Network programming codes

Lecture 1:

11/11 lecture

threads

```
package javaapplication3;
import java.io.*;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
class T implements Runnable
{
String nof;
T(String name)
{
nof =name;
}
public void run() {
String line;
String res;
try {
BufferedReader b = new BufferedReader (new FileReader(nof));
while((line=b.readLine())!=null)
System.out.println(line);
}
catch(Exception e)
{
}
}
}
public class JavaApplication3 {
public static void main(String[] args)throws Exception
{
Thread t = new Thread(new T("D:\\T1.TXT"));
Thread t1 = new Thread(new T("D:\\T2.TXT"));
Thread t2= new Thread(new T("D:\\T3.TXT"));
t1.start();t2.start();t.start();
}
}
```

Lecture2:

```
package javaapplication2;
import java.io.*;
class Process extends Thread{
private int arr[];
private int sum;
boolean isFinished=false;
public Process(int a[]) {
arr=a;
}
public void run() {
for (int value:arr)
sum +=value;
isFinished=true;
}
int getSum(){
return sum;
}
}
public class JavaApplication2{
public static void main(String[] args) throws Exception{
int a[]=new int [10000];
int b[]=new int [10000];
int c[]=new int [10000];
for (int i=0;i<10000;i++)
{
a[i]=b[i]=c[i]=i;
/*a[i]=(int)(Math.random()*50);
b[i]=(int)(Math.random()*50);
c[i]=(int)(Math.random()*50);*/
}
Process t1 = new Process(a);
Process t2 = new Process(b);
Process t3 = new Process(c);
t1.start();t2.start();t3.start();
while (!t1.isFinished || !t2.isFinished || !t3.isFinished);
System.out.println("T1: "+t1.getSum());
System.out.println("T2: "+t2.getSum());
System.out.println("T3: "+t3.getSum());
}
}
```

Lecture3:

25/11 lecture

```
package javaapplication2;
import java.io.*;
class Create extends Thread{
String fname;
volatile boolean isFinished;
public Create(String fname) {
this.fname = fname;
}
public void run() {
try{
BufferedWriter bf = new BufferedWriter(new FileWriter(fname));
for (int i=0;i<10000;i++)
bf.write ( ((int)(Math.random()*50))+"\n" );
bf.close();
isFinished=true;
}catch(Exception e){
e.printStackTrace();
}
}
}
class ReadFile extends Thread{
String fname;
double sum;
volatile boolean isFinished;
JavaApplication2 app;
public ReadFile(String fname, JavaApplication2 o) {
this.fname = fname;
app=o;
}
```

```
public void run(){
String num;
try{
BufferedReader bf = new BufferedReader(new FileReader(fname));
while ((num=bf.readLine())!=null)
if (Integer.parseInt(num)%2==1)
sum +=Integer.parseInt(num);
bf.close();
isFinished=true;
JavaApplication2.getInfo(sum);
app.getInfoRef(sum,fname);
}catch(Exception e){
e.printStackTrace();
}
}
}
public class JavaApplication2{
static int c=0;
static double max;
int counter=0;
public static void main(String[] args) throws Exception{
Create t1 = new Create ("e:\\f1.txt");
Create t2 = new Create ("e:\\f2.txt");Create t3 = new Create ("e:\\f3.txt");
t1.start();t2.start();t3.start();
boolean flag=false;
while (! t1.isFinished || !t2.isFinished || ! t3.isFinished);
System.out.println("done....");
JavaApplication2 obj = new JavaApplication2();
ReadFile r1 = new ReadFile("e:\\f1.txt",obj);
ReadFile r2 = new ReadFile("e:\\f2.txt",obj);
ReadFile r3 = new ReadFile("e:\\f3.txt",obj);
r1.start();r2.start();r3.start();
while (! r1.isFinished || !r2.isFinished || ! r3.isFinished);
System.out.println("r1: "+r1.sum);
System.out.println("r2: "+r2.sum);
System.out.println("r3: "+r3.sum);
}
static void getInfo(double s){
if (c++==0)
max=s;
if (s>max)
max=s;
if (c==3)
System.out.println("max: "+max);
}
void getInfoRef(double s, String name){
counter++;
System.out.println(name+"\t"+s);
if (counter ==3)
System.out.println("done ref method");
System.out.println("counter : "+counter);
}
}
```

Lecture4:

```
//1/12/2021 lecture
package javaapplication2;
import java.io.*;
class PrintValues{
void print(String name) {
```

```
for (int i=0;i<20;i++)
System.out.println(name+" unsynchonized zone: "+i);
synchronized(this){
try{
if (name.equals("T3"))
wait(1000);
}catch(Exception e){
e.printStackTrace();
}
for (int i=0;i<20;i++)
System.out.println(name+"\t"+i);
}
}
}
class ReadFile extends Thread{
PrintValues obj;
String name;
public ReadFile(String nam, PrintValues obj) {
this.obj=obj;
name=nam;
}
public void run(){
for (int i=0;i<20;i++)
System.out.println(name+" inside run "+i);
obj.print(name);
}
}
public class JavaApplication2{
public static void main(String[] args) throws Exception{
PrintValues pv = new PrintValues();
```

```
ReadFile r1 = new ReadFile("T1",pv);
ReadFile r2 = new ReadFile("T2",pv);
ReadFile r3 = new ReadFile("T3",pv);
r1.start();r2.start();r3.start();
}
}
```

