MultiThreadPrograms:

MultiDemo.java

import java.io.\*;

class MultiTask implements Runnable{

int availTickets;

//BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

MultiTask(int availTickets){

this.availTickets=availTickets;

}

public void run() {

for(int i=0;i<10;i++){

try{

task1();

}

catch(InterruptedException ie){

System.out.println("Task 1 interrupted");

}

try{

task2();

}

catch(InterruptedException ie){

System.out.println("Task 2 interrupted");

}

catch(IOException ie){

System.out.println("Task 2 interrupted");

}

/\*try{

task3();

}

catch(InterruptedException ie){

System.out.println("Task 3 interrupted");

}\*/

}

}

void task1() throws InterruptedException{

//Thread.sleep(1000);

System.out.println("I am Task1 \t"+Thread.currentThread().getName()+"\t available tickets \t "+availTickets);

}

synchronized void task2()throws InterruptedException,IOException{

//System.out.println("Enter number of tickets to be booked\t"+Thread.currentThread().getName());

//int wanted=Integer.parseInt(br.readLine());

int wanted=10;

if(availTickets<wanted){

System.out.println("Tickets Not Available\t");

System.exit(0);

}

else{

//synchronized(this){

Thread.sleep(1000);

availTickets-=wanted;

//}

System.out.println("I am Task2 \t"+Thread.currentThread().getName()+"\t available tickets \t "+availTickets);

}

}

}

class MultiDemo{

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

multiTasking();

}

static void multiTasking(){

MultiTask train1=new MultiTask(50);

MultiTask train2=new MultiTask(30);

Thread t1=new Thread(train1,"sudheer");

Thread t2=new Thread(train1,"Ramesh");

Thread t3=new Thread(train2,"Prasad");

Thread t4=new Thread(train2,"pavan");

t1.start();

t2.start();

t3.start();

t4.start();

}

}

SingleDemo.java

class SingleTask{

public void run() {

for(int i=0;i<10;i++){

try{

task1();

}

catch(InterruptedException ie){

System.out.println("Task 1 interrupted");

}

try{

task2();

}

catch(InterruptedException ie){

System.out.println("Task 2 interrupted");

}

try{

task3();

}

catch(InterruptedException ie){

System.out.println("Task 3 interrupted");

}

}

}

void task1() throws InterruptedException{

Thread.sleep(1000);

System.out.println("I am Task1"+Thread.currentThread().getName());

}

void task2()throws InterruptedException{

Thread.sleep(1000);

System.out.println("I am Task2"+Thread.currentThread().getName());

}

void task3()throws InterruptedException{

Thread.sleep(1000);

System.out.println("I am Task3"+Thread.currentThread().getName());

}

}

class SingleDemo{

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

long startsingle=System.nanoTime();

singleTasking();

long endsingle=System.nanoTime();

long totalsingle=(endsingle-startsingle);

System.out.println("single Tasking"+totalsingle);

}

static void singleTasking(){

SingleTask s1=new SingleTask();

SingleTask s2=new SingleTask();

SingleTask s3=new SingleTask();

SingleTask s4=new SingleTask();

s1.run();

s2.run();

s3.run();

s4.run();

}

}

MainThread.java

class MyThread1 extends Thread{

public void run(){

for(int i=0; i<10;i++){

System.out.println("child thread");

}

}

}

class MainThread{

public static void main(String args[]){

MyThread1 t1=new MyThread1();

t1.start();

for(int i=0; i<10;i++){

System.out.println("Main thread");

}

}

}

MultiDemo1.java

class MultiTask1 extends Thread{

MultiTask1(String s){

super(s);

}

public void run() {

//for(int i=0;i<10;i++){

try{

task1();

}

catch(InterruptedException ie){

System.out.println("Task 1 interrupted");

}

try{

task2();

}

catch(InterruptedException ie){

System.out.println("Task 2 interrupted");

}

try{

task3();

}

catch(InterruptedException ie){

System.out.println("Task 3 interrupted");

}

//}

}

void task1() throws InterruptedException{

Thread.sleep(1000);

