Education and Research Department

Demo Programs for Java Programming Part-2

September 2009

CONFIDENTIAL

Document No.	Authorized By	Ver. Revision	Signature / Date
ER/CORP/CRS/ERJEEML101/018	Srikantan Moorthy	Ver.*1.0*	September 2009

COPYRIGHT NOTICE

All ideas and information contained in this document are the intellectual property of Education and Research Department, Infosys Technologies Limited. This document is not for general distribution and is meant for use only for the person they are specifically issued to. This document shall not be loaned to anyone, within or outside Infosys, including its customers. Copying or unauthorized distribution of this document, in any form or means including electronic, mechanical, photocopying or otherwise is illegal.

Education and Research Department
Infosys Technologies Limited
Electronic City
Hosur Road
Bangalore - 561 229, India.

Tel: 91 80 852 0261-270
Fax: 91 80 852 0362
www.infosys.com
mailto:E&R@infosys.com

CONFIDENTIAL

Document Revision History

Ver. Revision	Date	Author(s)	Reviewer(s)	Description
1.0	Sep-2009	Avinash Pratap Singh and Sanjeev S.	Soni Ramchandani	Baseline

CONFIDENTIAL

Contents

CONFIDENTIAL

Demo Programs for *Java Programming Part 2*

1-Creating thread using Thread class

```
**
 * This demo explains Thread  creation and usage in
Java
 */
import java.io.IOException;

class MyThread extends Thread {
  public MyThread(String firstName, String secondNa
me, long delay) {
    this.firstName = firstName;
    this.secondName = secondName;
    aWhile = delay;
    setDaemon(true);
  }
  public void run() {
    try {
      while (true) {
        System.out.print(firstName);
        sleep(aWhile);
        System.out.print(secondName + "\n");
      }
    } catch (InterruptedException e) {
      System.out.println(firstName + secondName + e
);
    }
  }
  private String firstName;
  private String secondName;
  private long aWhile;
}
public class MainClass {
  public static void main(String[] args) {
    Thread first = new MyThread
("A ", "a  ", 200L);
    Thread second = new MyThread
("B ", "b ", 300L);
```

CONFIDENTIAL

```
    Thread third = new MyThread ("C ", "c ", 500L);
    System.out.println("Press Enter when you have h
ad enough...\n");
    first.start();
    second.start();
    third.start();
    try {
      System.in.read();
      System.out.println("Enter pressed...\n");
    } catch (IOException e) {
      System.out.println(e);
    }
    return;
  }
}
```

2-Creating thread using Runnable interface

```
/**
 * This demo explains how to use Runnable interface
to create Threads in Java
 */
import java.io.IOException;

class MyThread implements Runnable {
  public MyThread
(String firstName, String secondName, long delay) {
    this.firstName = firstName;
    this.secondName = secondName;
    aWhile = delay;
  }
  public void run() {
    try {
      while (true) {
        System.out.print(firstName);
        Thread.sleep(aWhile);
        System.out.print(secondName + "\n");
```

CONFIDENTIAL

```
      }
    } catch (InterruptedException e) {
      System.out.println(firstName + secondName + e
);
    }
  }
  private String firstName;
  private String secondName;
  private long aWhile;
}
public class MainClass {
  public static void main(String[] args) {
 Thread first = new Thread(new MyThread("A ", "a ",
 200L));
 Thread second = new Thread(new MyThread("B ", "b "
, 300L));
    Thread third = new Thread(new MyThread
("C ", "c ", 500L));
    System.out.println("Press Enter when you have h
ad enough...\n");
    first.start();
    second.start();
    third.start();
    try {
      System.in.read();
      System.out.println("Enter pressed...\n");
    } catch (IOException e) {
      System.out.println(e);
    }
    System.out.println("Ending main()");
    return;
  }
}
```

CONFIDENTIAL

3-Example to demonstrate ExecutorService class

```
/**
```

```
 * This demo explains how to use ExecutorService
in Java.
 */


import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

public class MainClass {
  public static void main(String[] args) {
    MyTask  task1 = new MyTask  ("thread1");
    MyTask  task2 = new MyTask  ("thread2");
    MyTask  task3 = new MyTask  ("thread3");

    System.out.println("Starting threads");

    ExecutorService threadExecutor = Executors.newC
achedThreadPool();

    // start threads and place in runnable state
    threadExecutor.execute(task1); // start task1
    threadExecutor.execute(task2); // start task2
    threadExecutor.execute(task3); // start task3

    threadExecutor.shutdown(); // shutdown worker t
hreads

    System.out.println("Threads started, main ends\
n");
  }
}

class MyTask implements Runnable {
  private int sleepTime;

  private String threadName;

  public MyTask  (String name) {
    threadName = name;
    sleepTime = 1000;
```

