**PROGRAM**

**Dining.java:-**

import com.mongodb.\*;

import java.util.concurrent.locks.Lock;

import java.util.concurrent.locks.ReentrantLock;

public class Dining {

public static void main(String[] args) {

Lock forks[] = new ReentrantLock[5];

try {

MongoClient mongoClient = new MongoClient("localhost");

System.out.println("Connection to mongodb successful.");

DB db = mongoClient.getDB( "mydb" );

System.out.println("Database 'mydb' created.");

DBCollection coll = db.createCollection("mycol", null);

System.out.println("Collection 'mycol' created.");

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} for(int i = 0; i<5; i++){

forks[i] = new ReentrantLock();

} Thread p1 = new Thread(new Philosopher(forks[4], forks[0], "first"));

Thread p2 = new Thread(new Philosopher(forks[0], forks[1], "second"));

Thread p3 = new Thread(new Philosopher(forks[1], forks[2], "third"));

Thread p4 = new Thread(new Philosopher(forks[2], forks[3], "fourth"));

Thread p5 = new Thread(new Philosopher(forks[3], forks[4], "fifth"));

p1.start();

p2.start();

p3.start();

p4.start();

p5.start();

}

} class Philosopher implements Runnable {

Lock leftFork = new ReentrantLock();

Lock rightFork = new ReentrantLock();

String name;

public Philosopher(Lock leftFork, Lock rightFork, String name) {

this.leftFork = leftFork;

this.rightFork = rightFork;

this.name = name;

} @Override

public void run() {

try {

think(name);

eat(leftFork, rightFork, name);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

} private void eat(Lock leftFork, Lock rightFork, String name) throws Exception{

leftFork.lock();

rightFork.lock();

try

{ MongoClient mongoClient = new MongoClient("localhost");

DB db = mongoClient.getDB( "mydb" );

DBCollection coll = db.getCollection("mycol");

System.out.println(name + " eating...");

BasicDBObject doc1 = new BasicDBObject(name , " eating...");

coll.insert(doc1);

Thread.sleep(1000);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

System.out.println(name + " done eating and now thinking...");

MongoClient mongoClient = new MongoClient("localhost");

DB db = mongoClient.getDB( "mydb" );

DBCollection coll = db.getCollection("mycol");

BasicDBObject doc2 = new BasicDBObject(name , " done eating and now

thinking...");

coll.insert(doc2);

leftFork.unlock();

rightFork.unlock();

}

} public void think(String name) throws Exception{

try

{ MongoClient mongoClient = new MongoClient("localhost");

DB db = mongoClient.getDB( "mydb" );

DBCollection coll = db.getCollection("mycol");

System.out.println(name + " thinking...");

BasicDBObject doc = new BasicDBObject(name , " thinking...");

coll.insert(doc);

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

} } }

**OUTPUT**



