
// Create a new object of RMIExampleServer   
RMIExampleServer server = new RMIExampleServer();   
  
// Bind this object with the RMI registry   
Naming.bind( "RMIExample", server );   
} catch( Exception e ) {   
e.printStackTrace();   
}   
}   
}

**Java核心代码例程之：java.policy**

grant {   
permission java.security.AllPermission;   
};

**Java核心代码例程之：JSP----simple.jsp**

Date: <%= new java.util.Date() %>

**Java核心代码例程之：Servlet----SimpleServlet.java**

import java.io.\*;   
import javax.servlet.\*;   
import javax.servlet.http.\*;   
  
  
public class SimpleServlet   
       extends HttpServlet   
{   
    public void doGet(HttpServletRequest request, HttpServletResponse response)   
                throws ServletException, IOException   
    {   
        ServletOutputStream sos = response.getOutputStream();   
        sos.println("<h1>doGet: Hello World!</h1>");   
    }   
       
       
    public void doPost(HttpServletRequest request, HttpServletResponse response)   
                throws ServletException, IOException   
    {   
        ServletOutputStream sos = response.getOutputStream();   
        sos.println("<h1>doPost: Hello World!</h1>");   
    }   
       
  
    /\*   
    // Instead of calling doGet() or doPost(), service() can accomodate both    
    public void service(HttpServletRequest request, HttpServletResponse response)   
                throws ServletException, IOException   
    {   
    }   
    \*\*/   
}

**Java核心代码例程之：JavaMail ----PopMail.java**

import java.io.\*;   
import java.util.\*;   
  
import javax.mail.\*;   
import javax.mail.internet.\*;   
import javax.activation.\*;   
  
/\*\*   
 \* Demonstrate POP mail using JavaMail.  Requires mail.jar activation.jar   
 \* Sample run: java -cp .;mail.jar;activation.jar PopMail   
 \*\*/   
public class PopMail   
{   
   public static void main(String args[]) throws Exception   
   {   
        String pop3Host="myhost.com", popUser="john", popPassword="letmein";   
  
        Session session = Session.getDefaultInstance(System.getProperties(), null);   
        Store store = session.getStore("pop3");   
        store.connect(pop3Host, -1, popUser, popPassword);   
                   
  
        // Open the default folder   
        Folder folder = store.getDefaultFolder();   
        if (folder == null)   
            throw new NullPointerException("No default mail folder");   
  
        folder = folder.getFolder("INBOX");   
        if (folder == null)   
            throw new NullPointerException("Unable to get folder: " + folder);   
  
        // Get message count   
        folder.open(Folder.READ\_WRITE);   
        int totalMessages = folder.getMessageCount();   
        if (totalMessages == 0)   
        {   
            System.out.println("No messages found in inbox");   
            folder.close(false);   
            store.close();   
            return;   
        }   
           
        // Get attributes & flags for all messages   
        Message[] messages = folder.getMessages();   
        FetchProfile fp = new FetchProfile();   
        fp.add(FetchProfile.Item.ENVELOPE);   
        fp.add(FetchProfile.Item.FLAGS);   
        fp.add("X-Mailer");   
        folder.fetch(messages, fp);   
  
        // Process each message   
        for (int i = 0; i < messages.length; i++)   
        {   
            if (!messages[i].isSet(Flags.Flag.SEEN))       
                process(messages[i]);   
            //messages[i].setFlag(Flags.Flag.DELETED, true);    
        }   
           
        folder.close(true);   
        store.close();   
    }   
  
    private static void process(Message message)   
                 throws Exception   
    {   
        System.out.println("subject: " + message.getSubject()   
                          + ", sent: " + message.getSentDate()   
                          + ", size: " + message.getSize());   
    }   
}

**Java核心代码例程之：JavaMail----SmtpMail.java**

import java.io.\*;   
import java.util.\*;   
  
import javax.mail.\*;   
import javax.mail.internet.\*;   
import javax.activation.\*;   
  
  
public class SmtpMail   
{   
   public static void main(String args[]) throws Exception   
   {   
        // create some properties and get the default Session   
        Properties props = new Properties();   
        props.put("mail.smtp.host", "yoursmtpserver");   
  
        Session session    = Session.getDefaultInstance(props, null);   
        Message message = new MimeMessage(session);   
  
        // create a message   
        InternetAddress ia[] = new InternetAddress[1];   
        ia[0] = new InternetAddress("myself@abc.com", "Me");   
           
        message.setFrom(new InternetAddress("test@abc.com", "some developer"));   
        message.setRecipients(Message.RecipientType.TO, ia);   
        message.setSubject("Hello World");   
        message.setText("Hello World!  This message was generated using JavaMail.");   
        message.saveChanges();   
           
        Transport transport = session.getTransport("smtp");   
        transport.connect();   
        transport.send(message);   
    }   
}

**Java核心代码例程之：(JAXP) DOM**

import javax.xml.parsers.\*;   
import org.w3c.dom.\*;   
import org.xml.sax.\*;   
  
/\*\*   
 \* DOMDemo uses JAXP to acquire a DocumentBuilder to build a DOM Document from an XML file.   
 \* The example XML file represents a shopping cart.   
 \*   
 \* The following JARs must be in your CLASSPATH:   
 \* - jaxp.jar   
 \* - xerces.jar (for SAX parser and DOM object implementations)   
 \*   
 \* Download JAXP (which includes these JARs) here: http://java.sun.com/xml/   
 \* Find additional Xerces info here: http://xml.apache.org/   
 \*   
 \*\*/   
  
public class DOMDemo   
{   
  
  public static void main( String[] args )   
  {   
    try   
    {   
      DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();   
      System.out.println( "DocumentBuilderFactory classname: " + factory.getClass().getName() );   
  
      DocumentBuilder builder = factory.newDocumentBuilder();   
      System.out.println( "DocumentBuilder classname: " + builder.getClass().getName() );   
  
      //parse the XML file and create the Document   
      Document document = builder.parse( "cart.xml" );   
  
      /\*   
      At this point, all data in the XML file has been parsed and loaded into memory   
      in the form of a DOM Document object. The Document is a tree of Node objects.   
      This printNode() method simply recurses through a Node tree and displays info   
      about each node.\*\*/   
      printNode( document, "" );   
  
