EXP.NO:8	
DATE:16-09-19	MULTI THREADED APPLICATION

AIM:

To develop a java program for implementing multithread application.

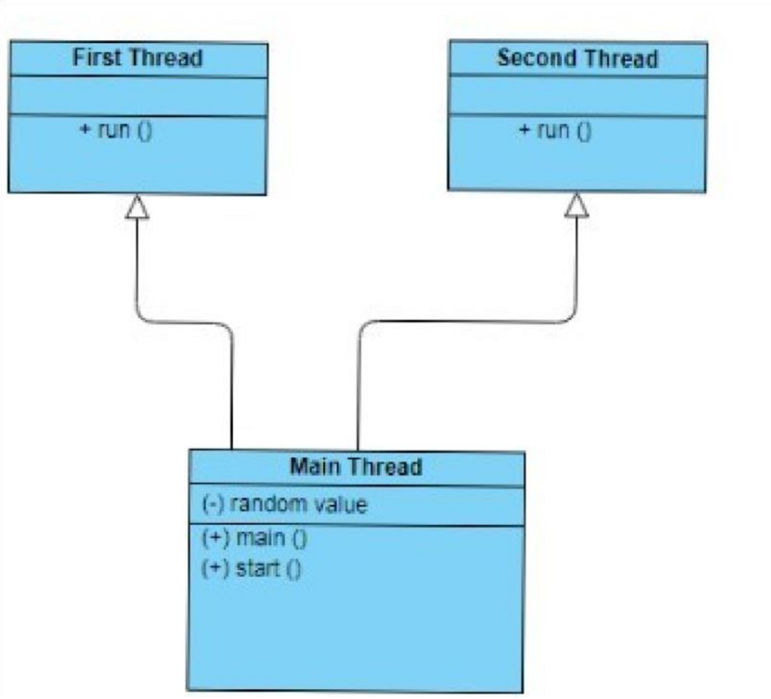
REQUIREMENTS:

Develop a java program that implements a multithread application that has 3 threads.First generates a random integer for every 1 second and if the value is even,second thread computes the square of the number and prints.If the value is odd,the third thread will print the value of cube of the number.

ALGORITHM:

STEP 1: Declare a package called multithread.
STEP 2: Declare a class name of FirstThread,SecondThread which extends from Thread.
STEP 3: Declare a object in the respective classes.
STEP 4: Create a condition to check the statements.
STEP 5: Print the result.

CLASS DIAGRAM:



PROGRAM:

```
//created by P.SAIRAM, eee-b,212217105044 MainThread:
package multithread; import
java.util.Random; public class
MainThread { public static Integer
RandomValue; public static void
main(String[] args) {
```

```
FirstThread t1;
SecondThread t2;
Random r;
t1=new FirstThread();t2=new SecondThread();
r=new Random(); RandomValue=-
1;
t1.start();
t2.start();
try {
while(true
) {
synchronized(RandomValue)
{
if(RandomValue==-1)
{
RandomValue=r.nextInt(200);
System.out.println("Placed a new number "+RandomValue);
}
}
Thread.sleep(4000);
}
}catch(InterruptedException ex)
{
System.out.println("Error:"+ex);
}
}
}
FirstThread: package
multithread;
public class FirstThread extends Thread { public
void run()
{
try
{
System.out.println("First thread started...");
while(true)
{
synchronized(MainThread.RandomValue)
{
if(MainThread.RandomValue%2==0&&MainThread.RandomValue!=-1)
{
System.out.println("Value is even");
System.out.println("Answer="(MainThread.RandomValue*MainThread.RandomValue));
MainThread.RandomValue=-1;
}
}
Thread.sleep(1000);
}
}catch(InterruptedException ex)
{
System.out.println("Error:"+ex);
}}
}
```

SecondThread:

```
package multithread;
public class SecondThread extends Thread {
public void run()
{
try
{
System.out.println("Second thread
started..."); while(true) {
synchronized(MainThread.RandomValue)
{ if(MainThread.RandomValue%2!
=0&&MainThread.RandomValue!=-1)
{
System.out.println("Value is odd");
System.out.println("Answer="+
(MainThread.RandomValue*MainThread.RandomValue*MainThread.RandomValue));
MainThread.RandomValue=-1;
}
}
Thread.sleep(1000);
}
}catch(InterruptedException ex)
{
System.out.println("Error:"+ex);
}
}
}
```

OUTPUT:

```
First thread started...
Second thread started...
Placed a new number 175
Value is odd
Answer=5359375
```

```
Placed a new number 73
Value is odd
Answer=389017
Placed a new number 162
value is even
Answer=26244
Placed a new number 95
Value is odd
Answer=857375
Placed a new number 68
value is even
Answer=4624
```

RESULT: Thus a java application that performs multithreading is developed.