

14 May 2023

Thread class - inbuilt class

methods of Thread class -

sleep() method

start() method

run() method

DUAL CAMERA  
shot by Nidhu

2023/06/02 21:54

Thread बनाने के दो तरीके होते हैं-

1 - thread class को extend करके

2 - runnable interface को implement करके

in our program Snip2 we have used 1st way.

Thread - How to make thread

# By extending Thread class

# By implementing Runnable interface

Thread → important methods run, start, sleep

# run method runs automatically when we call start();

Runnable interface → only run method. then how will we start it?

Thread class - inbuild class

methods of Thread class -

sleep() method

start() method

run() method

# Snp2.java in this program we have

without synchronization → one resource

and two threads

// in output we find overlapping

अगर synchronization नहीं है, तो आप एक thread ने थोड़ी भी देर के लिए भी resource को use करना बंद कर्या, तो दुसरा thread उसे use करने के लिए इसे जारी।

line 12, 20 linking (resource का attachment thread से)

resource class ⇒ Table, we have one method printTable in this.  
we used parameter इसका मतलब formal argument है, एवं value actual argument से मिलेगी।

try catch ⇒ अब किसी thread को sleep करते हैं through sleep method

400 millisecond = 0.4 second

1 second = 1000 millisecond

→ इसका use करने पर exception आने का खतरा होता है, so try catch

For multithreading  $n$  threads and  $(n-1)$  or less than  $(n-1)$   
resources are required ( $1 \rightarrow (n-1)$ ).  
AI DUAL CAMERA  
Shot by Nidhu

```
1 class Table // resource class
2 {
3     void printTable (int n){
4         for(int i= 1;i<=5;i++)
5         {
6             System.out.println(n*i);
7             try{Thread.sleep(100);}
8             catch(Exception e)
9             { System.out.println(e); }}}
10
11 class MyThread1 extends Thread{
12     // our own thread class where we create thred 1
13     Table t ;
14     MyThread1(Table t)
15     {this.t=t;}
16     public void run()
17     {t.printTable(5);}
18 }
19
20 class MyThread2 extends Thread{
21     // our second own thread class where we create thred 2
22     Table t ;
23     MyThread2(Table t)
24     {this.t=t;}
25     public void run()
26     {t.printTable(100);}
27 }
28 public class Snp2 { // program test class which have main function
29     public static void main(String args[])
30     { Table obj = new Table();
31         MyThread1 t1 = new MyThread1(obj);
32         MyThread2 t2 = new MyThread2(obj);
33         t1.start();
34         t2.start();}}
35 }
```

```
Command Prompt
C:\Users\Shubhams-PC\Des
100
5
200
10
15
300
400
20
500
25
```

```
C:\Users\Shubhams-PC\Desktop
```

Line 9, Column 38

Type here to search



sleep method use करने का मतलब ये है कि जो भी thread उस resource का उपयोग कर रहा, वह तो resource छोड़ा की वो thread शोड़ी दें के लिए waiting condition में चले जाए।

thread बनाने के दो तरीके होते हैं-

- 1 - thread class को extend करके
- 2 - runnable interface को implement करके

in our program Snip2 we have used 1st way.

Thread class belongs from java.lang package but we do not have need to import this, it is automatically imported in all java programs.  
methods of Thread class  $\Rightarrow$  sleep, run, start

So now we  $\Rightarrow$  create a class MyThread that extends Thread class.

in this line 12, we have created a ref. of resource class.  
ताकि thread को पता चल सके कि इस resource को use करना है (connectivity बनाने के)

First constructor

इसी दृष्टि से जैसे जैसे बनाया था। अब MyThread (Table 3)

constructor का लिया और इसके अंदर this.t = t;

actual argument  $\rightarrow$  formal arg.  $\rightarrow$  जौही इसके value ref. var.  
जहाँ से ~~this~~ object (Table t) द्वारा सिर्फ मतलब object देनेंगा।

Second method इसने run method (this is an overridden method)  
and compulsory इसे बनाना है, कि start  
method जैसे ही आएगी वो run को हुएगी।

thread को run करने का काम  
run method करेगी।



AI DUAL CAMERA  
Shot by Nidhu

2023/06/02 22:33

A S D F G H J K

Thread + ; } लिखने का reson हो था  
Table + ; कि Mythreadclass को resource class का use

करना था, तो simple स्या association का concept लाता है कि

ये statement हमें Mythread class के अंदर लिखा है,

मतलब हमने

reference var. बनाया Table class का, जो कि हमारी resource class है।

इसलिए बनाया कि हम use

कर रहे हैं method, resource class की, और हम connectivity बना रहे हैं।

तो अब तक हमें MyThread नाम के constructor की help से Table class का object पुरी तरह से लिख गया है।

अब हमने एक method बनाया public void run()

run method Thread class के अंदर ही होता है और Runnable interface के अंदर ही होता है। हम इसके override करते हैं, # compulsory override कहा गया कि interface की method है,

इस object से हम resource doi method को call कर पाएंगे।

अब हमें object को डिक्टना complicated बनाया, जिसकी class (Thread) के अंदर बना सकते हैं new keyword में।

Ans. → main fun. के अंदर एक object बनाया है उसको हम आप threads में pass करवा पा रहे हैं।

आपका thread class के लिए डिलग-उल्लग object बनाना पड़ता

R AI DUAL CAMERA

Shot by Nidhu

2023/06/02 22:33

without synchronized program (resource) में →

- जब sleep method चलनी उसके parameter में आपने जितना time दिया है ( Thread.sleep(100); ) (100 milisecond) , we can not give negative time तो thread1 द्वारा कौन से time के लिए use करना बहुत बड़ा देगा )

# इस free time में operating system ⇒ Context switching के concept के according, OS thread2 को उस resource का use कर लेने देगा ,

# Sleep method का time खत्म होने से पहले thread2 को resource का use करना बहुत कूटा दिया जाएगा OS. के द्वारा और Thread1 को resource है दिया जाएगा इस कारण overlapping होगी output में, Thread1 & 2 का output overlap होकर आएगा ।

तो अगर हम पाते हैं कि overlapping ना हो, मतलब एक thread1 उस resource का पूरा use कर ले, उसके बाद ही thread2 उस resource का use कर पाए ।

तो हम synchronization concept का use कर सकते हैं ।

इस synchronization concept लाने के लिए resource class के अंदर synchronization concept लाने के लिए

\* methods के आगे synchronized (keyword) लिख लकड़े हैं

\* synchronized block भी बना सकते हैं

→ # Smp3.java

AI DUAL CAMERA

Shot by Nidhu

2023/06/02 22:34

More than two resources  
OS में तो sleep method नहीं होती पर ऐसे imagine threads ( $P_1$ ) Process  
Threads ( $P_2$ ) Process

Resource ( $R_1$ )  
Resource ( $R_2$ )

### Synchronization

$P_1$ ,  $R_1$  को use कर रहा था, पर उसे  $R_2$  use करने की ज़ज़रत पर गया, तो  $R_2$  use करते रहा गया लेकिन  $R_2$  को कोई और process ( $P_2$ ) use कर रही नहीं, अब  $P_2$ ,  $R_1$  को use करना चाहता है पर वो नहीं कर पाएगा।

और  $P_3$  को बीच से  $R_1$  को unlock नहीं गयी तो deadlock condition आ जाएगी।

### Synchronized

$T_1$  using  $R_1$ , अगर sleep mode पर भी रहा गया तब भी अभी यहि  $T_1$  ही use कर सकता है resource को पुरा use करना उपर्युक्त बाद ही कोई दुश्यमा नहीं कर पाएगा।

what is deadlock ? and for avoiding it what we do in java.

