

Practical-5

Name: Sojal Jyoti Kadam

Roll No.:CO2058

Title: Create a multi-threaded Java application that simulates any real time application with required functionalities. For eg. Basic chat system in which each user (thread) sends and receives messages. Use `isAlive()` to check the status of threads and `join()` to ensure proper synchronization. Implement thread priorities to handle high-priority messages and demonstrate thread suspension, resumption, and stopping.

INPUT:-

```
public class TwoUserChatDemo
{
    public static void main(String[] args) throws Exception
    {
        ChatUser u1=new ChatUser("Rishi");
        ChatUser u2=new ChatUser("Rashi");

        u1.setPriority(Thread.MAX_PRIORITY);
        u2.setPriority(Thread.MIN_PRIORITY);

        System.out.println("Chatting between Rishi and Rashi");

        u1.start();
        u2.start();

        //check if alive
        System.out.println("Is Rashi alive?" +u2.isAlive());

        //pause rishi for some time
        Thread.sleep(1000);
        u1.pauseChat();
        System.out.println("Rishi is paused");

        Thread.sleep(1000);
        u1.resumeChat();
        System.out.println("Rishi is resumed");

        //stop rashi early
        Thread.sleep(1000);
        u1.stopChat();
        System.out.println("Rashi stopped");

        //wait for both to finish
        u1.join();
        u2.join();

        System.out.println("Is Rashi alive after join?" +u2.isAlive());
        System.out.println("Chat ended!!");
    }
}
```

```
}
```

```
class ChatUser extends Thread
```

```
{
```

```
    private volatile boolean paused=false;
```

```
    private volatile boolean running=true;
```

```
    ChatUser(String name)
```

```
    {
```

```
        super(name);
```

```
    }
```

```
    public void run()
```

```
    {
```

```
        int i =1;
```

```
        while(running&& i<=5)
```

```
        {
```

```
            if(paused==false)
```

```
            {
```

```
                System.out.println(getName()+" is texting Message "+i);
```

```
                i++;
```

```
                try
```

```
                {
```

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

```
                }
```

```
                catch(Exception e)
```

```
                {
```

```
                }
```

```
            }
```

```
        }
```

```
        System.out.println(getName()+"finished chatting");
```

```
    }
```

```
    //pause/resume/stop methods
```

```
    public void pauseChat()
```

```
    {
```

```
        paused=true;
```

```
    }
```

```
    public void resumeChat()
```

```
    {
```

```
        paused=false;
```

```
    }
```

```
    public void stopChat()
```

```
    {
```

```
        running=false;
```

```
    }
```

```
}
```

OUTPUT:

```
Chatting between Rishi and Rashi
Is Rashi alive?true
Rishi is texting Message 1
Rashi is texting Message 1
Rishi is texting Message 2
Rashi is texting Message 2
Rishi is paused
Rashi is texting Message 3
Rashi is texting Message 4
Rishi is resumed
Rishi is texting Message 3
Rashi is texting Message 5
Rishi is texting Message 4
Rashi finished chatting
Rashi stopped
Rishi finished chatting
Is Rashi alive after join? false
Chat ended!!
```