System.out.println("I am Task1"+Thread.currentThread().getName());

}

void task2()throws InterruptedException{

Thread.sleep(1000);

System.out.println("I am Task2"+Thread.currentThread().getName());

}

void task3()throws InterruptedException{

Thread.sleep(1000);

System.out.println("I am Task3"+Thread.currentThread().getName());

}

}

class MultiDemo1{

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

//long startsingle=System.nanoTime();

singleTasking();

//long endsingle=System.nanoTime();

//long totalsingle=(endsingle-startsingle);

// System.out.println("single Tasking"+totalsingle);

}

static void singleTasking(){

MultiTask1 s1=new MultiTask1("Thread 1");

MultiTask1 s2=new MultiTask1("Thread 2");

MultiTask1 s3=new MultiTask1("Thread 3");

MultiTask1 s4=new MultiTask1("Thread 4");

s1.setPriority(1);

s2.setPriority(4);

s3.setPriority(8);

s4.setPriority(10);

s1.start();

s2.start();

s3.start();

s4.start();

StackTraceElement[] st1=s1.getStackTrace();

for(int i=0;i<st1.length;i++)

System.out.println("hello1"+st1[i]);

StackTraceElement[] st2=s2.getStackTrace();

for(int i=0;i<st2.length;i++)

System.out.println("hello2"+st2[i]);

StackTraceElement[] st3=s3.getStackTrace();

for(int i=0;i<st3.length;i++)

System.out.println("hello3"+st3[i]);

StackTraceElement[] st4=s4.getStackTrace();

for(int i=0;i<st4.length;i++)

System.out.println("hello4"+st4[i]);

}

}

ThreadLifeDemo.java

//import java.util.Random;

class ThreadLife implements Runnable{

//Random rn=new Random();

public void run(){

//for(int i=0;i<10;i++)

running();

}

void running(){

System.out.println("I am Running"+Thread.currentThread().getName());

//int n=rn.nextInt(10);

//System.out.println(n);

System.out.println("My state\t"+Thread.currentThread().getPriority()+"\t"+Thread.currentThread().getName()+"\t"+Thread.currentThread().getState());

Thread.yield();

try{

//Thread.sleep(1000);

Thread.currentThread().join(10);

throw new ArithmeticException("I called join"+Thread.currentThread().getName());

}catch(InterruptedException ie){

System.out.println(ie);

}

catch(ArithmeticException ae){

System.out.println(ae.getMessage());

}

catch(Exception ae){

System.out.println(ae.getMessage());

}

}

}

class ThreadLifeDemo{

public static void main(String args[]){

ThreadLife tl1=new ThreadLife();

ThreadLife tl2=new ThreadLife();

ThreadLife tl3=new ThreadLife();

Thread t1=new Thread(tl1,"Thread 1");

Thread t2=new Thread(tl2,"Thread 2");

Thread t3=new Thread(tl3,"Thread 3");

t1.setPriority(10);

t2.setPriority(4);

t3.setPriority(8);

t1.start();

t2.start();

t3.start();

try{

t1.join();

System.out.println("t1 after join\t"+Thread.currentThread().getName()+"\t"+t1.getState());

if(t1.isAlive())

System.out.println("t1 is alive\t"+t1.getState());

else

System.out.println("t1 is \t"+t1.getState());

Thread.sleep(1000);

if(t2.isAlive())

System.out.println("t2is alive\t"+t2.getState());

else

System.out.println("t2 is \t"+t2.getState());

t3.join(1);

System.out.println("t3 is alive\t"+t3.getState());

if(t3.isAlive())

System.out.println("t3 is alive\t"+t3.getState());

else

System.out.println("t3 is \t"+t3.getState());

}

catch(InterruptedException ie){

System.out.println("main method catch block");

ie.printStackTrace();

}

}

}

ThreadA.java

//import java.util.Random;

class ThreadLife implements Runnable{

//Random rn=new Random();

public void run(){

//for(int i=0;i<10;i++)

running();

}

void running(){

System.out.println("I am Running"+Thread.currentThread().getName());

