Multi-Threading

import java.util.Date;

public class Main implements Runnable

{

Thread t;

static int[] a=new int[51];

static int sum=0;

Main(String name)

{

t=new Thread(this, name);

System.out.println("childthread:"+t);

t.start();

}

public void run()

{

System.out.println(Thread.currentThread().getName());

if(Thread.currentThread().getName().compareTo("one")==0)

{

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

{

sum=sum+a[i];

try

{

Thread.sleep(1000);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

System.out.println("Sum of 1-10 : " +sum);

}

//total=total+sum;

}

else if(Thread.currentThread().getName().compareTo("two")==0)

{

for(int j=10;j<20;j++)

{

sum=sum+a[j];

try

{

Thread.sleep(1000);

}

catch (InterruptedException e)

{

// TODO Auto-generated catch block

e.printStackTrace();

}

System.out.println("Sum of 10-20 : " +sum);

}

//total=total+sum;

}

else if(Thread.currentThread().getName().compareTo("three")==0)

{

for(int k=20;k<30;k++)

{

sum=sum+a[k];

try

{

Thread.sleep(1000);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

System.out.println("Sum of 20-30 : " +sum);

}

//total=total+sum;

}

else if(Thread.currentThread().getName().compareTo("four")==0)

{

for(int l=30;l<40;l++)

{

sum=sum+a[l];

try{

Thread.sleep(1000);

}catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

System.out.println("Sum of 30-40 : " +sum);

}

//total=total+sum;

}

else if(Thread.currentThread().getName().compareTo("five")==0)

{

for(int m=40;m<50;m++)

{

sum=sum+a[m];

try

{

Thread.sleep(1000);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

System.out.println("Sum of 40-50 : " +sum);

//total=total+sum;

}

//System.out.println("Total sum is : " +total);

}

}

public static void main(String[] args)

{

for(int x=0;x<51;x++)

{

a[x]=x+1;

}

System.out.println(Thread.currentThread().getName());

Main ob1=new Main("one");

Main ob2=new Main("two");

Main ob3=new Main("three");

Main ob4=new Main("four");

Main ob5=new Main("five");

Date start=new Date();

System.out.println("First Thread is Alive? : " +ob1.t.isAlive());

System.out.println("Second Thread is Alive? : " +ob2.t.isAlive());

System.out.println("Third Thread is Alive? : " +ob3.t.isAlive());

System.out.println("Fourth Thread is Alive? : " +ob4.t.isAlive());

System.out.println("Fivth Thread is Alive? : " +ob5.t.isAlive());

try

{

System.out.println("waiting for Threads to complete");

ob1.t.join();

ob2.t.join();

ob3.t.join();

ob4.t.join();

ob5.t.join();

}

catch (InterruptedException e)

{

// TODO Auto-generated catch block

e.printStackTrace();

}

System.out.println("Total sum is : " +sum);

System.out.println("First Thread is Alive? :"+ob1.t.isAlive());

System.out.println("Second Thread is Alive? :"+ob2.t.isAlive());

System.out.println("Third Thread is Alive? :"+ob3.t.isAlive());

System.out.println("Fourth Thread is Alive? :"+ob4.t.isAlive());

System.out.println("Fivth Thread is Alive? :"+ob5.t.isAlive());

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

Date end=new Date();

long difference=end.getTime()-start.getTime();

System.out.println("Whole process took "+difference/1000 +" " +"seconds");

System.out.println("Mian thread is exiting");

}

}





