A thread represents a separate path of execution of set of statements. The statements are executed by JVM one by one..This execution is called Thread…In every java program there is always a thread running internally .This thread is used by the JVM to execute the statements…

class RuningThread

{

public static void main(String args[])

{

Thread t=Thread.currentThread();

System.out.println("Current Thread is :"+t);

System.out.println("Thread name is :"+t.getName());

}

}

o/p….Thread[main,5,main]

main

In the output **main** indicates the name of the Thread and 5 is the priority of the thread… priority are from 1 to 10..

A Thread represents the execution of the statements …

**Uses of Threads**

1. These are mainly used in server side program to server needs of multiple clients

2. Used to create Games and animations

Creation of Thread and Running

We know that ther is a main thread .. apart from this main thread we can create our own threads …

1.create a class that extends thread class or implements runnable interface which are in java.lang package

Class Myclass extends Thread

( Or)

Class Myclass implements Runnable

2. write run method in this class

Public void run()

{

}

3.create object to Myclass

Myclass obj=new Myclass();

4.create Thread and attach to Object obj

Thread t=new Thread(obj);

Or

Thread t=new Thread(obj,”ThreadName”);

5.Run the thread …..(use start() mathod)

t.start();

import java.lang.\*:

**Single Tasking**

//single tasking using Threads

class MyThread implements Runnable

{

public void run()

{

task1();

task2();

task3();

}

void task1()

{

System.out.println("This is from Task1");

}

void task2()

{

System.out.println("This is from Task2");

}

void task3()

{

System.out.println("This is from Task3");

}

}

class Single

{

class Myclass extends Thread

{

public void run()

{

for(int i=01;i<=1000;i++)

System.out.println(i);

}

public static void main(String args[])

{

Myclass obj=new Myclass;

Thread t=new Thread(obj);

t.start() ;

}

}

Thread class Methods:

t.start();….to start the thread

Thread.sleep(Millsec);…stop execution for a specied time

String name=t.getName()…gets the name of the thread

t.setName(“ThreadName”);…set the name of the Thread

t.isAlive()…to test a thread is still alive(Boolean);

t.join()……waite until the thread dies…

Thread class Methods:

1.to know the currently running Thread

Thread t=Thread.currentThread();

2.To Start a thread

t.start();

3.To stop execution of a thread for a specied time

Thread.sleep(milli seconds);

4.To Get the name of a Thread

String str=t.getName();

5.To set new name to a thread

t.setName(“new name”);

6.To get the priority of the thread

Int p=t.getPriority();

7.To set the Priority of the thread

t.setPriority(int p);

Thread Priority can be changed from 1 to 10

Thread.MAX\_PRIORITY value 10

Thread.MIN\_ PRIORITY value 1

Thread.Norm\_ PRIORITY value 5

Multi tasking using Threads

class MyThread implements Runnable

{

String str;

MyThread(String str)

{

this.str=str;

}

public void run()

{

for(int i=1;i<=10;i++)

{

System.out.println(str+" : "+i);

try

{

Thread.sleep(10);

}

catch(InterruptedException e)

{

System.out.println(e);

}

}

}

}

class Multi

{

public static void main(String args[])

{

MyThread obj=new MyThread("Cut the Ticket");

MyThread obj1=new MyThread("Show the Ticket");

Thread t1=new Thread(obj);

Thread t2=new Thread(obj1);

t1.start();

t2.start();

}

}

**Multiple Threads on a single object**

class Reserve implements Runnable

{

int av=1;

int w;

Reserve(int i)

{

w=i;

}

public void run()

{

System.out.println("Avalible Bearths are:"+av);

if(av>=w)

{

String name=Thread.currentThread().getName();

System.out.println(w+" Bearths reserverd for "+name);

try

{

Thread.sleep(1500);

av=av-w;

}

catch(InterruptedException ie)

{}

}

}

}

class Unsafe

{

public static void main(String args[])

{

Reserve obj=new Reserve(1);

Thread t1=new Thread(obj);

Thread t2=new Thread(obj);

t1.setName("FirstPerson");

t2.setName("SecondPerson");

t1.start();

t2.start();

}

}

**What is Thread Synchronization**

When a thread is acting on a object ,preventing any other thread from acting on the same object is called “Thread Safe” The object on which the synchronization is called Synchronized object….Thread Synchronization is recommended when multiple threads are used on the same object…(in multithreading)

Threr are two ways to Synchronized the object

1. Using Synchronized block..

Synchronized(object)

{

Statements;

}

The statements which in the block are available to one thread at a time…

1. Using synchronized key word..

Synchronized void display()

{

Statement;

}

Display method is available to only one thread at a time

Difference b/w synchronized block and synchronized keyword:

synchronized block used to synchronize block of statements but where as synchronized keyword is used to synchronize entair method..