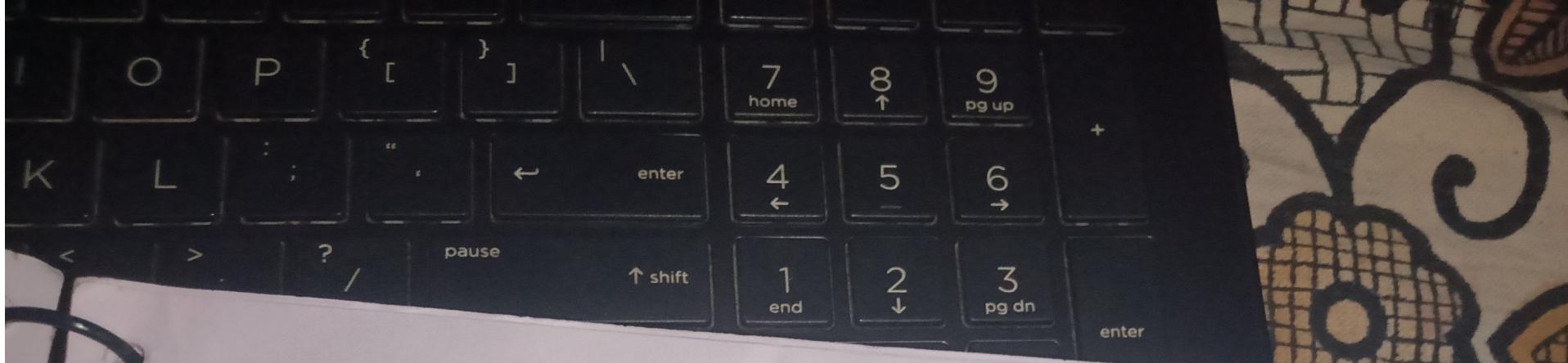
$T_1$  using | holding  $R_1$  but requirement is for  $R_2$

$T_2$  using | holding  $R_2$  but requirement is for  $R_3$

$T_3$  using | holding  $R_3$  but requirement is for  $R_1$

sleep method नहीं होगी तो deadlock condition आ जाएगी

अगर हमने synchronization use किया है तो इस forcefully  $T_1$  को बोलते ही  $R_1$  का पुरा use करें। तो deadlock नहीं होगा।



resource 1 ये हमें ही values मिल रही हैं।

resource 2 वो हम करने add करना पड़ते हैं।

Thread 2 using R2 पर requirement तो R1 की है।  
उसका synchronization करना भी हम तो add करने के लिए  
value नहीं मिल पाएगी तब भी करना तो पड़ता  
तो हम expected o/p जैसी मिल पाएंगा।



Date  
17 - May - 2023

## Multi Threading -

Program → Run  
( RAM )  
— Process —

# A program in execution is called process .

# A process have one thread or more then one thread .

threads can do context switching -

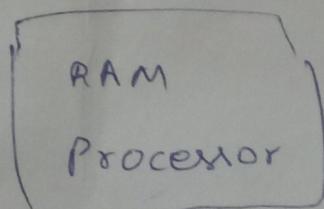
in Multiple threading [more then one thread works together , and do context switching .]

in context switching , current state is being saved

Waiting state

New state

Running state



Read only memory - you cannot modify CPU  
Rom have bootstrap program , this will helps to load operating system

~~context switching~~ की help से processor and RAM की  
thread की sharing background में फैल जाती है  
and it takes very little time, we can not even feel  
this time. (Light weight process)

Thread - How to make thread

- # By extending Thread class
- # By implementing Runnable interface

Thread  $\Rightarrow$  important methods run, start, sleep

# run method runs automatically when we call start();

Runnable interface  $\Rightarrow$  only run method. then how will  
we start id?

### # Trdeg1.java

Thread class belongs to java package.

→ If run method not defined, then error नहीं आया।  
परन्तु Thread class में run की definition है ( {} ) इसी बजाए ही  
Thread parent class है गलत child class बनाता है यह गलत है।

void run() { }      // method which have definition  
                      // but def. में कुछ नहीं है।

# here run method is not abstract गलत होती है तो error आती।  
और Thread class भी abstract हो जाती।

(run, runn,) are diff. then run



AWTmouseeg.java

Snp2.java

AWTpaneleg2.java

AWTfbpage.java

Trdeg1.java

untitled

AWTcheckbox4.java

```
2 //Java Thread Example by extending Thread class
3
4
5 class Trdeg1 extends Thread{
6
7     public void run(){
8
9         System.out.println("thread is running...");}
10
11
12     public static void main(String args[]){
13         Trdeg1 er = new Trdeg1();
14         er.start();}
15
16     }
17 }
18
19
20 //Output:
21
22 //thread is running
23
```

## Command Prompt

Microsoft Windows [Version 10.0.18362.1256]  
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Shubhams-PC>cd C:\Users\Shubhams-PC\Desktop\javafolder>

C:\Users\Shubhams-PC\Desktop\javafolder>javac Trdeg1.java

C:\Users\Shubhams-PC\Desktop\javafolder>java Trdeg1.java

thread is running...

C:\Users\Shubhams-PC\Desktop\javafolder>

## # Trdeg2.java

We have implemented Runnable interface

you must have to override run() method

if you do not, then compile time error will be thrown

run() method के लिए start method चाहिए होती है,

पर Runnable interface में तो इसके run() method हैं।

तो हमें तो start method चाहिए, जो कि Thread class में है।

तो हम inbuild Thread class का object अपनी class में बना देंगे (मतलब Association होगा)। तो हम



and इस object की linking हम normal class के object से

Thread t = new Thread(m1);



◀ ▶ AWTmouseeg.java | Snp2.java ● | AWTpaneleg2.java ✘ | AWTfbpage.java ✘ | Trdeg1.java ● | Trdeg2.java | untitled | AWTcheckbox4.java

```
1 // Java Thread example by implemeting Runnable interface
```

```
2  
3  
4 class Trdeg2 implements Runnable{  
5  
6     public void run(){  
7         //public void ruun(){  
8         System.out.println(Thread.currentThread().getName()+" "+"thread is running...");  
9     }  
10  
11    public static void main(String args[]){  
12        Trdeg2 m1=new Trdeg2();  
13        Thread t1 =new Thread(m1); // Using the constructor Thread(Runnable r)  
14        t1.start();  
15    }  
16    }  
17    }  
18    }  
19  
20 //Output:  
21
```

