**Append text to file in Java**

BufferedWriter out = null;

try {

    out = new BufferedWriter(new FileWriter(”filename”, true));

    out.write(”aString”);

} catch (IOException e) {

    // error processing code

} finally {

    if (out != null) {

        out.close();

    }

}

## Get name of current method in Java

String methodName = Thread.currentThread().getStackTrace()[1].getMethodName();

## Convert String to Date in Java

SimpleDateFormat format = new SimpleDateFormat( "dd.MM.yyyy" );

Date date = format.parse( myString );

## Connecting to Oracle using Java JDBC

public class OracleJdbcTest

{

    String driverClass = "oracle.jdbc.driver.OracleDriver";

    Connection con;

    public void init(FileInputStream fs) throws ClassNotFoundException, SQLException, FileNotFoundException, IOException

    {

        Properties props = new Properties();

        props.load(fs);

        String url = props.getProperty("db.url");

        String userName = props.getProperty("db.user");

        String password = props.getProperty("db.password");

        Class.forName(driverClass);

        con=DriverManager.getConnection(url, userName, password);

    }

    public void fetch() throws SQLException, IOException

    {

        PreparedStatement ps = con.prepareStatement("select SYSDATE from dual");

        ResultSet rs = ps.executeQuery();

        while (rs.next())

        {

            // do the thing you do

        }

        rs.close();

        ps.close();

    }

    public static void main(String[] args)

    {

        OracleJdbcTest test = new OracleJdbcTest();

        test.init();

        test.fetch();

    }

}

## Convert Java util.Date to sql.Date

java.util.Date utilDate = new java.util.Date();

java.sql.Date sqlDate = new java.sql.Date(utilDate.getTime());

## Java Fast File Copy using NIO

public static void fileCopy( File in, File out )

            throws IOException

    {

        FileChannel inChannel = new FileInputStream( in ).getChannel();

        FileChannel outChannel = new FileOutputStream( out ).getChannel();

        try

        {

//          inChannel.transferTo(0, inChannel.size(), outChannel);      // original -- apparently has trouble copying large files on Windows

            // magic number for Windows, 64Mb - 32Kb)

            int maxCount = (64 \* 1024 \* 1024) - (32 \* 1024);

            long size = inChannel.size();

            long position = 0;

            while ( position < size )

            {

               position += inChannel.transferTo( position, maxCount, outChannel );

            }

        }

        finally

        {

            if ( inChannel != null )

            {

               inChannel.close();

            }

            if ( outChannel != null )

            {

                outChannel.close();

            }

        }

    }

## Create Thumbnail of an image in Java

private void createThumbnail(String filename, int thumbWidth, int thumbHeight, int quality, String outFilename)

        throws InterruptedException, FileNotFoundException, IOException

    {

        // load image from filename

        Image image = Toolkit.getDefaultToolkit().getImage(filename);

        MediaTracker mediaTracker = new MediaTracker(new Container());

        mediaTracker.addImage(image, 0);

        mediaTracker.waitForID(0);

        // use this to test for errors at this point: System.out.println(mediaTracker.isErrorAny());

        // determine thumbnail size from WIDTH and HEIGHT

        double thumbRatio = (double)thumbWidth / (double)thumbHeight;

        int imageWidth = image.getWidth(null);

        int imageHeight = image.getHeight(null);

        double imageRatio = (double)imageWidth / (double)imageHeight;

        if (thumbRatio < imageRatio) {

            thumbHeight = (int)(thumbWidth / imageRatio);

        } else {

            thumbWidth = (int)(thumbHeight \* imageRatio);

        }

        // draw original image to thumbnail image object and

        // scale it to the new size on-the-fly

        BufferedImage thumbImage = new BufferedImage(thumbWidth, thumbHeight, BufferedImage.TYPE\_INT\_RGB);

        Graphics2D graphics2D = thumbImage.createGraphics();

        graphics2D.setRenderingHint(RenderingHints.KEY\_INTERPOLATION, RenderingHints.VALUE\_INTERPOLATION\_BILINEAR);

        graphics2D.drawImage(image, 0, 0, thumbWidth, thumbHeight, null);

        // save thumbnail image to outFilename

        BufferedOutputStream out = new BufferedOutputStream(new FileOutputStream(outFilename));

        JPEGImageEncoder encoder = JPEGCodec.createJPEGEncoder(out);

        JPEGEncodeParam param = encoder.getDefaultJPEGEncodeParam(thumbImage);

        quality = Math.max(0, Math.min(quality, 100));

        param.setQuality((float)quality / 100.0f, false);

        encoder.setJPEGEncodeParam(param);

        encoder.encode(thumbImage);

        out.close();

    }

## Files-Directory listing in Java

File dir = new File("directoryName");

  String[] children = dir.list();

  if (children == null) {

      // Either dir does not exist or is not a directory

  } else {

      for (int i=0; i < children.length; i++) {

          // Get filename of file or directory

          String filename = children[i];

      }

  }

  // It is also possible to filter the list of returned files.

