

Name: Raghav Makashwari
Roll No: 53
Panel: A

Date: 5/05/22

Lab Assignment - 7 (JP)

Aim:

Write a Java program for Multithreading using Thread Class and Runnable Interface.

Objective:

Apply Multithreading in Java Application.

Theory:

Multitasking, Multiprocessing, Multithreading

Multitasking: It refers to execution of multiple processes or task at same time on a single device. In multitasking it works on principles of time sharing and context switching.

Multiprocessing: In a ~~uni-processor~~ system only one process executes at a time. The term also refers to ability of a system to support more than one ~~processors~~ / CPU's i.e. multi processes are executed simultaneously.

Multithreading:

A thread is a basic unit of CPU utilization, Multithreading

is an execution model that allows a single process to have multiple code segments running concurrently within "context" of that process.

Creating of thread using Thread class and implementation of Runnable Interfaces.

Thread class is to be extended by a Subclass. Then ~~Then~~ run() function should be overridden for execution of thread. Then object of subclass should be created and start() function is called for thread execution.

Runnable Interface is implemented by a class which has to implement run() function in its body. Object of this class is passed to thread class and then that object is used for execution of thread.

Method belongs to thread class.

setName(), getName(), getPriority(), isAlive(), isJoin(), run(), sleep(), start(), activeCount(), checkAccess(), Thread dumpStack(), getID(), getState(), getThreadGroup(), interrupt(), yield().

Platform:

Open source Java Programming tool like Eclipse Editor/Netbeans.

Conclusion:

Thus studied multithreading concept in Java.

* FAQ

Q1 Why do we use Multithreading in Java?

Ans Advantages are:

→ It does not block user because threads are independent and you can perform multiple operations at same time.

→ Many operation can be performed together.

→ Threads are independent hence exception in a single thread does not stop program.

Q2 What is Multithreading vs Multiprocessing?

Ans 1. Multiprocessing uses two or more CPU's to increase computing power. Multithreading uses a single process with multiple code segments to increase power.

2. Many operation can be performed together.

3. Threads are independent hence exception in a single thread does not stop program.

4. Multiprocesses increase computing power by adding CPU's. Multithreading focus on generating computing threads from a single process.

Q3 What is thread in Java?

Ans A thread is a thread of execution in a program. The java threads allow faster execution. The JVM creates and allows multithread execution.

Q4 What are two ways of implementing thread in Java?

Ans We can extend a thread class or implement a Runnable for implementing thread in Java.

Q5 What's the difference thread and process?

Ans A process is program under execution whereas a thread is a lightweight process that can be managed by scheduler. A process takes high context switching time; a thread takes less context switch time. Process hold their separate storage area but threads share memory.

Seen

CODE:

```
class Even extends Thread
{
    public void run()
    {
        for(int i=0;i<20;i=i+2)

            {
                System.out.println("Even: "+i);
            }
    }
}

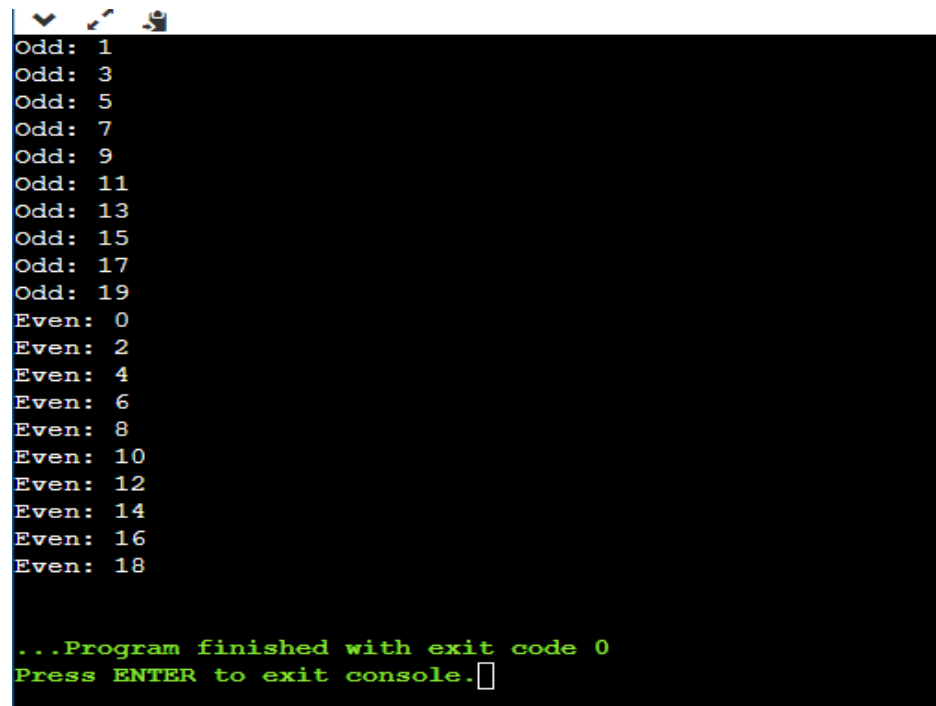
class Odd extends Thread
{
    public void run()
    {
        for(int i=1;i<20;i=i+2)
            {
                System.out.println("Odd: "+i);
            }
    }
}

public class Main {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }
}
```

```
        Odd o=new Odd();  
        Even e=new Even();  
        o.start();  
        e.start();  
    }
```

Output:



```
Odd: 1  
Odd: 3  
Odd: 5  
Odd: 7  
Odd: 9  
Odd: 11  
Odd: 13  
Odd: 15  
Odd: 17  
Odd: 19  
Even: 0  
Even: 2  
Even: 4  
Even: 6  
Even: 8  
Even: 10  
Even: 12  
Even: 14  
Even: 16  
Even: 18  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```



Dr. Vishwanath Karad

**MIT WORLD PEACE
UNIVERSITY** | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