Command Prompt

```
C:\Users\Shubhams-PC\Desktop\javafolder>javac Trdeg2.java  
C:\Users\Shubhams-PC\Desktop\javafolder>java Trdeg2  
Thread-0 thread is running...
```

```
C:\Users\Shubhams-PC\Desktop\javafolder>
```

\*\*\* and इस object की linking से normal class के object को  
मालिन Thread t = new Thread(m1);

### # TMWiththreading1.java

Thread t = Thread.currentThread(); → returns a object of Thread class  
 4. getName(); → gives name of thread (String - Thread)      10, 26, 36, 45  
 5. getId(); → gives id of thread (Integer -      or ...)  
 setName (" "); → we can set name

### Methods of Thread Class

```

AWTmouseeg.java | Snp2.java • | AWTpaneleg2.java ✘ | AWTfbpage.java ✘ | Trdeg1.java • | Trdeg2.java ✘ | TMultithreading1.java ✘ | untitled • | AWTcheckbox4.java
import java.util.*;
//java code for thread creation by extending
// Thread class
class Multithreading extends Thread{
    public void run()
    {
        try{ // Displaying the thread that is running
            Thread t = Thread.currentThread();
            System.out.println("My name : " +t.getName());
            System.out.println("my id : "+t.getId());
            /* or we can write System.out.println
            ("Thread"+Thread.currentThread().getId()+"is running");*/
        }
        catch( Exception e)
        {
            // Throwing an exception
            System.out.println("Exception is caught ");
        }
    }
    // main class
    public class TMultithreading1{
        public static void main(String[] args)
        {
            int n = 8;//number of threads
            for(int i = 0 ;i<n;i++)
            {
                Multithreading object = new Multithreading();
                object.start();
            }
        }
    }
}

```

## Command Prompt

```

C:\Users\Shubhams-PC\Desktop\javafolder>javac TMultithreading
Note: TMultithreading1.java uses or overrides a deprecated API
Note: Recompile with -Xlint:deprecation for details.

C:\Users\Shubhams-PC\Desktop\javafolder>java TMultithreading1
My name : Thread-5
My name : Thread-2
My name : Thread-7
My name : Thread-4
My name : Thread-0
My name : Thread-3
My name : Thread-1
My name : Thread-6
my id : 22
my id : 21
my id : 26
my id : 23
my id : 24
my id : 25
my id : 28
my id : 27

```

C:\Users\Shubhams-PC\Desktop\javafolder&gt;

Activate  
Go to Settings



AI DUAL CAMERA  
Shot by Nidhu



21 May 2023

Thread can be created by - 1 extending Thread class

2. Implementing Runnable interface

# अगर Runnable interface को implement करवाया है, तो run method को override करना ही पड़ा।

\* we do not have start() method in Runnable interface, so we create Thread class's object and link it with normal class's object then we are able to call start() and start() will call run() method.

# Trdeg2.java as we done in previous program

But now? \* क्या आप run() method को direct call करना चाहते हैं?  
→ है, करना चाहते हैं।

means इस start() का use ही ना करें।

modified # Trdeg2.java → # Trdeg22.java

class Trdeg2 implements Runnable  
{  
 p.v. run()  
 {  
 laptop();  
 }  
}

रही पर उग्र एवं Thread का नाम जानना पड़े, तो getName() का use करें। Thread.currentThread.getName();

तो हो main मिलेगा।

→ ये default thread होंगा, यह Thread class का कोई thread नहीं बना, मतलब multithreading concept use ही नहीं हुआ।

main thread वाली threads (आज बनते हैं) तो उनको तभी चलाता है, जब start method बनी हों।

Edit Selection Find View Goto Tools Project Preferences Help

AWTmouseeg.java | Snp2.java • | AWTpaneleg2.java ✘ | AWTfbpage.java ✘ | Trdeg1.java • | Trdeg2.java ✘ | TMultithrea

```
1 // Java Thread example by implemeting Runnable interface
2
3
4 class Trdeg22 implements Runnable{
5
6     public void run(){
7
8         System.out.println(Thread.currentThread().getName()+" "+"thread is running...");
9     }
10
11    public static void main(String args[]){
12        Trdeg22 m1=new Trdeg22();
13        Thread t1 =new Thread(m1); // Using the constructor Thread(Runnable r)
14        // start method ka use hee na kre , to multithreading
15        // concept use hee nhi hogा
16        t1.run();
17
18    }
19
20 }
21
22 //Output:
23
```

Command Prompt  
C:\Users\Shubhams-PC\Desktop\javafolder>javac Trdeg22.java  
  
C:\Users\Shubhams-PC\Desktop\javafolder>java Trdeg22  
main thread is running...

# start() method thread का instance return करता है, तो run() method को background में call करेगा।

main thread ये multithreading का concept explain करना ही क्या जो करता।

उत्तर M.T. concept explain करना है, तो thread बनाने पर्सी। start method, thread का object use करेगी।

thread को नहीं पड़ा। start method, thread को object use करने की तरीकी।

# अगर हमारा thread एक बना है और हमने start() method को दो बार call कर दिया, मतलब एक ही thread को दो बार start कर दिया, तो क्या होंगा?

# Thread2.java

Ans ⇒ एक thread के प्रारंभिक विषय, पर  
first exception आएगा।

# एह ऐ thread को जिस एक बार ही start कर दिया है

IllegalThreadStateException

Qki Multithreading से thread context switching को use करेंगे,  
resource को लेण्ड problem भी सकती है,

Mult.T. से multiple thread parallelly चलते हैं



AWTmouseeg.java | Snp2.java ● | AWTpaneleg2.java ✘ | AWTfbpage.java ✘ | Trdeg1.java ● | Trdeg2.java | TMultithreading1.java ✘ | Trdeg22.java

// Java Thread example by implemeting Runnable interface

```
1 class Trdeg22 implements Runnable{  
2  
3     public void run(){  
4  
5         System.out.println(Thread.currentThread().getName()+" "+"thread is running...");  
6     }  
7  
8     public static void main(String args[]){  
9         Trdeg22 m1=new Trdeg22();  
10        Thread t1 =new Thread(m1); // Using the constructor Thread(Runnable r)  
11        // sirf ek thread bana h t1 , isse do bar startmehtod call krwa dee  
12  
13        t1.start();  
14        t1.start();  
15  
16    }  
17  
18    }  
19 }  
20 }  
21  
22 //Output:  
23
```

Command Prompt

```
C:\Users\Shubhams-PC\Desktop\javafolder>java Trdeg22.java  
Exception in thread "main" Thread-0 thread is running...  
java.lang.IllegalThreadStateException  
        at java.base/java.lang.Thread.start(Thread.java:1534)  
        at Trdeg22.main(Trdeg22.java:17)
```

C:\Users\Shubhams-PC\Desktop\javafolder&gt;

HQ

# यार इन्हें sleep( ) method के अंदर negative value दी  
तो क्या होगा ?

# Snpr.java

तो runtime exception होगा।

Java.lang.illegalArgumentExeception : timeout value is negative

illegalArgumentExeception कहाँ आता है ? 1. जब sleep( ) के अंदर  
negative value दी है ।

2. ...

