EX.NO:8	MULTITHREADED APPLICATION
DATE:16-09-19	

AIM:

To develop a java program for implementing multithread application.

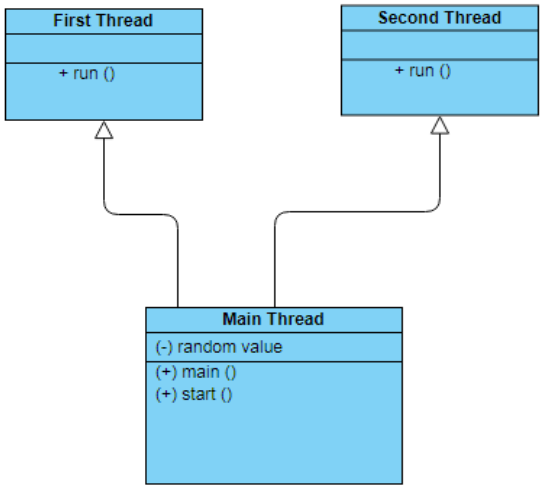
REQUIREMENTS:

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:

First Thread:

```
/*
* developed by Monica
* EEE-B
* 212217105039
*
*/
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(2000);
}
}catch(InterruptedException ex)
{
	System.out.println("Error:"+ex);
}

}
}
```

Second thread:

```
	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(2000);
}
}catch(InterruptedException ex)
{
	System.out.println("Error:"+ex);
```

```
}
}
}
```

Main thread:

```
	package multithread;

	import java.util.*;

	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(1000);
}
}catch(InterruptedException ex)
{
	System.out.println("Error:"+ex);
}
}

}
```

Output:

```
	First thread started...
	Placed a new number 35
	Second thread started...
	Value is odd
	Answer=42875
	Placed a new number 192
```

Value is even
Answer=36864
Placed a new number 187
Value is odd
Answer=6539203
Placed a new number 150
Value is even
Answer=22500
Placed a new number 168

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