    }   
    catch( Exception e )   
    {   
      e.printStackTrace();   
    }   
  }   
  
  
  
  /\*\*   
   \* printNode is a recursive method that prints info about each Node   
   \* in a Node tree to System.out. Call it with the root Node of your Node tree and   
   \* an initial indent of ""   
   \*\*/   
  
  public static void printNode( Node node, String indent )   
  {   
    String text = null;   
  
    if( node.getNodeType() == Node.TEXT\_NODE )   
      text = node.getNodeValue().trim();   
    else   
      text = node.getNodeName();   
  
    if( text.length() > 0 )   
      System.out.println( indent + getNodeTypeName( node ) + ": " + text );   
  
    NodeList childNodes = node.getChildNodes();   
    for( int i = 0; i < childNodes.getLength(); i++ )   
      printNode( childNodes.item( i ), indent + "  " );   
  
  }   
  
  /\*\*   
   \* getNodeTypeName returns a String containing the type-name of the specified Node.   
   \*\*/   
  public static String getNodeTypeName( Node node )   
  {   
    switch( node.getNodeType() )   
    {   
      case Node.TEXT\_NODE:   
        return "TEXT";   
      case Node.ELEMENT\_NODE:   
        return "ELEMENT";   
      case Node.ATTRIBUTE\_NODE:   
        return "ATTRIBUTE";   
      case Node.ENTITY\_NODE:   
        return "ENTITY";   
      case Node.DOCUMENT\_NODE:   
        return "DOCUMENT";   
      case Node.CDATA\_SECTION\_NODE:   
        return "CDATA\_SECTION";   
      case Node.COMMENT\_NODE:   
        return "COMMENT";   
      case Node.NOTATION\_NODE:   
        return "NOTATION";   
      case Node.PROCESSING\_INSTRUCTION\_NODE:   
        return "PROCESSING INSTRUCTION";   
      case Node.DOCUMENT\_FRAGMENT\_NODE:   
        return "DOCUMENT FRAGMENT";   
      case Node.ENTITY\_REFERENCE\_NODE:   
        return "ENTITY REFERENCE";   
    }   
  
    return "UNKNOWN NODE TYPE";   
  }   
  
}

**Java核心代码例程之：(JAXP) XSL Transformation**

import javax.xml.transform.\*;   
import javax.xml.transform.stream.\*;   
  
/\*\*   
 \* TransformDemo uses JAXP to acquire an XML Transformer. It uses the Transformer   
 \* to transform an XML shopping cart into an HTML view of the shopping cart.   
 \* The Transformer uses transform instructions in an XSLT (.xsl) file.   
 \*   
 \* The following JARs must be in your CLASSPATH:   
 \* - jaxp.jar   
 \* - xerces.jar (for SAX parser and DOM object implementations)   
 \* - xalan.jar (for XSLT implementation)   
 \*   
 \* Download JAXP (which includes these JARs) here: http://java.sun.com/xml/   
 \* Find additional Xerces and Xalan info here: http://xml.apache.org/   
 \*   
 \* Note: XSLT authoring/programming is beyond the scope of this tutorial.   
 \* You"ll find good XSL info here: http://www.w3.org/Style/XSL/#learning   
 \*\*/   
  
public class TransformDemo   
{   
  
  public static void main( String[] args  )   
  {   
    try   
    {   
      TransformerFactory factory = TransformerFactory.newInstance();   
      System.out.println( "TransformerFactory classname: " + factory.getClass().getName() );   
  
      Transformer transformer = factory.newTransformer( new StreamSource( "cart.xsl" ) );   
      System.out.println( "Transformer classname: " + transformer.getClass().getName() );   
  
      //This single line applies the XSL file to transform the XML into HTML.   
      transformer.transform( new StreamSource( "cart.xml" ), new StreamResult( System.out ) );   
    }   
    catch( Exception e )   
    {   
      e.printStackTrace();   
    }   
  
 }   
  
}

**Java核心代码例程之：(JAXP) SAX**

import javax.xml.parsers.\*;   
import org.xml.sax.\*;   
  
  
/\*\*   
 \* SAXDemo uses JAXP to acquire a SAX parser to parse an XML file.   
 \* The example XML file represents a shopping cart.   
 \*   
 \* The following JARs must be in your CLASSPATH:   
 \* - jaxp.jar   
 \* - xerces.jar (for SAX parser implementation)   
 \*   
 \* Download JAXP (which includes these JARs) here: http://java.sun.com/xml/   
 \* Find additional Xerces info here: http://xml.apache.org/   
 \*   
 \* Note: Unlike DOM, SAX parsing does not load the XML file into memory.   
 \* SAX parsers traverse the XML file and report parse "events" to an event handler.   
 \*\*/   
  
 public class SAXDemo   
  extends org.xml.sax.HandlerBase   
{   
  
  /\*\*   
   \* main creates and runs a SaxTest instance.   
   \*\*/   
  public static void main( String[] args )   
  {   
    SAXDemo me = new SAXDemo();   
    me.run();   
  }   
  
  
  public void run()   
  {   
    try   
    {   
      SAXParserFactory factory = SAXParserFactory.newInstance();   
      log( "SAXParserFactory classname: " + factory.getClass().getName() );   
  
      SAXParser saxParser = factory.newSAXParser();   
      log( "SAXParser classname: " + saxParser.getClass().getName() );   
  
      /\*   
      The SAXParser.parse method initiates parsing of the XML file.   
      The second parameter specifies which class will handle parse events.   
      This class must extend org.xml.sax.HandlerBase   
      \*\*/   
      saxParser.parse( "cart.xml", this );   
    }   
    catch( Exception e )   
    {   
      e.printStackTrace();   
    }   
  }   
  
  /\*\*   
   \* log simply prints the specified message to System.out   
   \*\*/   
  public void log( String message )   
  {   
    System.out.println( message );   
  }   
  
  
  /\*\*   
   \* SAXParser calls startDocument() when it starts parsing a document   
   \*\*/   
  public void startDocument ()   
    throws SAXException   
  {   
    log( "Start SAX parse" );   
  }   
  
  /\*\*   
   \* SAXParser calls endDocument() when it finishes parsing a document   
   \*\*/   
  public void endDocument ()   
    throws SAXException   
  {   
    log( "End SAX parse" );   
  }   
  