Selection Find View Goto Tools Project Preferences Help

AWTmouseeg.java Snp2.java AWTpaneleg2.java AWTfbpage.java Trdeg1.java Trdeg2.java TMultithreading1.java Trdeg22.java untitled

```
class Table // resource class
{
    void printTable (int n){
        for(int i= 1;i<=5;i++)
        {
            System.out.println(n*i);
            try{Thread.sleep(-100);}
            catch(Exception e)
            { System.out.println(e); }}}

class MyThread1 extends Thread{
    // our own thread class where we create thred 1
    Table t ;
    MyThread1(Table t)
    {this.t=t;}
    public void run()
    {t.printTable(5);}

}

class MyThread2 extends Thread{
    // our second own thread class where we create thred 2
    Table t ;
    MyThread2(Table t)
    {this.t=t;}
    public void run()
    {t.printTable(100);}

}

public class Snp2 { // program test class which have main function
    public static void main(String args[])
    { Table obj = new Table();
        MyThread1 t1 = new MyThread1(obj);
        MyThread2 t2 = new MyThread2(obj);
        t1.start();
        t2.start();}

}
```

Command Prompt

```
C:\Users\Shubhams-PC\Desktop\javafolder>java Snp2
5
100
java.lang.IllegalArgumentException: timeout value is negative
200
java.lang.IllegalArgumentException: timeout value is negative
10
java.lang.IllegalArgumentException: timeout value is negative
300
java.lang.IllegalArgumentException: timeout value is negative
java.lang.IllegalArgumentException: timeout value is negative
400
java.lang.IllegalArgumentException: timeout value is negative
500
15
java.lang.IllegalArgumentException: timeout value is negative
java.lang.IllegalArgumentException: timeout value is negative
20
25
java.lang.IllegalArgumentException: timeout value is negative
java.lang.IllegalArgumentException: timeout value is negative
C:\Users\Shubhams-PC\Desktop\javafolder>
```

AI DUAL CAMERA  
Shot by Nidhu

Type here to search

O D E M S G

2023/06/10 04:11

## join method

22 - May - 2023

# Tjoinmthd.java long example

# Tjoinmthd1.java

→ here you will notice thread 1 first complete its task thanks to join method / operation.

then thread2 and thread3 will execute.

explained →

start() method को call नहीं किया तो thread बनेगा ही नहीं।

run() method को call नहीं किया तो thread चलेगा ही नहीं।

```
import java.util.*;
```

```
class Tjoinmthd1 extends Thread
```

```
{ public void run()
```

```
{ for (int i=1; i<=4; i++)
```

```
{ try
```

```
{ Thread.sleep(500); }
```

```
catch (Exception e)
```

```
{ s.o.p.(e); }
```

```
s.o.p.(i);
```

```
}
```

p. S. V. main (String[] args)

```
{ Tjoinmthd t1 = new Tjoinmthd1();
```

```
Tjoinmthd t2 = new Tjoinmthd1();
```

```
Tjoinmthd t3 = new Tjoinmthd1();
```

```
t1.start();
```

```
try { t2.join();
```

```
catch (Exception e)
```

```
{ System.out.p(e); }
```

```
t2.start(); -----
```

```
t3.start(); -----
```

```
}
```

```
}
```

extended Thread class  
for creating thread.

run method को override किया  
क्यि start() method direct run  
method को कूली।

↪ run के अंदर हमें for loop  
से 1 print करवाना पाएँ।

पर बीच में हमें sleep() method  
का use भी किया।

- अब sleep method को try में  
कौन से लिखा?

Ans ⇒ critical situation से हो सकता  
है thread, और उसको sleep mode  
पर लाने से exception आ सकता है।

sleep mode पर जाने के  
पास thread उपर्युक्त current  
position को store करके उसका  
आगे memory नहीं। फिर से exception  
आ सकता है।

↪ हमें हमें 3 thread बनाए  
क्यि 3 object हाइ Thread & क्यि  
इन Thread को extend किया है।

→ अब हमें पहले thread को start  
कर दिया, तो run को call करेंगा।

sleep चला तोhen s.o.p से 1 print हुआ।

अब sleep चला, तो compiler ने क्या  
महां कराएँ, वो Thread2 को start  
करने आएगा। पर बीच में join method  
उपर्युक्ती।

C:\Users\Shubhams-PC\Desktop\javafolder\Tjoinmthd1.java - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

AWTmouseeg.java Tjoinmthd1.java AWTpaneleg2.java AWTfbpage.java Trdeg1.j

```
2 import java.util.*;
3 class Tjoinmthd1 extends Thread{
4
5     public void run()
6     {
7         for(int i=1;i<=4;i++)
8         {
9             try
10            {
11                Thread.sleep(1000);
12            }
13            catch(Exception e)
14            {
15                System.out.println(e);
16            }
17            System.out.println(i);
18        }
19    }
20    public static void main(String args[])
21    {
22        Tjoinmthd1 t1 = new Tjoinmthd1();
23        Tjoinmthd1 t2 = new Tjoinmthd1();
24        Tjoinmthd1 t3 = new Tjoinmthd1();
25
26        t1.start();
27        try{
28            t1.join();
29        }
30        catch(Exception e)
31        {
32            System.out.println(e);
33        }
34        t2.start();
35        t3.start();}}
```

Command Prompt

Microsoft Windows [Version 10.0.18363]

(c) 2019 Microsoft Corporation. All Rights Reserved.

C:\Users\Shubhams-PC>cd C:\Users\Shubhams-PC\Desktop\javafolder

C:\Users\Shubhams-PC\Desktop\javafolder>

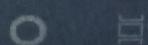
```
1
2
3
4
1
1
2
2
3
3
4
4
```

C:\Users\Shubhams-PC\Desktop\javafolder>

Line 1, Column 1



AI DUAL CAMERA  
Shot by Nidhu



2023/06/10 04:14

join method def. →

ये current thread को waiting stat पर डालेगा, और जिस thread ने join method को call किया है उसको completely चलाएगा।

\*\* Imp.

अब compiler ने thread 1 को तो sleep() पर डाल दिया था, मतलब ये current thread नहीं रहा,

और thread 2 के start होने के पहले join method मिल गयी।

तो join method के लिए current thread कौन-क्या है? जिसको join method waiting stage पर पहुंचाएगी?