CONFIDENTIAL

```
  }

  public void run() {
    try {
      System.out.printf("%s going to sleep for %d m
illiseconds.\n", threadName, sleepTime);

      Thread.sleep(sleepTime); //  thread to sleep
    } catch (InterruptedException exception) {
      exception.printStackTrace();
    }
    System.out.printf("%s done sleeping\n", threadN
ame);
  }
}
```

4-Using Scanner class to receive user input

```
/**
 * This demo explains how Scanner class  is used in
Java.
 */


import java.util.Scanner;

public class MainClass{

  public static void main(String[] args){

    Scanner scanner = new Scanner (System.in);
    String s = scanner.next ();

    System.out.println(s);
  }

}
```

CONFIDENTIAL

5-Understanding FileInputStream and FileOutputStream

```
/**
 * This demo explains how File Stream is
implemented in Java.
 */

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;

public class FileInputOutputExample {
  public static void main(String[] args) throws Exc
eption {

    InputStream is = new FileInputStream("in.txt");
    OutputStream os = new FileOutputStream("out.txt
");
    int c;
    while ((c = is.read()) != -1) {
      System.out.print((char) c);
      os.write(c);
    }
    is.close();
    os.close();

  }
}
```

CONFIDENTIAL

6-Creating FileWriter using BufferedWriter

```
/**
 * This demo explains how FileWriter and
BufferedWriter  is implemented in Java.
 */
```

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.PrintWriter;

class TestClass {

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

    FileWriter fw = new FileWriter(args[0]);
    BufferedWriter bw = new BufferedWriter(fw);
    PrintWriter pw = new PrintWriter(bw, false);


    pw.println(true);
    pw.println('A');
    pw.println(500);
    pw.println(40000L);
    pw.println(45.67f);
    pw.println(45.67);
    pw.println("HelloWorld");
    pw.println(new Integer("99"));


    pw.close();
  }
}
```

CONFIDENTIAL

7-Understanding How to create scrollable ResultSet

```
/**
 * This demo explains how ResultSet is implemented
in Java.
Please change driver name and url as you change
your database.
 */
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class TestClass {
  public static void main(String[] args) throws Exc
eption {
    Class.forName("com.mysql.jdbc.Driver");
    Connection connection = DriverManager.getConnec
tion("jdbc:mysql://localhost/testdb", "root", "");

    Statement statement = connection.createStatemen
t(ResultSet.TYPE_SCROLL_INSENSITIVE,
        ResultSet.CONCUR_READ_ONLY);

    ResultSet resultSet = statement.executeQuery("S
ELECT * FROM products");
    connection.close();
  }
}
```

CONFIDENTIAL

8-Example to use updatable ResultSet

```
/**
 * This demo explains how updatable ResultSet is
implemented in Java.
 */

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class PrintResultSetClass {
  public static void main(String args[]) throws Exc
eption {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection con = DriverManager.getConnection("j
dbc:odbc:Contacts");
    Statement stmt = con.createStatement(ResultSet.
TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE
);
    ResultSet rs = stmt.executeQuery("select * from
 employee");
    rs.moveToInsertRow();

    rs.updateInt("Contact_ID", 150);
    rs.updateString("First_Name", "India");
    rs.updateString("Last_Name", "Bharat");

    rs.insertRow();
  }
}
```

CONFIDENTIAL

9-Example to use database Transaction

```
/**
 * This demo explains how Transaction can be
implemented in Java.
Please replace your driver name and url as you
change you database or driver.
 */


import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class TestClass {
  public static void main(String[] args) throws Exc
eption {
    Connection connection = null;
    try {
      Class.forName("com.mysql.jdbc.Driver ");

      connection = DriverManager.getConnection("url
","USERID", "PASSWORD");

      connection.setAutoCommit(false);

      Statement statement = connection.createStatem
ent();

      statement.executeUpdate("UPDATE Table1 SET Va
lue = 1 WHERE Name = 'xyz'");
      statement.executeUpdate("UPDATE Table2 SET Va
lue = 2 WHERE Name = 'abc'");

      connection.commit();

    } catch (SQLException ex) {
      connection.rollback();
    }

  }
}
```

10-Example to understand Generics

```
/**
 * This demo explains how Generics is implemented
in Java.
 */


public class TestGenericMethods {
public static <E> void displayArray(E[] inputArray)
 {
for (E element : inputArray)
System.out.printf("%s ", element);
```

CONFIDENTIAL

```
System.out.println();
}
public static void main(String args[]) {
Integer[] array = { 10, 20, 30, 40, 50, 60 };
System.out.println("The Array contains:");
displayArray(array);
}
}
```

End of Demo Document

CONFIDENTIAL