  /\*\*   
   \* SAXParser calls startElement() when it encounters an opening element tag (eg <item>)   
   \*\*/   
  public void startElement (String name, AttributeList attrs)   
    throws SAXException   
  {   
    log( "<" + name + ">" );   
  }   
  
  /\*\*   
   \* SAXParser calls endElement() when it encounters a closing element tag (eg </item>)   
   \*\*/   
  public void endElement (String name)   
    throws SAXException   
  {   
    log( "</" + name + ">" );   
  }   
  
  /\*\*   
   \* SAXParser calls characters() when it encounters text outside of any tags   
   \*\*/   
  public void characters(char[] p0, int p1, int p2)   
    throws SAXException   
  {   
    String str = new String( p0, p1, p2 ).trim();   
    if( str.length() > 0 )   
      log( str );   
  }   
  
  
}

**Java核心代码例程之：(JAXP) cart.xml**

<?xml version="1.0"?>   
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">   
  
  <xsl:template match="/">   
    <html>   
      <head>   
         <title>Shopping Cart</title>   
      </head>   
      <body>   
<h1>Shopping Cart</h1>   
  
<table border="1">   
<tr>   
<td><b>ID</b></td>   
<td><b>Make</b></td>   
<td><b>Model</b></td>   
<td><b>Price</b></td>   
<td><b>Count</b></td>   
</tr>   
  
<xsl:for-each select="cart/item">   
<tr>   
<td><xsl:value-of select="id"/></td>   
     <td><xsl:value-of select="make"/></td>   
     <td><xsl:value-of select="model"/></td>   
     <td><xsl:value-of select="price"/></td>   
<td><xsl:value-of select="count"/></td>   
</tr>   
</xsl:for-each>   
</table>   
  
      </body>   
    </html>   
  </xsl:template>   
  
</xsl:stylesheet>

**Java核心代码例程之：(EJB) Home Interface**

/\*\*   
 \* Home interface for HelloWorldSessionBean   
 \*   
 \* @see HelloWorldSessionBean   
 \*\*/   
public interface HelloWorldSessionHome extends javax.ejb.EJBHome   
{   
    HelloWorldSession create()    
                      throws javax.ejb.CreateException,    
                             java.rmi.RemoteException;   
}

**Java核心代码例程之：(EJB) Remote Interface**

/\*\*   
 \* Remote interface for HelloWorldSessionBean   
 \*   
 \* @see HelloWorldSessionBean   
 \*\*/   
public interface HelloWorldSession extends javax.ejb.EJBObject   
{   
   public String sayHello()   
                  throws java.rmi.RemoteException;   
}

**Java核心代码例程之：(EJB) Bean**

import java.io.\*;   
import java.util.\*;   
import java.rmi.\*;   
  
import javax.ejb.\*;   
  
  
/\*\*   
 \* Actual Session Bean code.   
 \*   
 \* @see HelloWorldSessionBeanHome   
 \* @see HelloWorldSession   
 \*\*/   
public class HelloWorldSessionBean   
             implements SessionBean   
{   
    SessionContext \_ctx = null;   
    Properties \_props   = null;   
  
  
    public void setSessionContext(SessionContext ctx)   
    {   
        \_ctx   = ctx;   
        \_props = ctx.getEnvironment();   
    }   
  
  
    public String sayHello()   
    {   
        return "Hello, World!";   
    }   
  
  
    /\*\* Required EJB method \*\*/   
    public void ejbCreate () throws CreateException {}   
  
    /\*\* Required EJB method \*\*/   
    public void ejbActivate() {}   
  
    /\*\* Required EJB method \*\*/   
    public void ejbRemove() {}   
  
    /\*\* Required EJB method \*\*/   
    public void ejbPassivate() {}   
}

**Session(Resin Example)**

<%@ page language=javascript %><%    
/\* session.jsp    
\*    
\* Trivial example of session variables, just stores a counter    
\* in the session variable.    
\*    
\* Since the Request object comes from the Servlet API, Resin    
\* scripts can share session variables with Java Servlets.    
\*/    
var count = session.value.counter++;    
  
%>    
  
<html>    
<head><title>Counter</title></head>    
<body bgcolor=#ffffff>    
<%    
  
if (session.isNew()) {    
%><h1>Welcome to a new session.</h1><%    
} else if (session.value.name) {    
%><h1>Welcome back: <%= session.value.name %> <%= count %> </h1><%    
} else {    
%><h1>Welcome back: <%= count %> </h1><%    
}    
  
%>    
<p/><a href="<%= response.encodeUrl("session.jsp") %>">click</a> to enable    
session rewriting.    
</body>    
</html>

**树形结构的实现**

本例采用Access做为实例，所有还有很多要改进的地方。    
请读者自已改进    
  
package product;    
import java.sql.\*;    
public class ConnectDB{    
String strDBDriver="sun.jdbc.odbc.JdbcOdbcDriver";    
String strConnstr="jdbc:odbc:product";    
Connection conn=null;    
ResultSet rs=null;    
Statement stmt=null;    
// Statement stmt=null;    
//构造函数    
public ConnectDB(){    
try{    
Class.forName(strDBDriver); conn=DriverManager.getConnection(strConnstr);    
stmt=conn.createStatement();    
}    
catch(Exception e){    
System.out.println(e);    
}    
}    
  
public ResultSet execute(String sql){    
rs=null;    
try{    
  
rs=stmt.executeQuery(sql);    
}    
catch(SQLException e){    
System.out.println(e.getMessage());    
}    
return rs;    
}    
  
public Connection getConnection(){    
return conn;    
}    
  
public boolean close(){    
try{    
if(this.rs!=null){    
this.rs.close();    
}    
if(stmt!=null){    
this.stmt.close();    
}    
if(conn!=null){    
this.conn.close();    
}    
return true;    
}catch(Exception err){    
return false;    
}    
}    
  