  // This example does not return any files that start with `.'.

  FilenameFilter filter = new FilenameFilter() {

      public boolean accept(File dir, String name) {

          return !name.startsWith(".");

      }

  };

  children = dir.list(filter);

  // The list of files can also be retrieved as File objects

  File[] files = dir.listFiles();

  // This filter only returns directories

  FileFilter fileFilter = new FileFilter() {

      public boolean accept(File file) {

          return file.isDirectory();

      }

  };

  files = dir.listFiles(fileFilter);

## Creating ZIP and JAR Files in Java

import java.util.zip.\*;

import java.io.\*;

public class ZipIt {

    public static void main(String args[]) throws IOException {

        if (args.length < 2) {

            System.err.println("usage: java ZipIt Zip.zip file1 file2 file3");

            System.exit(-1);

        }

        File zipFile = new File(args[0]);

        if (zipFile.exists()) {

            System.err.println("Zip file already exists, please try another");

            System.exit(-2);

        }

        FileOutputStream fos = new FileOutputStream(zipFile);

        ZipOutputStream zos = new ZipOutputStream(fos);

        int bytesRead;

        byte[] buffer = new byte[1024];

        CRC32 crc = new CRC32();

        for (int i=1, n=args.length; i < n; i++) {

            String name = args[i];

            File file = new File(name);

            if (!file.exists()) {

                System.err.println("Skipping: " + name);

                continue;

            }

            BufferedInputStream bis = new BufferedInputStream(

                new FileInputStream(file));

            crc.reset();

            while ((bytesRead = bis.read(buffer)) != -1) {

                crc.update(buffer, 0, bytesRead);

            }

            bis.close();

            // Reset to beginning of input stream

            bis = new BufferedInputStream(

                new FileInputStream(file));

            ZipEntry entry = new ZipEntry(name);

            entry.setMethod(ZipEntry.STORED);

            entry.setCompressedSize(file.length());

            entry.setSize(file.length());

            entry.setCrc(crc.getValue());

            zos.putNextEntry(entry);

            while ((bytesRead = bis.read(buffer)) != -1) {

                zos.write(buffer, 0, bytesRead);

            }

            bis.close();

        }

        zos.close();

    }

}

## Parsing / Reading XML file in Java

<?xml version="1.0"?>

<students>

    <student>

        <name>John</name>

        <grade>B</grade>

        <age>12</age>

    </student>

    <student>

        <name>Mary</name>

        <grade>A</grade>

        <age>11</age>

    </student>

    <student>

        <name>Simon</name>

        <grade>A</grade>

        <age>18</age>

    </student>

</students>

import java.io.File;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

public class XMLParser {

    public void getAllUserNames(String fileName) {

        try {

            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

            DocumentBuilder db = dbf.newDocumentBuilder();

            File file = new File(fileName);

            if (file.exists()) {

                Document doc = db.parse(file);

                Element docEle = doc.getDocumentElement();

                // Print root element of the document

                System.out.println("Root element of the document: "

                        + docEle.getNodeName());

                NodeList studentList = docEle.getElementsByTagName("student");

                // Print total student elements in document

                System.out

                        .println("Total students: " + studentList.getLength());

                if (studentList != null && studentList.getLength() > 0) {

                    for (int i = 0; i < studentList.getLength(); i++) {

                        Node node = studentList.item(i);

                        if (node.getNodeType() == Node.ELEMENT\_NODE) {

                            System.out

                                    .println("=====================");

                            Element e = (Element) node;

                            NodeList nodeList = e.getElementsByTagName("name");

                            System.out.println("Name: "

                                    + nodeList.item(0).getChildNodes().item(0)

                                            .getNodeValue());

                            nodeList = e.getElementsByTagName("grade");

                            System.out.println("Grade: "

                                    + nodeList.item(0).getChildNodes().item(0)

                                            .getNodeValue());

                            nodeList = e.getElementsByTagName("age");

                            System.out.println("Age: "

                                    + nodeList.item(0).getChildNodes().item(0)

                                            .getNodeValue());

                        }

                    }

                } else {

                    System.exit(1);

                }

            }

        } catch (Exception e) {

            System.out.println(e);

        }

    }

    public static void main(String[] args) {

        XMLParser parser = new XMLParser();

        parser.getAllUserNames("c:\\test.xml");

    }

}