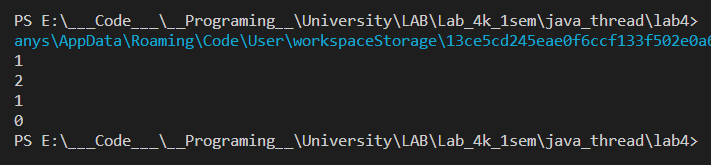
В цій лабораторній я продемонстрував знання використання синхронізацї за допомогою обекту ReentrantLock:



import java.util.concurrent.TimeUnit;

import java.util.concurrent.locks.Lock;

import java.util.concurrent.locks.ReentrantLock;

public class Main {

    static MyThread thread1;

    static MyThread thread2;

    public static void main(String[] args) throws InterruptedException {

        Counter counter = new Counter();

        new MyThread(counter).start();

        new MyThread(counter).start();

    }

}

class MyThread extends Thread {

    Counter counter = null;

    public MyThread(Counter param) {

        this.counter = param;

    }

    @Override

    public void run() {

        try {

            counter.increment();

            System.out.println(counter.value());

            Thread.sleep(500);

            counter.decrement();

            System.out.println(counter.value());

        } catch (InterruptedException e) {

        }

    }

}

class Counter {

    private int c = 0;

    Lock lock = new ReentrantLock();

    public void increment() throws InterruptedException {

        lock.tryLock(10, TimeUnit.SECONDS);

        int a;

        Thread.sleep(150);

        a = c;

        a++;

        c = a;

        lock.unlock();

    }

    public void decrement() throws InterruptedException {

        lock.tryLock(10, TimeUnit.SECONDS);

        int a;

        Thread.sleep(100);

        a = c;

        a--;

        c = a;

        lock.unlock();

    }

    public int value() {

        return c;

    }

}