```java
1   import java.util.Random;
2   class RandomNumberThread extends Thread {
3   public void run() {
4   Random random = new Random();
5   for (int i = 0; i < 10; i++) {
6   int randomInteger = random.nextInt(100);
7   System.out.println("Random Integer generated : " + randomInteger);
8   if((randomInteger%2) == 0) {
9   SquareThread sThread = new SquareThread(randomInteger);
10  sThread.start();
11  }
12  else {
13  CubeThread cThread = new CubeThread(randomInteger);
14  cThread.start();
15  }
16  try {
17  Thread.sleep(1000);
18  }
19  catch (InterruptedException ex){
20  System.out.println(ex);
21  }
22  }
23  }
24  }
25  class SquareThread extends Thread {
26  int number;
27  SquareThread(int randomNumbern) {
28  number = randomNumbern;
29  }
30  public void run() {
31  System.out.println("Square of " + number + " = " + (number * number));
32  }
33  }
34  class CubeThread extends Thread {
35  int number;
36  CubeThread(int randomNumber) {
37  number = randomNumber;
38  }
39  public void run() {
40  System.out.println("Cube of " + number + " = " + number * number * number);
```
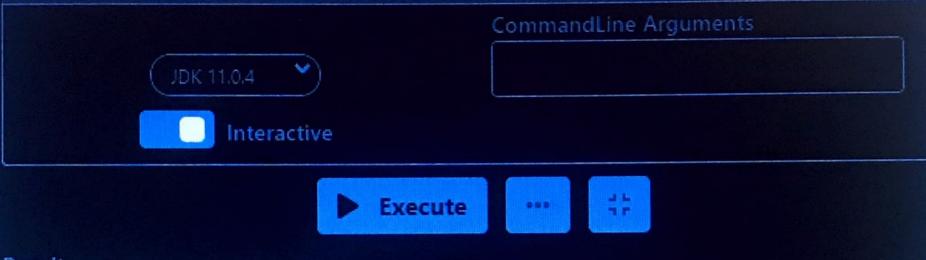
```java
CubeThread(int randomNumber) {

number = randomNumber;

}


public void run() {

System.out.println("Cube of " + number + " = " + number * number * number);

}

}


public class Mtest {

public static void main(String args[]) {

RandomNumberThread rnThread = new RandomNumberThread();

rnThread.start();

}

}
```

JDK 11.0.4

Interactive

▶ Execute   ...   ⌗

Result

compiled and executed in 11.008 sec(s)

```
Random Integer generated : 45
Cube of 45 = 91125
Random Integer generated : 73
Cube of 73 = 389017
Random Integer generated : 23
Cube of 23 = 12167
Random Integer generated : 59
Cube of 59 = 205379
Random Integer generated : 4
Square of 4 = 16
Random Integer generated : 18
Square of 18 = 324
Random Integer generated : 27
Cube of 27 = 19683
Random Integer generated : 75
Cube of 75 = 421875
Random Integer generated : 79
Cube of 79 = 493039
Random Integer generated : 5
Cube of 5 = 125
```

```java
 1  public class DemoEvenOdd {
 2    public static void main(String args[]) {
 3    B ob2 = new B();
 4    int sum=0;
 5    try {
 6    ob2.t.join();
 7    for(int i=2;i<=100;i +=2) {
 8    sum +=i;
 9    }
10    } catch (InterruptedException e) {
11    System.out.println("Main thread Interrupted");
12    }
13    System.out.println("Main thread exiting. Even Sum = " +sum); }
14    }
15    class B implements Runnable {
16    Thread t;
17    B() {
18    t = new Thread(this, "Demo Thread");
19    System.out.println("Start odd sumation .");
20    t.start();
21    }
22    public void run() {
23    int sum=0;
24    try {
25    for(int i = 1; i <=100; i+=2) {
26    sum +=i;
27    Thread.sleep(10);
28    }
29    } catch (InterruptedException e) {
30    System.out.println("B interrupted.");
31    }
32    System.out.println("Exiting Odd thread. Sum ="+sum);
33    }
34    }
```

CommandLine Arguments

JDK 11.0.4

Interactive

▶ Execute   •••   ⌣⌃

Result

compiled and executed in 1.591 sec(s)

Start odd sumation .
Exiting Odd thread. Sum =2500
Main thread exiting. Even Sum = 2550