}    
//=======================================    
package product;    
import java.sql.\*;    
public class catalog    
{    
ConnectDB conn=new ConnectDB();    
String str=null;    
String sql=null;    
ResultSet rs=null;    
String AncestorID,LinkStr;    
  
public String getCatalog(){    
  
str="<select name="FatherID" class="p9">  
"+    
"<option value="-1" selected>请选择父类</option>  
";    
String sql=null;    
sql="select \* from catalog order by AncestorID,Linkstr";    
ResultSet rs=conn.execute(sql);    
try{    
while (rs.next()) {    
str=str+"<option value="+rs.getString("Productid")+">";    
int nbspCount=rs.getString("LinkStr").length()-1;    
for(int i=0;i<nbspCount;i++){    
str=str+"&nbsp;";    
}    
if(nbspCount>0)    
str=str+"┠"+rs.getString("title")+"</option>  
";    
else    
str=str+rs.getString("title")+"</option>  
";    
  
}    
str=str+"</select>";    
rs.close();    
}    
catch(SQLException e){    
System.out.println(e.getMessage());    
}    
return str;    
}    
public void setCatalog(String CatalogName,String FatherID){    
int ChildNum=0;    
if(FatherID.compareTo("-1")!=0){    
sql="select AncestorID,ChildNum,linkstr from catalog where productid="+FatherID;    
//out.println(sql);    
try{    
rs=conn.execute(sql);    
if(rs.next()){    
AncestorID=rs.getString("AncestorID");    
ChildNum=rs.getInt("ChildNum");    
LinkStr=rs.getString("linkstr");    
}    
else{    
AncestorID="";    
LinkStr="1";    
ChildNum=0;    
}    
LinkStr=LinkStr+(ChildNum+1);    
//out.println(AncestorID+":"+LinkStr+":"+ChildNum);    
sql="insert into catalog(title,fatherID,AncestorID,LinkStr) values(""+CatalogName+"",""+FatherID+"",""+AncestorID+"",""+LinkStr+"")";    
//out.println(sql);    
conn.execute(sql);    
sql="update catalog set childNum="+(ChildNum+1)+" where productID="+FatherID;    
//out.println("<br>"+sql);    
conn.execute(sql);    
}catch(Exception e){    
System.out.println(e);    
}    
  
}else{    
sql="insert into catalog(title,fatherID,LinkStr) values(""+CatalogName+"",""+FatherID+"","1")";    
//out.println(sql);    
try{    
conn.execute(sql);    
sql="select top 1 productID from catalog order by productID desc ";    
rs=conn.execute(sql);    
if(rs.next()){    
AncestorID=rs.getString("productID");    
}    
else{    
AncestorID="";    
}    
rs.close();    
sql="update catalog set AncestorID=""+AncestorID+"" where productID="+AncestorID;    
//out.println(sql);    
conn.execute(sql);    
}catch(Exception e){    
System.out.println(e);    
  
}    
  
}//end if    
  
if(rs!=null){    
try{    
rs.close();    
}catch(Exception e){    
System.out.println(e);    
}    
}    
  
}    
  
public void deleteCatalog(String FatherID){    
int tempData=0;    
sql="select fatherID,linkstr,AncestorID from catalog where productID="+FatherID;    
rs=conn.execute(sql);    
try{    
if(rs.next()){    
tempData=rs.getInt("fatherID");    
LinkStr=rs.getString("linkstr");    
AncestorID=rs.getString("AncestorID");    
}    
  
try{rs.close();}    
catch(Exception e){    
System.out.println(e);    
}    
sql="delete from catalog where ProductID="+FatherID;    
// out.println(sql);    
try{    
conn.execute(sql);    
}catch(Exception e){    
System.out.println(e);    
}    
sql="delete from catalog where linkstr like ""+LinkStr+"%" and AncestorID="+AncestorID;    
//out.println(sql);    
try{    
conn.execute(sql);    
}catch(Exception e){    
System.out.println(e);    
}    
sql="update catalog set ChildNum=ChildNum-1 where productID="+tempData;    
//out.println(sql);    
try{    
conn.execute(sql);    
}catch(Exception e){    
System.out.println(e);    
}    
}catch(Exception e){    
System.out.println(e);    
}    
} //end public    
  
public void updateCatalog(int id){    
  
}    
  
public String getCatalog(int id){    
  
str="<select name="FatherID" class="p9">  
"+    
"<option value="-1" >请选择父类</option>  
";    
String sql=null;    
sql="select \* from catalog order by AncestorID,Linkstr";    
ResultSet rs=conn.execute(sql);    
try{    
while (rs.next()) {    
int ProductID=rs.getInt("productID");    
if (ProductID==id) {    
str=str+"<option value="+ProductID+" selected>";    
}    
else{    
str=str+"<option value="+ProductID+">";    
}    
int nbspCount=rs.getString("LinkStr").length()-1;    
for(int i=0;i<nbspCount;i++){    
str=str+"&nbsp;";    
}    
if(nbspCount>0)    
str=str+"┠"+rs.getString("title")+"</option>  
";    
else    
str=str+rs.getString("title")+"</option>  
";    
  
}    
str=str+"</select>";    
rs.close();    
}    
catch(SQLException e){    
System.out.println(e.getMessage());    
}    
return str;    
}    
  
public String getCatalogList(){    
  
str="<select name="FatherID" class="p9" size="20" onClick="MM\_jumpMenu("parent.mainFrame",this,0)">  
"+    
"<option value="-1" selected>请选择父类</option>  
";    
String sql=null;    
sql="select \* from catalog order by AncestorID,Linkstr";    
ResultSet rs=conn.execute(sql);    
try{    
while (rs.next()) {    
str=str+"<option value="+rs.getString("Productid")+">";    
int nbspCount=rs.getString("LinkStr").length()-1;    
for(int i=0;i<nbspCount;i++){    
str=str+"&nbsp;";    
}    
if(nbspCount>0)    
str=str+"┠"+rs.getString("title").trim()+"</option>  
";    
else    
str=str+rs.getString("title")+"</option>  
";    
  
}    
str=str+"</select>";    
rs.close();    
}    
catch(SQLException e){    
System.out.println(e.getMessage());    
}    
return str;    
}    
}

