package cloud\_matris;

import java.util.Random;

import java.util.concurrent.CountDownLatch; /// multithread kullanım

public class matris\_cloud {

    private int[][] birincimatris;

    private int[][] ikincimatris;

    private int[][] sonucmatris;

private dizim[] toplamthread;

    final CountDownLatch multithread;

public static final int toplam = 7980;

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

        new matris\_cloud ();

    }

  int threadler = (toplam / 4);

public void run() {

            for (int k = durumumuz; k < (durumumuz + threadler); k++) {

                for (int i = 0; i < toplam; i++) {

                    for (int j = 0; j < toplam; j++) {

                       sonucmatris [k][i] += birincimatris [k][j] \* ikincimatris [j][i];

                    }} }

multithread.countDown();

        }

    public matris\_cloud() throws InterruptedException {

        Random sayi\_ata = new Random();

       birincimatris= new int[toplam][ toplam];

       ikincimatris = new int[toplam][ toplam];

       sonucmatris = new int[toplam][ toplam];

        int atanan = sayi\_ata.nextInt(1000) + 1;

        for (int i = 0; i < toplam; i++) {

            for (int j = 0; j < toplam; j++) {

               birincimatris [i][j] = atanan;

               atanan = sayi\_ata.nextInt(1000) + 1;

ikincimatris [i][j] = atanan;

atanan = sayi\_ata.nextInt(1000) + 1;

            }

        }

toplamthread = new dizim[4];

int baslanan = 0;

       multithread = new CountDownLatch(4);

        for (int i = 0; i < 4; i++) {

toplamthread [i] = new dizim (baslanan);

         toplamthread [i].start();

           baslanan += threadler

        }

        long baslamazamani = System.nanoTime();

        long zamansonuc = 0;

        long bitis = System.nanoTime();

multithread.await();

       zamansonuc = bitis - baslamazamani;

        System.out.println(zamansonuc);

    }

    private class dizim extends Thread {

        int durum;

       dizim (int durum) {

            this.durum = durum;

        }

   }

}