Answer - main thread

तो, t2.join() से, main thread waiting पर गया और, तो अपना काम पूरी तरह से finish करेगा अब, इसके बाद ही कोई दुसरे thread start हो पाएगा।

t2 से 1, 2, 3 print हो जाएगा।

according to our code **# Tjoinmethd1.java**

then t2 & t3 start होंगे, इनमें join method use नहीं हुई है तो overlapping होगी

O/P है 1, 1, 2, 2, 3, 3 print होगा।



AI DUAL CAMERA

Shot by Nidhu

1 2 3 1 1 2 2 3 3

अगर join method नहीं होती

t1.start से 1 print हुआ।

फिर sleep हो गया second thread का t2.start से 1 print हुआ।

फिर sleep हो गया, third thread start हुआ और 1 print किया।

अब t3 sleep हुआ,

तो आपना t1 ~~start~~ फिर से आपना काम चालू करेगा। ~~start~~ print 2 होंगा।

फिर t2 से sleep हो गया।

फिर t2 से print 2 होगा। और sleep होगा।

फिर t3 से 2 print होगा। sleep हो जाएगा।

अब t1 फिर से t2 3 print करेगा। और sleep हो जाएगा।

t2 print करेगा। 3 then sleep हो जाएगा।

t3 print करेगा। 3.

2023/06/10 04:15

# Tjoinmthd.java

code is given try to  
understand and explain  
by yourself

&& prefer recording for  
better understanding

```
1 // A program for understating joining of threads
2 import java.io.*;
3 // The ThreadJoin class is the child class of the class Thread
4 class ThreadJoin extends Thread
5 {
6     //overriding the run mehtod
7     public void run()
8     {   for(int j= 0;j<2;j++)
9     {
10         try
11             {//sleeping the thread for 300 mili seconds
12                 Thread.sleep(300);
13                 System.out.println("The current threaad name is :"+Thread.currentThread().getName());
14             catch ( Exception e)
15                 {System.out.println("the exception has benn caught : "+ e);}
16
17                 System.out.println(j);
18
19             }
20     }
21     class Tjoinmthd extends Thread{
22
23         public static void main(String[] args)
24         {
25             // Creating Threds
26             Tjoinmthd th1 = new Tjoinmthd();
27             Tjoinmthd th2 = new Tjoinmthd();
28             Tjoinmthd th3 = new Tjoinmthd();
29
30             // thread th1 starts
31             th1.start();
32             // Starting the second thread after when
33             // the first thread th1 has ended or died
```

Line 1, Column 1



Type here to search



```
AWTmouseeg.java | Tjoinmthd1.java | Tjoinmthd.java | AWTpanleg2.java | AWTfbpage.java | Trdeg1.java | Trdeg2.java | TMultithreading1
20 // thread th1 starts
21 th1.start();
22 // Starting the second thread after when
23 // the first thread th1 has ended or died
24 try{
25     System.out.println("The current thread name is : "+currentThread().getName());
26     // invoking the join() method
27     th1.join();
28 }
29 catch(Exception e)
30 {
31     System.out.println("The exception has benn caught");
32 }
33 // starting th2 starts
34 th2.start();
35 //starting the th2 thread after when the thread th1 has endded or died
36 try{
37     System.out.println("The current thread name is : "+currentThread().getName());
38     // invoking the join() method
39     th2.join();
40 }
41 catch(Exception e)
42 {
43     System.out.println("The exception has benn caught"+e);
44 }
45
46 // thred th3 starts
47 th3.start();
48 }
49
50
51
52
53
54
55
56
57
58
59 }
60
61
62
```

'The exception has been caught");

ad after when the thread th1 has ended or died

"The curr  
n() metho

C:\Users\Shubhams-PC\Desktop\javafolder>javac Tjoinmthd.java

C:\Users\Shubhams-PC\Desktop\javafolder>java Tjoinmthd

The current thread name is : main

The current thread name is : main

("The exce

C:\Users\Shubhams-PC\Desktop\javafolder>

Overall → in Multithreading concept , we creat more then two threads , and the top thread main thread main thread will start invite all the threads by using start method .

& main thread of JVM start badi है ।

Multithreading - 5 imp q.

18 May 2023

## Synchronization

Synchronization → जो काम करना है वह एक activity है।

### Synchronization in Java

Synchronization in Java is the capability to control the access of multiple threads to any shared resource.

Java synchronization is better option where we want to allow only one thread to access the shared resource. मतलब एक टाइम में एक ही thread ने कोई resource को

### Why use Synchronization?

⇒ The synchronization is mainly used to -

To prevent thread interference. → मतलब वीन-मा thread resource को use करेगा। इस मास्टर में आगा ना हो।

To prevent consistency problem.

### Types of Synchronization

shared resource

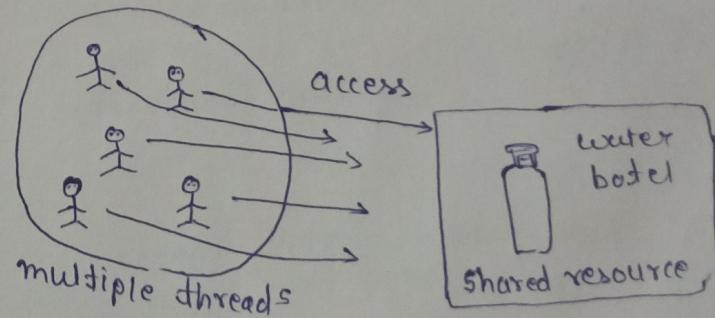
मतलब, एक share  
एक रहे जैसे वर्त

resource की ।  
(T.V., Auto, bottle)

in computer terms

→ keyboard

→ Memory



multiple threads shared resource को use  
करने के लिए, through Synchronization

Context switching ⇒

थोड़ा - थोड़ा time देकर सबका काम करवाना

physics  
2 min

Math  
3 min

Chem  
4 min



AI DUAL CAMERA  
Shot by Nidhu

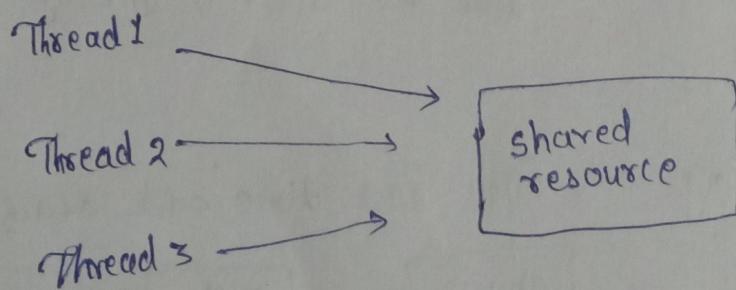
2023/06/10 04:24

Multiprogramming या multitasking  $\rightarrow$  <sup>Scme</sup> एक time पर दो या  
दो ये ज्यादा काम करवाना।

Memory - internal  $\rightarrow$  in cmp RAM

External  $\rightarrow$  in cmp Hard Disk

~~Program~~ Context switching - मतलब आगे इसके पास एक से  
ज्यादा candidate है किसी resource  
को use करने वाले, तो वो इस resource का new context  
switching की help से करता। Scheduling algorithm में होती है  
FCFS ... etc.



## Types of Synchronization -

- process synchronization (we study this in Operating system)
- Thread synchronization

Thread Synchronization - There are two types of thread synchronization.  
• mutual exclusive and • inter-thread communication.

