13-2-24

Lab-9

Demonstrate Interthread communication and deadlock

```
class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while (!valueSet)
        try {
            System.out.println("In consumer");
            wait();
        }
        catch (InterruptedException e)
        {
            System.out.println("Interrupted exception");
        }
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("Intimate Producer");
        notify();
        return n;
    }
    synchronized void put(int n) {
        while (valueSet)
        try {
            System.out.println(
                "Producer waiting");
            wait();
        }
        catch (InterruptedException e)
        {
        }
    }
}
```

```
Public void run (){
    int i = 0;
    while (i < 15);
        int r = q.get();
    }
    System.out.println ("consumed" + i);
    i++;
  }
}
class PCFixed{
  Public static void main (String args[])
  {
    Q q = new Q;
    new Producer (q);
    new Consumer (q);
  }
}
```

output

Put:0
Intimate consumer
Producer waiting
got: 0
Intimate producer
Consumed waiting
Put: 1
Intimate consumer
Producer waiting
got: 1
Intimate Producer
consumed 1

Put: 2
Intimate Consume
Producer waiting
Got: 2
Intimate consumer
consumed: 2
Put 3
Intimate consu
Produce waiting
got: 3
Intimate consume
Consumed: 3
Put: 4

SURYA Gold
Date Page

Deadlocking

```
class A{
    Synerouised void foo(B b){
    String name = thread.(urrent thread().getName();
    System.out.println(name + " enter A.foo);
    try{
        Thread.Sleep(1000);
    }
    catch(exception e){
      System.out.println(" A Interupted";
    }
    System.at.println(" Interrupted) trying to call
            B.last() " );
      b.last();
    }
    void last{
      System.out.println(" inside A.last");
    }
}
class B{
    Synerouized void bar(A a){
    String name = thread.current thread(); getName
    System.out.println(Name + "entered B.bar
    try{
        thread.steep(1000);
    }
    catch (Exception E)
    {
      System.out.println("B Interupted")
    }
    System.out.println(name + "trying to call
            A.last());
    a.last
    }
```

```
void last C) {
    System.out.println("Inside A last");
}
}
class Deadlock implements runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName
            ("main thread");
        Thread t = new Thread(this,
            "racing thread");
        t.start();
        a.foo(b);
        System.out.println("back in
            main thread");
    }

Public void run() {
    b.bar(a);
    system.out.println("Back in otherthread");
}
Public static void main(String args[])
    new Deadlock();
}
```

output
Mainthread entered A.foo
Racingthread entered B.bar
racing thread to call A.last()
Inside A.last
Back in otherthread

SURYA Gold
Date Page

Main thread trying to call B.last()
Inside A.last
Deadlock in main thread.

Shree Sudharsan K

12.08.24