**用户登录的验证完整程序**

public  String  validateUser(String  inputUserid,  String  inputPwd)     
throws  SQLException{   
String  returnString  =  null;   
String  dbUserid  =  "userid";  //  Your  Database  user  id   
String  dbPassword  =  "password"  ;  //  Your  Database  password   
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");   
Connection  con  =  DriverManager.getConnection("jdbc:odbc:myDriver",   
dbUserid  ,dbPassword  );   
Statement  stmt  =  con.createStatement();   
String  sql=  "select  USERID  from  USERTABLE  where  USERID  =  ""  +     
inputUserid  +  ""  and  PASSWORD  =  ""  +  inputPwd  +""  ;"  ;     
ResultSet  rs  =  stmt.executeQuery(sql);   
if  (rs.next())   
{   
returnString  =  rs.getString("USERID");   
}   
stmt.close();   
con.close();   
return  returnString  ;   
}

**一些非常有用的JAVA常用方法(1)**

<%!   
stc\_RSYB\_Name stcRSYName1[];   
stc\_SYB\_INF stcSYBINF1[];   
%>   
  
<%/\*   
   String p = (String) session.getValue("access");   
   if (p==null || !p.equals("1")) {   
     response.sendRedirect("error.htm");   
     return;   
   }\*/   
%>   
<%!   
file://得到机器名字或IP   
String get\_Myhostname(javax.servlet.http.HttpServletRequest request){   
  String myhostname = null;   
    try{   
      myhostname = request.getRemoteHost();   
      myhostname = myhostname.toUpperCase();   
    }catch(Exception e){}   
  return myhostname;   
}   
file://处理空字符串   
  String dealNull(String str) {   
    String returnstr = null;   
    if (str == null) returnstr = "";   
    else returnstr = str;   
    return returnstr;   
  }   
file://处理空对象     
  Object dealNull(Object obj){   
    Object returnstr = null;   
    if (obj == null) returnstr = (Object)("");   
    else returnstr = obj;   
    return returnstr;   
  }   
  
  int dealEmpty(String s) {   
    s = dealNull(s);   
    if (s.equals("")) return 0;   
    return Integer.parseInt(s);   
  }   
  
String replace(String str,String substr,String restr){   
  String[] tmp = split(str,substr);   
  String returnstr = null;   
  if(tmp.length!=0) {   
    returnstr = tmp[0];   
    for(int i = 0 ; i < tmp.length - 1 ; i++)   
      returnstr =dealNull(returnstr) + restr +tmp[i+1];   
  }    
  return dealNull(returnstr);   
}      
      
String htmlEncode(String txt){   
  txt = replace(txt,"&","&");   
  txt = replace(txt,"&","&");   
  txt = replace(txt,""",""");   
  txt = replace(txt,""",""");   
  txt = replace(txt,"<","<");   
  txt = replace(txt,"<","<");   
  txt = replace(txt,">",">");   
  txt = replace(txt,">",">");   
  txt = replace(txt,"&nbsp;","&nbsp;");   
  txt = replace(txt," ","&nbsp;");   
  return txt;   
}   
  
String unHtmlEncode(String txt){   
  txt = replace(txt,"&","&");   
  txt = replace(txt,""",""");   
  txt = replace(txt,"<","<");   
  txt = replace(txt,">",">");   
  txt = replace(txt,"&nbsp;"," ");   
  return txt;   
}   
  
file://ALERT Function   
  void sAlert(String title,String msg,String focus, JspWriter out){   
   try {   
      out.println("<script language="javascript">");   
      out.println("alert("" + title + "" + "   
" + "" + replace(replace(msg,"""","""),""",""") +"");");   
      out.println(focus);   
      out.println("</script>");   
   } catch(Exception e) { System.out.println(e);}   
  }   
file://CONFIRM Function   
  void sConfirm(String title,String msg ,String location,JspWriter out){   
    try{   
        out.println("<script language="javascript">");   
  out.println("if(confirm("" + title + "" + "   
" + "" + replace(msg,""",""") + "")){" );   
        out.println("var d = (new Date()).toString();");   
  out.println(location + "+ d;");   
        out.println("}");   
        out.println("parent.document.form1.text1.select();");   
        out.println("</script>");   
   } catch (Exception e) { }   
  }   
     
file://得到字符串长度     
  int getStringLength(String s) {   
   int p = 0;   
    try {   
      int iLength = 0;   
      int irealLength = 0;   
      int j = 0;   
      p = s.getBytes("Shift\_JIS").length;   
    } catch(java.io.UnsupportedEncodingException e) {}   
    return p;   
  }   
  
void formatString(String s,int b,JspWriter out){   
    b--;   
    String t = "";   
    try {   
        int leng = getStringLength(s);   
        int j = 0;   
if (leng > b) {    
   for (int i = 1; i <= b ; i++) {   
            t = s.substring(0,i);   
     if (t.getBytes("Shift\_JIS").length <= b ) {   
       j++;   
     } else break;    
   }   
          file://j = j+1;   
          file://System.out.println("b=" + b + "----" + "j=" + j);   
          for(int i = 0;i < j; i++){    
            if(s.substring(i,i+1).equals(" "))   
              out.print("&nbsp;");    
            if(!s.substring(i,i+1).equals(" "))   
       out.print(htmlEncode(s.substring(i,i+1)));   
           }       
          if (s.substring(0,j).getBytes("Shift\_JIS").length  == b){   
            out.print("|");   
          }else{   
            out.print("&nbsp;|");   
          }   
        } else {   
   for (int  i = 0; i < b - leng ; i++)   
            t = t + "&nbsp;";   
          for (int i = 0; i < s.length(); i++){   
            if(s.substring(i,i+1).equals(" "))   
              out.print("&nbsp;");   
            if(!s.substring(i,i+1).equals(" "))   
              out.print(htmlEncode(s.substring(i,i+1)));   
          }   
   out.print(htmlEncode(t + "|"));   
       }   
    } catch(Exception e){System.out.println("----" + e);}   
  }   
  