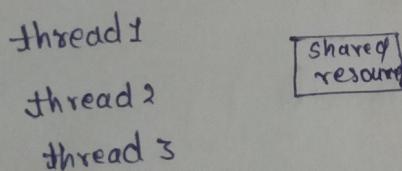
- Mutual exclusive - mutual exclusive helps keep threads from interfering with one another

while sharing data. It can be achieved by using the following three ways.

By using synchronized method

By using synchronized block

By using static synchronization



स्थिर thread तक use किना intrupdion के कर पाए इसलिए हम method को synchronized बना देंगे।

Program  $\Rightarrow$  फिलहाल हुआ code - (not in running state)  $\rightarrow$  in HD  
(not active program)

Process  $\Rightarrow$  Active program - (in running state)  $\rightarrow$  in RAM  
Active program situated in RAM (memory) is Process.

Threads  $\Rightarrow$  Process के अंदर होते हैं thread, वे उन एक से प्रभावी।  
जैसे W.P. के अंदर बहुत सारे thread हैं (process)

जोड़े जाना जाना thread 3 अंदर activity की perform करते हैं।  
1 thread messaging, state updating, 2023/06/10 04:24



AI DUAL CAMERA  
Shot by Nidhu

Multithreading - 5 imp q.

## Synchronization

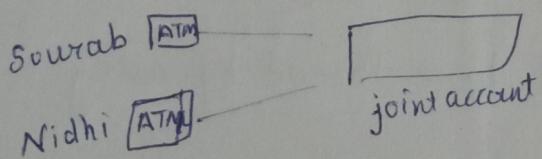
in Multithreading threads  $t_1$  and  $t_2$  are using Resource R1  
because of Multithreading  $t_1$  will use R1 for sometime  
then  $t_2$  will use R2 for sometime  
मिला single resource को at a same time, दो या दो ज्यादा  
thread use करे नहीं | लेटे interference होता |  
e.g. account balance, thread - Sourabh, Nidhi

# TAccountTestin.java

without synchronization

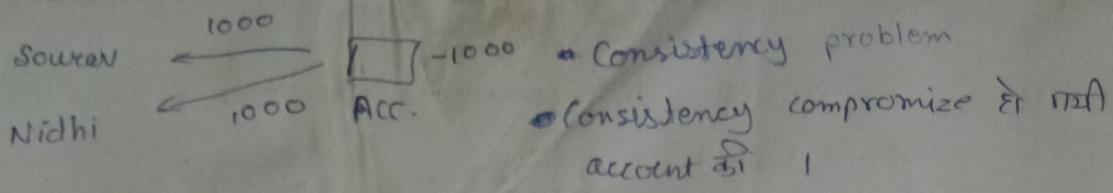
→ बिलकुल synchronization नहीं नहीं आए  
ई amount withdraw करता है,  
तो एक thread से withdraw होने के बाद thread 2 के withdraw ही  
होता amount without updation.  
thread  $t_1$  &  $t_2$  are using resources.

both thread will use this Resource R1 concurrently -



AI DUAL CAMERA  
Shot by Nidhi

2023/06/10 04:26



अगर synchronization नहीं होता है तो यह के बीच के चलने के बाद directly यह घटेगा, Account balance update ही नहीं हो पाएगा, वहाँ यहाँ यहाँ से यह amount withdraw हो जाएगा।

### # TAccountTestin.java

जबकि financial accounts, synchronized होते हैं, कभी-कभी दो transaction एक साथ नहीं होते।

आगे synchronized नहीं हुआ, तो जैसे normal thread चलते हैं concurrently, दोनों transaction साथ-साथ (concurrently)

हो जाएंगे, मतलब बहुत जल्दी context switching हो जाती है, जो amount है उसे update करने जिसना time भी नहीं रहता है।

और account balance negative हो जाएगा।

in our program ⇒ # TAccountTestin.java

- 1 we create a class which implements Runnable
- inside this class we have made a object of Account class which is our second class in program.

- after it from line 6 - 9 we have created two

(we have created here thread by using

non static method  
ये हर direct  
non static method  
को call कर सकते हैं।

by creating objects of TAccount class which implements Runnable ~~class~~ interface , For making a thread we need to call start() method , but in runnable interface there is only run() method ,

so we create object of thread class then pass our normal variable in those object , So that we will be able to call start() method

- TAccountTestin r = new TAccountTestin();
- Thread one = new Thread(r);
- Thread two = new Thread(r);

after it we set the names of our threads

```
one.setName ("Ranjeet");
two.setName ("Aeema");
```

after it we have override our run() method

inside run() method → we use for loop so that we can withdraw 5 times

and we called a method makeWithdrawal(10);

# ~~we have directly call this method without creating object means it is a~~ this method belongs from same class ,

non static method से direct call करना है तो non static होती है तो static method का call करते (e.g. withdraw)

static → all the static methods  
(e.g. main)

static → non static  
need to create object

Now see the definition of makeWithdraw() method

```
private void makeWithdraw (int amt)
```

```
{ }
```

inside its def. we have used getBalance() method

! Now where is getBalance(), so this is the method of our second class Account which returns balance (Account Balance)

makeWithdraw method में if (balance >= amt)

withdraw होगा । और किसी acet. withdraw (amt);

इस method में amount balance update होगा।

अगर नहीं है तो simply else block

चलेगा और S.U.P (not enough in account for....)

अब इसने makeWithdraw method को तो synchronised कर्दी गया है, और दो thread ने एक से बाहर एक (concurrently) use किया इस method को, तो amount update होने वाली method run ही नहीं हो पाएगी, और account में amount withdrawal हो जाएगा । balance negative में चला जाएगा ॥

# Observe the output of # TAccountTestin.java

this problem is called consistency problem.

Now for this problem, we use synchronization

synchronized void makeWithdrawal( )

इस method को अगर किसी तरह से use करते हैं तो पार्श्व तक वह  
पुरी तरह से use नहीं कर सकता, तब तक कोई और thread  
नहीं आ पाएगा।

एक time पर ज़िनी thread को bound होता है fun.  
method

C:\Users\Shubhams-PC\Desktop\javafolder\TAccountTestin.java - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

AWTmouseeg.java | Tjoinmthd1.java x | Tjoinmthd.java . | TAccountTestin.java . | AWTpanel2.java x | AWT

```
1 class TAccountTestin implements Runnable
2 {
3     private Account acct = new Account();
4     public static void main(String[] args)
5     {
6         TAccountTestin r = new TAccountTestin();
7         Thread one = new Thread(r);
8         Thread two = new Thread(r);
9         one.setName("Ranjeet");
10        two.setName("Reema");
11        one.start();
12        two.start();
13    }
14    @Override
15    public void run()
16    { for (int x = 0; x < 5; x++)
17    {
18        makeWithdrawal(10);
19        if (acct.getBalance() < 0)
20        { System.out.println("account is overdrawn!");}
21    }
22    }
23    private void makeWithdrawal(int amt)
24    //private synchronized void makeWithdrawal(int amt)
25    {
26        if (acct.getBalance() >= amt)
27        {
28            System.out.println(Thread.currentThread().getName() + " is going to withdraw");
29            try
30            { Thread.sleep(100);
31            }
32        }
33    }

