L7

Ushtrimi 1.

Ndertoni nje program, ne java e cila te afishoje emrin e thread-it duke ju dhene si emer nje numur (psh threade 1, thread 2 etj)

```
public class ThreadExample {

        public static void main(String[] args) {
                System.out.println(Thread.currentThread().getName());
           for(int i=0; i<10; i++){
             new Thread("" + i){
               public void run(){
                 System.out.println("Thread: " + getName() + " running");
               }
             }.start();
           }
        }
}
```

Ushtrim 2.

```
 class Numero extends Thread
{
  Numero()
  {
   super("Thread");
   System.out.println("krijimi i thread-eve tona" + this);
   start();
  }
  public void run()
  {
   try
   {
     for (int i=0 ;i<10;i++)
     {
       System.out.println("Printo numerimet " + i);
      // thread to sleep for 1000 milliseconds
       Thread.sleep(1000);
     }
   }
   catch(InterruptedException e)
   {
```

```
        System.out.println("my thread interrupted");
      }
      System.out.println("My thread run is over" );
    }
}
 public class ThreadExample
{
   public static void main(String args[])
   {
      Numero nr = new Numero();
      try
      {
        while(nr.isAlive())
        {
          System.out.println("Thread do te aktivizohet derisa thread-et e tjera 'femije' do te behen
aktive");
          Thread.sleep(1500);
        }
      }
      catch(InterruptedException e)
      {
        System.out.println("Main thread interrupted");
      }
      System.out.println("Main thread's run is over" );
   }
}
```

Ushtrimi 3.

Thread creation by implementing Runnable Interface

```
class ThreadExample implements Runnable{
  public void run(){
    System.out.println("My thread is in running state.");
  }
 // public class ThreadExample{
  public static void main(String args[]){
     ThreadExample obj=new ThreadExample();
     Thread tobj =new Thread(obj);
     tobj.start();
 }
}
```

Ushtrim 4.

```
class Example implements Runnable {

String name;
Thread t;
   Example (String thread){

   name = thread;
   t = new Thread(this, name);

System.out.println("New thread: " + t);

t.start();

}

public void run() {

 try {
  for(int i = 10; i>0;i--) {

     System.out.println(name + ": " + i);

      Thread.sleep(1000);
}
}catch (InterruptedException e) {
    System.out.println(name + "Interrupted");
}
    System.out.println(name + " exiting.");
}
}

class ThreadExample {

public static void main (String args[])  {

    new Example("One");
    new Example("Two");
    new Example("Three");
try {
    Thread.sleep(10000);
```

```
} catch (InterruptedException e) {

    System.out.println("Main thread Interrupted");

}
    System.out.println("Main thread exiting.");
    }
}
```

Ushtrimi 4.

```
/** * Simple Java program to demonstrate how to use multiple threads. * Steps to use * multiple threads in Java : * 1. Implement Runnable interface to put the code * you want to run in separate thread. * 2. Create an Instance of Thread class by * passing an instance of Runnable you just crea ted. * 3. Call Thread.start() * method, this will execute the code in separate thread. * * @autho r WINDOWS 8 * */ public class MultipleThreadDemo { private static class ParallelTask implem ents Runnable { @Override public void run() { System.out.println(Thread.currentThread().getNa me() + " is executing this code"); } } public static void main(String[] args) { // created three threa ds but none will start until you call // start() method Thread t1 = new Thread(new ParallelTask() , "Thread - T1"); Thread t2 = new Thread(new ParallelTask(), "Thread - T2"); Thread t3 = new T hread(new ParallelTask(), "Thread - T3"); // now, let's start all three threads t1.start(); t2.start(); t 3.start(); } }
```

Ushtrimi 5.