file://判断是否为数字   
boolean isNumeric(String number)   
{   
try   
{   
  Integer.parseInt(number);   
  return true;   
}   
catch(NumberFormatException sqo)   
{   
  return false;   
}   
}   
file://判断是否为日期   
String mk\_date(String c\_date,int c\_mode)   
{   
String wk\_buf= "";   
String ToDay = "";   
try   
{   
  ToDay = Integer.toString(java.util.Calendar.getInstance().get(java.util.Calendar.YEAR));   
  switch(c\_mode)   
  {   
   case 0:   
   {   
    wk\_buf = c\_date.substring(2, 4) + "/" + c\_date.substring(4, 6);   
             if (c\_date.length() == 8)   
    {   
                 wk\_buf = wk\_buf + "/" + c\_date.substring(6, 8);   
             }   
    break;   
   }   
   case 1:   
   {   
    for (int i=0; i<c\_date.length();i++)   
    {   
              if (c\_date.substring(i, i+1) == " ")   
     {    
                     wk\_buf = wk\_buf + "0";   
                 }   
     else   
     {   
                     wk\_buf = wk\_buf + c\_date.substring(i, i+1);   
                 }   
    }   
             c\_date = wk\_buf;   
    wk\_buf = "";   
             if (c\_date.substring(0, 1) != "0" )   
    {   
                 wk\_buf = "19";   
             }   
    else   
    {   
                 wk\_buf = "20";   
             }   
       
             wk\_buf = wk\_buf + c\_date.substring(0, 2) + c\_date.substring(3, 5);   
       
             if (c\_date.length() == 8)   
    {   
                  wk\_buf = wk\_buf + c\_date.substring(6, 8);   
             }   
    break;   
   }   
   case 2:   
   {   
    wk\_buf = c\_date.substring(0, 4) + c\_date.substring(5, 7) + c\_date.substring(8, 10);   
    break;   
   }   
   case 3:   
   {   
    for(int i=0; i<c\_date.length();i++ )   
    {   
                 if (c\_date.substring(i, i+1) == " ")   
     {   
                     wk\_buf = wk\_buf + "0";   
                 }   
     else   
     {   
                     wk\_buf = wk\_buf + c\_date.substring(i, i+1);   
                 }   
             }   
             c\_date = wk\_buf;   
             wk\_buf = "";   
             if (Integer.parseInt(c\_date.substring(0,2)) > Integer.parseInt(ToDay.substring(2, 4)))   
    {   
                 wk\_buf = Integer.toString(Integer.parseInt(ToDay.substring(0,2)) - 1);   
             }   
    else   
    {   
                 wk\_buf = ToDay.substring(0,2);   
             }   
             wk\_buf = wk\_buf + c\_date.substring(0,2) + c\_date.substring(3,5);   
             if (c\_date.length() == 8)   
    {   
                 wk\_buf = wk\_buf + c\_date.substring(6,8);   
              }   
    break;   
   }   
   case 4:   
   {   
    wk\_buf = c\_date.substring(0,4) + "/" + c\_date.substring(4,6) + "/" + c\_date.substring(6, 8);   
    break;   
   }   
   case 5:   
   {   
    wk\_buf = c\_date.substring(0,4) + c\_date.substring(5, 7);   
    break;   
   }   
  }   
}   
catch(Exception e)   
{   
  System.out.println("wrong : " + e);   
}   
return wk\_buf;   
}

**阴阳历算法（转）**

西历农历转换程式 黄晓鸣 1995,7,25   
  
prototype: int calconv( struct convdate \* );   
  
struct convdate   
{   
    int source; ==0 则输入日期为西历, !=0 则输入为农历   
    int solaryear; 输出或输入之西历年份   
    int solarmonth; 西历月   
    int solardate; 西历日   
    int lunaryear; 输出或输入之农历年份   
    int lunarmonth; 农历月   
    int lunardate; 农历日   
    int weekday; 该日为星期几 ( 0==星期日, 1==星期一, ... )   
    int kan; 该日天干 ( 0==甲, 1==乙, ..., 9==癸 )   
    int chih; 该日地支 ( 0==子, 1==丑, ..., 11==亥 )   
};   
  
呼叫时须设定 souce 的值, 若为 0 则为西历转农历, 否则为农历转西历. 然後视   
输入为西历或农历来设定西历或农历的年月日. 转换後的年月日会填入结构中( 农   
历或西历 ), 以及该日为星期几, 天干地支.   
若函式的返回值为 0 表示没有错误, 1 为输入之年份错误, 2 为输入之月份错误,   
3 为输入之日期错误.   
输入之西历年须在 1937 - 2031 间   
输入之农历年须在 1936 - 2030 间   
若须扩充, 则增加 lunarcal[]   
  
\*/   
  
#define firstyear 1936 /\* the first year in lunarcal[] \*/   
  
struct convdate   
{   
    int source;   
    int solaryear;   
    int solarmonth;   
    int solardate;   
    int lunaryear;   
    int lunarmonth;   
    int lunardate;   
    int weekday;   
    int kan;   
    int chih;   
};   
  
struct taglunarcal   
{   
    int basedays; /\* 到西历 1 月 1 日到农历正月初一的累积日数 \*/   
    int intercalation; /\* 闰月月份. 0==此年没有闰月 \*/   
    int baseweekday; /\* 此年西历 1 月 1 日为星期几再减 1 \*/   
    int basekanchih; /\* 此年西历 1 月 1 日之干支序号减 1 \*/   
    int monthdays[13]; /\* 此农历年每月之大小, 0==小月(29日), 1==大月(30日)\*/   
};   
  