```

Line 1, Column 7

Type here to search



AWTmouseeg.java

Tjoinmthd1.java

Tjoinmthd.java

TAccountTestin.java

AWTpanieg2.java

AWTfbpage.java

Trdeg1.java

Trdeg2.java

TMultithreading.java

```
30
31     try
32     {
33         Thread.sleep(100);
34
35         catch (InterruptedException ex)
36         {
37
38             acct.withdraw(amt);
39             System.out.println(Thread.currentThread().getName() + " completes the withdrawal");
40         }
41     }
42
43     else
44     {
45         System.out.println("Not enough in account for " + Thread.currentThread().getName() + " to withdraw " + acct.getBalance());
46     }
47 }
48 }
49 }
50
51 class Account
52 {
53     private int balance = 50;
54     public int getBalance()
55     {
56         return balance;
57     }
58     public void withdraw(int amount)
59     {
60         balance = balance - amount;
61     }
62 }
```

Line 1, Column 7



Type here to search

Activate Windows  
Go to Settings to activate

Species: 4



withdraw होगा । और ~~पर~~ acct.withdraw (amt);

→ इस method से amount balance update होंगा।

अगर नहीं है तो simply else block

चलेगा और S.U.P (not enough in account for .....

अब हमने makeWithdraw method को तो synchronised नहीं  
बनाया है, और हो thread ने एक से बाद एक (concurrently)  
use किया इस method को, तो amount update होने वाली method  
run ही नहीं हो पाएगी, और account से amount withdrawal की  
जाएगी। balance negative में चला जाएगा।

# Observe the output of # TAccountTestin.java

```
3     private void makeWithdrawal(int amt)
4     //priv  C:\ Command Prompt
5     {
6         if (C:\Users\Shubhams-PC\Desktop\javafolder>java Tjoinmthd
7             The current thread name is : main
8             { The current thread name is : main
9                 Sys
10                C:\Users\Shubhams-PC\Desktop\javafolder>javac TAccountTestin.java
11
12                C:\Users\Shubhams-PC\Desktop\javafolder>java TAccountTestin
13                {Ranjeet is going to withdraw
14                  Reema is going to withdraw
15                  Ranjeet completes the withdrawal
16                  Ranjeet is going to withdraw
17                  Ranjeet completes the withdrawal
18                  Reema completes the withdrawal
19                  Reema is going to withdraw
20                  Ranjeet is going to withdraw
21                  Ranjeet completes the withdrawal
22                  acct.Reema completes the withdrawal
23                  System.out.println("Not enough in account for Ranjeet to withdraw 0")
24                  System.out.println("Not enough in account for Reema to withdraw 0")
25                  System.out.println("Not enough in account for Ranjeet to withdraw 0")
26                  System.out.println("Not enough in account for Reema to withdraw 0")
27                  System.out.println("Not enough in account for Reema to withdraw 0")
28
29                  el
30                  { C:\Users\Shubhams-PC\Desktop\javafolder>
```



this problem is called consistency problem.

Now for this problem, we use synchronization

synchronized void makWithdrawal( )

इस method को आप तो use कर सकते हैं तो पर्वत का बहुत  
पुरी तरह से use नहीं कर लेंगा, तब तक कोई और thread  
नहीं आ पाएगा।

एक time पर एकी thread को bound होता है fun.  
method

```
//private void makeWithdrawal(int amt)
4  private synchronized void makeWithdrawal(int amt)
5  {
6      if (acct.getBalance() < amt)
7      {
8          System.out.println("Not enough in account for Ranjeet to withdraw 0");
9          System.out.println("Not enough in account for Reema to withdraw 0");
10         System.out.println("Not enough in account for Reema to withdraw 0");
11     }
12     try
13     {
14         Thread.sleep(100);
15     }
16     catch (InterruptedException e)
17     {
18         e.printStackTrace();
19     }
20     acct.withdraw(amt);
21     System.out.println("Ranjeet completes the withdrawal");
22     System.out.println("Ranjeet is going to withdraw");
23     System.out.println("Reema is going to withdraw");
24     System.out.println("Reema completes the withdrawal");
25     System.out.println("Ranjeet is going to withdraw");
26     System.out.println("Ranjeet completes the withdrawal");
27     System.out.println("Not enough in account for Ranjeet to withdraw 0");
28     System.out.println("Not enough in account for Reema to withdraw 0");
29     System.out.println("Not enough in account for Reema to withdraw 0");
30     System.out.println("Not enough in account for Reema to withdraw 0");
31 }
32 }
33 }
34 }
35 }
36 }
37 }
38 }
39 }
40 }
41 }
42 }
43 }
44 }
45 }
46 }
47 }
48 }
49 }
```

Command Prompt



no. of threads are more than no. of resources.

3 threads → resource 2 or less than 2

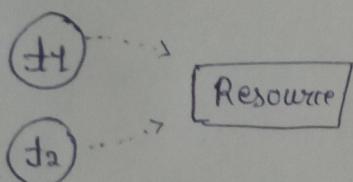
5 threads → resource 4 or less than 4

In multithreading more than one threads runs, and they want to complete their execution (means use of resource) at same time and complete work.

thread → want to use resource at same time

In this situation, race condition occurs.

Dead lock की शुरूआत race condition से होती है।



- \* Multithreading कोडी-योडी हो दोनों thread use करें।
- \* By synchronization पहले एक thread पुरी तरह से use कर लें। उसके बाद दूसरा thread पुरी तरह से use कर लें।

Multithreading is very useful.

But in some special condition, multithreading generates

mislead output \* for this we use synchronization method

H.W.

# TSeg.java

our thread class में बनवा करे class  
प्रैग्मेसी एक thread बनाएंगे।

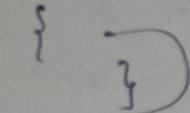
#. STThread.java

In Synchronisation

- resource class
- thread class
- test class

In our thread class

class STThread extends Thread



\* main fun thread class से बनाया  
तो resource & thread class  
ये भी आगामी ऐ explain हो  
सकते हैं

- we have created
1. constructor parameterised
  2. override run method
  3. main method

3. inside our main method, we created two threads and start them. and before this we created a object of Resource class (Table) → why? Table obj = new Table();

so that we can call resource class's methods by using the resource class object.

But this object is present only in main fun. if we want to use this in our thread class. so for this we pass this objects in our thread class's object.

STThread t1 = new STThread (obj);  
STThread t2 = new STThread (obj);

(10);  
(20);  
↳ these are just parameter names.