//int n=rn.nextInt(10);

//System.out.println(n);

System.out.println("My state\t"+Thread.currentThread().getPriority()+"\t"+Thread.currentThread().getName()+"\t"+Thread.currentThread().getState());

Thread.yield();

try{

//Thread.sleep(1000);

Thread.currentThread().join(10);

throw new ArithmeticException("I called join"+Thread.currentThread().getName());

}catch(InterruptedException ie){

System.out.println(ie);

}

catch(ArithmeticException ae){

System.out.println(ae.getMessage());

}

catch(Exception ae){

System.out.println(ae.getMessage());

}

}

}

class ThreadLifeDemo{

public static void main(String args[]){

ThreadLife tl1=new ThreadLife();

ThreadLife tl2=new ThreadLife();

ThreadLife tl3=new ThreadLife();

Thread t1=new Thread(tl1,"Thread 1");

Thread t2=new Thread(tl2,"Thread 2");

Thread t3=new Thread(tl3,"Thread 3");

t1.setPriority(10);

t2.setPriority(4);

t3.setPriority(8);

t1.start();

t2.start();

t3.start();

try{

t1.join();

System.out.println("t1 after join\t"+Thread.currentThread().getName()+"\t"+t1.getState());

if(t1.isAlive())

System.out.println("t1 is alive\t"+t1.getState());

else

System.out.println("t1 is \t"+t1.getState());

Thread.sleep(1000);

if(t2.isAlive())

System.out.println("t2is alive\t"+t2.getState());

else

System.out.println("t2 is \t"+t2.getState());

t3.join(1);

System.out.println("t3 is alive\t"+t3.getState());

if(t3.isAlive())

System.out.println("t3 is alive\t"+t3.getState());

else

System.out.println("t3 is \t"+t3.getState());

}

catch(InterruptedException ie){

System.out.println("main method catch block");

ie.printStackTrace();

}

}

}

MainDataItem.java

//import java.util.Random;

class ThreadLife implements Runnable{

//Random rn=new Random();

public void run(){

//for(int i=0;i<10;i++)

running();

}

void running(){

System.out.println("I am Running"+Thread.currentThread().getName());

//int n=rn.nextInt(10);

//System.out.println(n);

System.out.println("My state\t"+Thread.currentThread().getPriority()+"\t"+Thread.currentThread().getName()+"\t"+Thread.currentThread().getState());

Thread.yield();

try{

//Thread.sleep(1000);

Thread.currentThread().join(10);

throw new ArithmeticException("I called join"+Thread.currentThread().getName());

}catch(InterruptedException ie){

System.out.println(ie);

}

catch(ArithmeticException ae){

System.out.println(ae.getMessage());

}

catch(Exception ae){

System.out.println(ae.getMessage());

}

}

}

class ThreadLifeDemo{

public static void main(String args[]){

ThreadLife tl1=new ThreadLife();

ThreadLife tl2=new ThreadLife();

ThreadLife tl3=new ThreadLife();

Thread t1=new Thread(tl1,"Thread 1");

Thread t2=new Thread(tl2,"Thread 2");

Thread t3=new Thread(tl3,"Thread 3");

t1.setPriority(10);

t2.setPriority(4);

t3.setPriority(8);

t1.start();

t2.start();

t3.start();

try{

t1.join();

System.out.println("t1 after join\t"+Thread.currentThread().getName()+"\t"+t1.getState());

if(t1.isAlive())

System.out.println("t1 is alive\t"+t1.getState());

else

System.out.println("t1 is \t"+t1.getState());

Thread.sleep(1000);

if(t2.isAlive())

System.out.println("t2is alive\t"+t2.getState());

else

System.out.println("t2 is \t"+t2.getState());

t3.join(1);

System.out.println("t3 is alive\t"+t3.getState());

if(t3.isAlive())

System.out.println("t3 is alive\t"+t3.getState());

else

System.out.println("t3 is \t"+t3.getState());

}

catch(InterruptedException ie){

System.out.println("main method catch block");

ie.printStackTrace();

}

}

}