struct taglunarcal lunarcal[] = {   
{ 23, 3, 2, 17, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0 }, /\* 1936 \*/   
{ 41, 0, 4, 23, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1 },   
{ 30, 7, 5, 28, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1 },   
{ 49, 0, 6, 33, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1 },   
{ 38, 0, 0, 38, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1 }, /\* 1940 \*/   
{ 26, 6, 2, 44, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0 },   
{ 45, 0, 3, 49, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0 },   
{ 35, 0, 4, 54, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1 },   
{ 24, 4, 5, 59, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1 }, /\* 1944 \*/   
{ 43, 0, 0, 5, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1 },   
{ 32, 0, 1, 10, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1 },   
{ 21, 2, 2, 15, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1 },   
{ 40, 0, 3, 20, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1 }, /\* 1948 \*/   
{ 28, 7, 5, 26, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1 },   
{ 47, 0, 6, 31, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1 },   
{ 36, 0, 0, 36, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0 },   
{ 26, 5, 1, 41, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1 }, /\* 1952 \*/   
{ 44, 0, 3, 47, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1 },   
{ 33, 0, 4, 52, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0 },   
{ 23, 3, 5, 57, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1 },   
{ 42, 0, 6, 2, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1 }, /\* 1956 \*/   
{ 30, 8, 1, 8, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0 },   
{ 48, 0, 2, 13, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0 },   
{ 38, 0, 3, 18, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1 },   
{ 27, 6, 4, 23, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0 }, /\* 1960 \*/   
{ 45, 0, 6, 29, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0 },   
{ 35, 0, 0, 34, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1 },   
{ 24, 4, 1, 39, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0 },   
{ 43, 0, 2, 44, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0 }, /\* 1964 \*/   
{ 32, 0, 4, 50, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1 },   
{ 20, 3, 5, 55, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0 },   
{ 39, 0, 6, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0 },   
{ 29, 7, 0, 5, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1 }, /\* 1968 \*/   
{ 47, 0, 2, 11, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1 },   
{ 36, 0, 3, 16, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0 },   
{ 26, 5, 4, 21, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1 },   
{ 45, 0, 5, 26, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1 }, /\* 1972 \*/   
{ 33, 0, 0, 32, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1 },   
{ 22, 4, 1, 37, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1 },   
{ 41, 0, 2, 42, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1 },   
{ 30, 8, 3, 47, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1 }, /\* 1976 \*/   
{ 48, 0, 5, 53, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1 },   
{ 37, 0, 6, 58, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1 },   
{ 27, 6, 0, 3, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0 },   
{ 46, 0, 1, 8, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0 }, /\* 1980 \*/   
{ 35, 0, 3, 14, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1 },   
{ 24, 4, 4, 19, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1 },   
{ 43, 0, 5, 24, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1 },   
{ 32, 10, 6, 29, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1 }, /\* 1984 \*/   
{ 50, 0, 1, 35, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0 },   
{ 39, 0, 2, 40, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1 },   
{ 28, 6, 3, 45, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0 },   
{ 47, 0, 4, 50, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1 }, /\* 1988 \*/   
{ 36, 0, 6, 56, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0 },   
{ 26, 5, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1 },   
{ 45, 0, 1, 6, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0 },   
{ 34, 0, 2, 11, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0 }, /\* 1992 \*/   
{ 22, 3, 4, 17, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0 },   
{ 40, 0, 5, 22, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0 },   
{ 30, 8, 6, 27, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1 },   
{ 49, 0, 0, 32, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1 }, /\* 1996 \*/   
{ 37, 0, 2, 38, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1 },   
{ 27, 5, 3, 43, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1 },   
{ 46, 0, 4, 48, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1 },   
{ 35, 0, 5, 53, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1 }, /\* 2000 \*/   
{ 23, 4, 0, 59, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1 },   
{ 42, 0, 1, 4, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1 },   
{ 31, 0, 2, 9, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0 },   
{ 21, 2, 3, 14, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1 }, /\* 2004 \*/   
{ 39, 0, 5, 20, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1 },   
{ 28, 7, 6, 25, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1 },   
{ 48, 0, 0, 30, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1 },   
{ 37, 0, 1, 35, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1 }, /\* 2008 \*/   
{ 25, 5, 3, 41, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1 },   
{ 44, 0, 4, 46, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1 },   
{ 33, 0, 5, 51, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1 },   
{ 22, 4, 6, 56, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0 }, /\* 2012 \*/   
{ 40, 0, 1, 2, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0 },   
{ 30, 9, 2, 7, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1 },   
{ 49, 0, 3, 12, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1 },   
{ 38, 0, 4, 17, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0 }, /\* 2016 \*/   
{ 27, 6, 6, 23, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1 },   
{ 46, 0, 0, 28, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0 },   
{ 35, 0, 1, 33, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0 },   
{ 24, 4, 2, 38, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1 }, /\* 2020 \*/   
{ 42, 0, 4, 44, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1 },   
{ 31, 0, 5, 49, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0 },   
{ 21, 2, 6, 54, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1 },   
{ 40, 0, 0, 59, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1 }, /\* 2024 \*/   
{ 28, 6, 2, 5, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0 },   
{ 47, 0, 3, 10, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1 },   
{ 36, 0, 4, 15, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1 },   
{ 25, 5, 5, 20, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0 }, /\* 2028 \*/   
{ 43, 0, 0, 26, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1 },   
{ 32, 0, 1, 31, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0 },   
{ 22, 3, 2, 36, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0 } };   
  
#define lastyear (firstyear+sizeof(lunarcal)/sizeof(struct taglunarcal)-1)   
  
/\* 西历年每月之日数 \*/   
int solarcal[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };   
  
/\* 西历年每月之累积日数, 平年与闰年 \*/   
int solardays[2][14] = {   
{ 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365, 396 },   
{ 0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335, 366, 397 } };   
  
/\* 求此西历年是否为闰年, 返回 0 为平年, 1 为闰年 \*/   
int getleap( int year )   
{   
    if ( year % 400 == 0 )   
        return 1;   
    else if ( year % 100 == 0 )   
        return 0;   
    else if ( year % 4 == 0 )   
        return 1;   
    else   
        return 0;   
}   
  