Now for catching this parameters we have created

1. Parameterised constructor

STThread (Table t, int a)  
{ }

→ inside this we use this pointer and make these local variables as instance, or we give value at class field now these are available at class level

2023/06/10 04:43

R

AI DUAL CAMERA

Shot by Nidhi

C:\Users\Shubhams-PC\Desktop\javafolder\STThread1.java - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

AWTmouseeg.java | Tjoinmthd1.java | Tjoinmthd.java | TAccountTestin.java | STThread1.java | AWTpanieg2.java | AW

```
1 class Table //Resource class
2 { void printTable(int n) // synchronized method
3 { System.out.println("hello");
4     for(int i = 1; i <= 10; i++)
5     { System.out.println(n * i);
6         try
7         { Thread.sleep(500); }
8         catch(InterruptedException e)
9         {System.out.println(e);}
10    }
11 }
12 class STThread1 extends Thread //Thread class
13 { Table t; // Declaring t as class type table
14     int a;
15 // Declaring parameterized constructor and passing
16 // variable t as a parameter to the thread.
17     STThread1(Table t,int a)
18     { this.t = t;
19         this.a=a;
20     }
21     public void run()
22     { t.printTable(a); } //calling resource method
23
24 public static void main(String[] args)
25 {
26 // Creating an object of Table class.
27     Table obj = new Table();
28     STThread1 t1 = new STThread1(obj,10);
29     STThread1 t2 = new STThread1(obj,20);
30
31     t1.start();
32     t2.start();
33 }}
```

Command Prompt

```
C:\Users\Shubhams-PC\Desktop\j
hello
20
hello
10
40
20
60
30
80
40
100
50
120
60
140
70
160
80
180
90
200
100
```

C:\Users\Shubhams-PC\Desktop\java

Line 16, Column 3



Type here to search



specially the reference of resource class.  
Tablet ; we got bcz of constructor.

now

2. we have overrided the run() method

insid this method we called a Resource class's method  
t.printTable(a);

so we use here a and t, that's why we create  
a Resource class object and make it available  
at class level for the by using parameterized  
constructor and this pointer.

In our Resource class

we simply made a resource method.

void printtable (int n)

inside this we write logic for printing table .

and we also have used sleep method  
inside try block following by catch block .

bcz of multithreading and we also have used sleep  
method here. output will overlap of thread1 and

thread2 . for this you can use synchronization here ,  
if we make our method synchronised .

then at one time one object will be associated  
with that method , firstly that object will completely use id  
then any other object will be able to use it .



H.W. #SThread2.java without synchronization

why we use multithreading

for multi task or chat control possib

```

AWTmouseeg.java | Tjoinmthd1.java | Tjoinmthd.java | TAccountTestin.java | STThread1.java | STThread2.java | AWTpanleg2.java | AWTf
1 class Table //Resource class
2 { void printTable(int n)
3 // synchronized method
4 {System.out.println("hello");
5 // synchronized () {
6 for(int i = 1; i <= 10; i++)
7 {
8 System.out.println( Thread.currentThread().getName() + " " +n * i);
9 try
10 { Thread.sleep(500); }
11 catch(InterruptedException e)
12 {System.out.println(e); } }}}
```

Command Prompt

```
C:\Users\Shubhams-PC>cd C:\Users\Shubhams-PC\Desktop\javafolder
C:\Users\Shubhams-PC\Desktop\javafolder>javac STThread2.java
C:\Users\Shubhams-PC\Desktop\javafolder>java STThread2
hello
hello
Thread-1 10
Thread-0 10
Thread-1 20
Thread-0 20
Thread-1 30
Thread-0 30
Thread-1 40
Thread-0 40
Thread-1 50
Thread-0 50
Thread-1 60
Thread-0 60
Thread-1 70
Thread-0 70
Thread-1 80
Thread-0 80
Thread-1 90
Thread-0 90
Thread-1 100
Thread-0 100
```

Line 17, Column 3



Type here to search

AI DUAL CAMERA  
Shot by Nidhu

2023/06/10 04:56

## ? Why we use multithreading

1. wp. से राय-वाय status लाना और chat करना possible है multithreading की वजह से है।
2. Account वाले e.g. में डबलिए प्रॉफ़िल account को access करने के diff. तरीके available होने चाहिए UPI, ATM, from bank.
3. table वाला e.g. इन normal fun बनाएंगी और 10 की table आएगी और 20 की table आएगी point करवाएंगी,  
→ तो it will behave like synchronised multithreading  
अगर multithreading है तो same time पर दोनों thread table print कर पाते हैं (Actual में same time पर नहीं होता )  
Context switching होती है, और पास भी नहीं चलता।  
# ऐसके योड़ी से problems के लिए multithreading को जानते होंगे लोग अक्षर multithreading के बहुत खोरे फोयदे हैं।

### E.g. of Multithread Context switching

you are on a call and with this you are cleaning your house or cooking food.

here mind focus →

if something is important in your conversation you will stop doing your cooking.

and if you get tricky work in cooking (like ~~कठोर~~ कठोर)

then you will stop focussing on your conversation.

synchronized block.

# STThreadB.java

मुझे method को एक synchronize करना पड़ते हैं।  
एक ओसके भिन्न लाइन को (block) को synchronize  
करना पड़ते हैं। तब synchronized block का use करें।

synchronized (this)

{ }

this → use - it points current object / thread.

this → current thread को bound करता है synchronized block  
में, ये sequence में जारी रहते हैं (st2, t1)  
(st1, st2)  
, main

उत्तर (this) नहीं लिया, तो

compile time error Illegal start of expression synchronized,

Output may vary → diff. diff.



AI DUAL CAMERA  
Shot by Nidhu

2023/06/10 04:57

File Edit Selection Find View Goto Tools Project Preferences Help

AWTmouseeg.java Tjoinmthd1.java Tjoinmthd.java TAccountTestin.java STThread1.java STThread2.java STThread8.java STThre...

```
class Table //Resource class
{
    void printTable(int n)
    // synchronized block
    {
        System.out.println("hello");
        synchronized (this) {
            for(int i = 1; i <= 10; i++)
            {
                System.out.println(n * i);
                try
                {
                    Thread.sleep(500);
                }
                catch(InterruptedException e)
                {
                    System.out.println(e);
                }
            }
        }
    }

    class STThreadB extends Thread //Thread class
    {
        Table t; // Declaring t as class type table
        int a;
        // Declaring parameterized constructor and passing
        //variable t as a parameter to the thread.
        STThreadB(Table t,int a)
        {
            this.t = t;
            this.a=a;
        }

        public void run()
        {
            t.printTable(a); //calling resource method
        }
    }

    public static void main(String[] args)
    {
        // Creating an object of Table class.
        Table obj = new Table();
        STThreadB t1 = new STThreadB(obj,10);
        STThreadB t2 = new STThreadB(obj,20);
        t1.start();
        t2.start();
    }
}
```

Line 19, Column 13

Command Prompt

C:\Users\Shubhams-PC\Desktop\javafolder>javac STTh...

C:\Users\Shubhams-PC\Desktop\javafolder>java STTh...

hello  
hello  
20  
40  
60  
80  
100  
120  
140  
160  
180  
200  
10  
20  
30  
40  
50  
60  
70  
80  
90  
100

C:\Users\Shubhams-PC\Desktop\javafolder>\_



AI DUAL CAMERA  
Shot by Nidhu



2023/06/10 05:04