The following Java application shows how the transactions in a bank can be carried out concurre ntly. */

```
class Account {
        public int balanca;
        public int NrLlogarise;
        void displayBalanca() {
                System.out.println("Numuri i llogarise:  " + NrLlogarise + "    Balanca/gjendja ne
llogari: " + balanca);
        }

          synchronized void deposit(int shuma){
                        balanca = balanca + shuma;
                        System.out.println( shuma + "    eshte depozituar");
```

```
                displayBalanca();
        }

        synchronized void terheqje(int shuma){
                balanca = balanca - shuma;
                System.out.println( shuma + "   vlera qe terhiqe nga llogaria");
                displayBalanca();
        }
}

class DepoziataTransaksionit implements Runnable{
        int shuma;
        Account accountX;
        DepoziataTransaksionit(Account x, int shuma){
                accountX = x;
                this.shuma = shuma;
                new Thread(this).start();
        }

        public void run(){
                accountX.deposit(shuma);
        }
}

class TransaksionTerheqje implements Runnable{
        int shuma;
        Account accountY;

        TransaksionTerheqje(Account y, int shuma) {
                accountY = y;
                this.shuma = shuma;
                new Thread(this).start();
        }

        public void run(){
                accountY.terheqje(shuma);
        }
}
class Balanca{
        public static void main(String args[]) {
                Account ABC = new Account();
                ABC.balanca = 1000;
                ABC.NrLlogarise = 111;
                DepoziataTransaksionit t1;
                TransaksionTerheqje t2;
                t1 = new DepoziataTransaksionit(ABC, 500);
```

```
                t2 = new TransaksionTerheqje(ABC,900);
        }
}
```
https://cse.iitkgp.ac.in/~dsamanta/java/ch6.htm

Ushtrimi 6.

```
import java.util.Random;

public class Ushtr {

//Krijojme matricen
static int[][] mat = new int[3][3];
static int[][] mat2 = new int[3][3];
static int[][] rezultati = new int[3][3];

public static void main(String [] args){

   //Krijojme nje objekt per klasen Random
   Random rand = new Random();


   //Mbushim matricen me vlera random
   for (int i = 0; i < mat.length; i++) {
      for (int j = 0; j < mat[i].length; j++) {
         mat[i][j]=rand.nextInt(10);
      }
   }

   //Mbushim matricen e dyte serisht me vlera te rastit
   for (int i = 0; i < mat2.length; i++) {
      for (int j = 0; j < mat2[i].length; j++) {
         mat2[i][j]=rand.nextInt(10);
      }
   }

   try{
      //Objekti i klases Multiply
     Shumfishim shumfishim = new Shumfishim(3,3);
//Threads
      MatricaShumfishim thread1 = new MatricaShumfishim(shumfishim);
      MatricaShumfishim thread2 = new MatricaShumfishim(shumfishim);
      MatricaShumfishim thread3 = new MatricaShumfishim(shumfishim);

      //Implementing threads
```

```
        Thread th1 = new Thread(thread1);
        Thread th2 = new Thread(thread2);
        Thread th3 = new Thread(thread3);

        //Starting threads
        th1.start();
        th2.start();
        th3.start();

        th1.join();
        th2.join();
        th3.join();

    }catch (Exception e) {
        e.printStackTrace();
    }

    //Printing the result
    System.out.println("\n\nResult:");
    for (int i = 0; i < rezultati.length; i++) {
        for (int j = 0; j < rezultati[i].length; j++) {
            System.out.print(rezultati[i][j]+" ");
        }
        System.out.println();
    }
}
}//End Class

   //Multiply Class
   class Shumfishim extends Ushtr{

private int i;
private int j;
private int rasti;

public Shumfishim(int i, int j){
    this.i=i;
    this.j=j;
    rasti=0;
}

//Matrix Multiplication Function
public synchronized void matricaShumfishim(){

    int sum=0;
    int a=0;
```

```
    for(a=0;a<i;a++){
        sum=0;
        for(int b=0;b<j;b++){
            sum=sum+mat[rasti][b]*mat2[b][a];
        }
        rezultati[rasti][a]=sum;
    }

    if(rasti>=i)
        return;
    rasti++;
}
}//mbaron klasa e shumfishimit

//Klasa thread
    class MatricaShumfishim implements Runnable {

private final Shumfishim mul;

public MatricaShumfishim(Shumfishim mul){
    this.mul=mul;
}

@Override
public void run() {
    mul.matricaShumfishim();
}
    }
```