/\* 西历农历转换 \*/   
int calconv( struct convdate \*cd )   
{   
    int leap, d, sm, y, im, l1, l2, acc, i, lm, kc;   
    if ( cd->source == 0 ) /\* solar \*/   
    {   
        if ( cd->solaryear <= firstyear || cd->solaryear > lastyear )   
            return 1;   
        sm = cd->solarmonth - 1;   
        if ( sm < 0 || sm > 11 )   
            return 2;   
        leap = getleap( cd->solaryear );   
        if ( sm == 1 )   
            d = leap + 28;   
        else   
            d = solarcal[sm];   
        if ( cd->solardate < 1 || cd->solardate > d )   
            return 3;   
        y = cd->solaryear - firstyear;   
        acc = solardays[leap][sm] + cd->solardate;   
        cd->weekday = ( acc + lunarcal[y].baseweekday ) % 7;   
        kc = acc + lunarcal[y].basekanchih;   
        cd->kan = kc % 10;   
        cd->chih = kc % 12;   
        if ( acc <= lunarcal[y].basedays )   
        {   
            y--;   
            cd->lunaryear = cd->solaryear - 1;   
            leap = getleap( cd->lunaryear );   
            sm += 12;   
            acc = solardays[leap][sm] + cd->solardate;   
        }   
    else   
        cd->lunaryear = cd->solaryear;   
    l1 = lunarcal[y].basedays;   
    for ( i=0; i<13; i++ )   
    {   
        l2 = l1 + lunarcal[y].monthdays[i] + 29;   
        if ( acc <= l2 )   
            break;   
        l1 = l2;   
    }   
    cd->lunarmonth = i + 1;   
    cd->lunardate = acc - l1;   
    im = lunarcal[y].intercalation;   
    if ( im != 0 && cd->lunarmonth > im )   
    {   
        cd->lunarmonth--;   
        if ( cd->lunarmonth == im )   
            cd->lunarmonth = -im;   
    }   
    if ( cd->lunarmonth > 12 )   
        cd->lunarmonth -= 12;   
    }   
    else /\* lunar \*/   
    {   
        if ( cd->lunaryear < firstyear || cd->lunaryear >= lastyear )   
            return 1;   
        y = cd->lunaryear - firstyear;   
        im = lunarcal[y].intercalation;   
        lm = cd->lunarmonth;   
        if ( lm < 0 )   
        {   
            if ( lm != -im )   
                return 2;   
        }   
        else if ( lm < 1 || lm > 12 )   
            return 2;   
        if ( im != 0 )   
        {   
            if ( lm > im )   
                lm++;   
            else if ( lm == -im )   
                lm = im + 1;   
        }   
        lm--;   
        if ( cd->lunardate > lunarcal[y].monthdays[lm] + 29 )   
            return 3;   
        acc = lunarcal[y].basedays;   
        for ( i=0; i acc += lunarcal[y].monthdays[i] + 29;   
            acc += cd->lunardate;   
            leap = getleap( cd->lunaryear );   
        for ( i=13; i>=0; i-- )   
            if ( acc > solardays[leap][i] )   
                break;   
            cd->solardate = acc - solardays[leap][i];   
            if ( i <= 11 )   
            {   
                cd->solaryear = cd->lunaryear;   
                cd->solarmonth = i + 1;   
            }   
            else   
            {   
                cd->solaryear = cd->lunaryear + 1;   
                cd->solarmonth = i - 11;   
            }   
            leap = getleap( cd->solaryear );   
            y = cd->solaryear - firstyear;   
            acc = solardays[leap][cd->solarmonth-1] + cd->solardate;   
            cd->weekday = ( acc + lunarcal[y].baseweekday ) % 7;   
            kc = acc + lunarcal[y].basekanchih;   
            cd->kan = kc % 10;   
            cd->chih = kc % 12;   
        )   
return 0;   
}

**给你个比较全的关于时间的操作的东西**

import java.text.DateFormat;   
import java.util.\*;   
  
public class JspCalendar {   
    Calendar  calendar = null;   
  
    public JspCalendar() {   
    calendar = Calendar.getInstance();   
    Date trialTime = new Date();   
    calendar.setTime(trialTime);   
    }   
  
    public int getYear() {   
    return calendar.get(Calendar.YEAR);   
    }   
       
    public String getMonth() {   
    int m = getMonthInt();   
    String[] months = new String [] { "January", "February", "March",   
                    "April", "May", "June",   
                    "July", "August", "September",   
                    "October", "November", "December" };   
    if (m > 12)   
        return "Unknown to Man";   
       
    return months[m - 1];   
  
    }   
  
    public String getDay() {   
    int x = getDayOfWeek();   
    String[] days = new String[] {"Sunday", "Monday", "Tuesday", "Wednesday",    
                      "Thursday", "Friday", "Saturday"};   
  
    if (x > 7)   
        return "Unknown to Man";   
  
    return days[x - 1];   
  
    }   
       
    public int getMonthInt() {   
    return 1 + calendar.get(Calendar.MONTH);   
    }   
  
    public String getDate() {   
    return getMonthInt() + "/" + getDayOfMonth() + "/" +  getYear();   
  
    }   
  
    public String getTime() {   
    return getHour() + ":" + getMinute() + ":" + getSecond();   
    }   
  
    public int getDayOfMonth() {   
    return calendar.get(Calendar.DAY\_OF\_MONTH);   
    }   
  
    public int getDayOfYear() {   
    return calendar.get(Calendar.DAY\_OF\_YEAR);   
    }   
  
    public int getWeekOfYear() {   
    return calendar.get(Calendar.WEEK\_OF\_YEAR);   
    }   
  
    public int getWeekOfMonth() {   
    return calendar.get(Calendar.WEEK\_OF\_MONTH);   
    }   
  
    public int getDayOfWeek() {   
    return calendar.get(Calendar.DAY\_OF\_WEEK);   
    }   
        
    public int getHour() {   
    return calendar.get(Calendar.HOUR\_OF\_DAY);   
    }   
       
    public int getMinute() {   
    return calendar.get(Calendar.MINUTE);   
    }   
  
  
    public int getSecond() {   
    return calendar.get(Calendar.SECOND);   
    }   
  
    public static void main(String args[]) {   
    JspCalendar db = new JspCalendar();   
    p("date: " + db.getDayOfMonth());   
    p("year: " + db.getYear());   
    p("month: " + db.getMonth());   
    p("time: " + db.getTime());   
    p("date: " + db.getDate());   
    p("Day: " + db.getDay());   
    p("DayOfYear: " + db.getDayOfYear());   
    p("WeekOfYear: " + db.getWeekOfYear());   
    p("era: " + db.getEra());   
    p("ampm: " + db.getAMPM());   
    p("DST: " + db.getDSTOffset());   
    p("ZONE Offset: " + db.getZoneOffset());   
    p("TIMEZONE: " + db.getUSTimeZone());   
    }   
  
    private static void p(String x) {   
    System.out.println(x);   
    }   
  
  
    public int getEra() {   
    return calendar.get(Calendar.ERA);   
    }   
  
    public String getUSTimeZone() {   
    String[] zones = new String[] {"Hawaii", "Alaskan", "Pacific",   
                       "Mountain", "Central", "Eastern"};   
       
    return zones[10 + getZoneOffset()];   
    }   
  
    public int getZoneOffset() {   
    return calendar.get(Calendar.ZONE\_OFFSET)/(60\*60\*1000);   
    }   
  
  
    public int getDSTOffset() {   
    return calendar.get(Calendar.DST\_OFFSET)/(60\*60\*1000);   
    }   
  
        public int getAMPM() {   
    return calendar.get(Calendar.AM\_PM);   
    }   
}