Bottom of Form

Lecture5: 2-12-2021

```
package javaapplication3;
import java.io.*;
import static java.lang.Math.random;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
class ReadFile extends Thread
{
volatile boolean finish;
String nof;
int sum;
int count;
String arr[] = new String[4];
public ReadFile(String nof){
this.nof=nof;
}
public void run() {
String line;
```

```
try {
BufferedReader b = new BufferedReader (new FileReader(nof));
while((line=b.readLine())!=null){
for(int i=0;i<4;i++)
{
arr= line.split(" ");
}
for (int i = 1; i < 4; i++) {
sum=0;
sum+=Integer.parseInt(arr[i]);
}
if(sum>50)
count++;
System.out.println(count);
}
finish=true;
b.close();
}
catch(Exception e)
{
}
}
}
public class JavaApplication3 {
public static void main(String[] args)throws Exception
{
ReadFile r1=new ReadFile("d://f1.txt");
ReadFile r2=new ReadFile("d://f1.txt");
ReadFile r3=new ReadFile("d://f1.txt");
r1.start();
```

```
r2.start();
r3.start();
while(!r1.finish||!r2.finish||!r3.finish);
System.out.println(r1.count);
System.out.println(r2.count);
System.out.println(r3.count);
}
}
```

Lecture6:

```
package javaapplication2;
import java.util.concurrent.*;
import java.io.*;
class ReadFile implements Callable <Integer>{
String fname;
public ReadFile(String fname) {
this.fname = fname;
}
public Integer call(){
int sum=0;
try{
BufferedReader buf = new BufferedReader(new FileReader(fname));
String n;
while ((n=buf.readLine())!=null)
sum +=Integer.parseInt(n);
}catch(Exception e){
e.printStackTrace();
}
return sum;
}
}
public class JavaApplication2 {
public static void main(String[] args) throws Exception{
ReadFile r1 = new ReadFile("d:\\1.txt");
ReadFile r2 = new ReadFile("d:\\2.txt");
ReadFile r3 = new ReadFile("d:\\3.txt");
ExecutorService service = Executors.newFixedThreadPool(3);
Future <Integer> x = service.submit(r1);
Future <Integer> y = service.submit(r2);
Future <Integer> z = service.submit(r3);
System.out.println(x.get());
System.out.println(y.get());
System.out.println(z.get());
service.shutdown();
}
}
```

Ayman Rabaya

```
package javaapplication7;
import java.util.concurrent.*;
import java.io.*;
class array implements Callable <Void>{
int s,e,arr[];
public array(int s, int e, int[] arr) {
this.s = s;
this.e = e;
this.arr = arr;
}
public Void call(){
for (int i=s;i<e;i++)
```

```
arr[i]=(int)(Math.random()*100);
return null;
}
}
class maxarr implements Callable <Integer>{
Object O;
int arr[];
int s,e;
public maxarr(int arr[],int s,int e) {
this.arr=arr;
this.e=e;
this.s=s;
}
public Integer call(){
int max=arr[s];
for (int i=s;i<e;i++){
if(arr[i]>max)
max=arr[i];
}
return max;
}
}
public class JavaApplication7 {
public static void main(String[] args) throws Exception{
int a[]=new int[1000];
array t1=new array(0,a.length/2,a);
array t2=new array(a.length/2,a.length,a);
ExecutorService service = Executors.newFixedThreadPool(2);
//ExecutorService service2 = Executors.newFixedThreadPool(2);
Future <Void> x = service.submit(t1);
Future <Void> y = service.submit(t2);
x.get();y.get();
service.shutdown();
maxarr r1=new maxarr(a,0,a.length/2);
maxarr r2=new maxarr(a,a.length/2,a.length);
service = Executors.newFixedThreadPool(2);
Future <Integer> x1 = service.submit(r1);
Future <Integer> y1 = service.submit(r2);
int z,d;
z=x1.get();
d=y1.get();
if(z>d)
System.out.println(z);
else
System.out.println(d);
service.shutdown();
}
}
```

Lecture7: 9-12-2021

```
package javaapplication8;
import java.net.*;
import java.io.*;
import java.util.ArrayList;
import java.util.concurrent.*;
class FindD implements Callable <String>
{
String dName;
public FindD(String name) {
this.dName = name;
```

```
}
public String call()
{
String ip="";
ip=dName+" \n "+" ---------- "+"\n";
try{
InetAddress a[]=InetAddress.getAllByName(dName);
for(InetAddress a1:a)
ip=ip+" "+a1.getHostAddress();
}catch(Exception e){}
return ip;
}
}
public class JavaApplication8 {
public static void main(String[] args) throws Exception{
ArrayList <String> as =new ArrayList<String>();
ArrayList <Future> f =new ArrayList<Future>();
ArrayList <FindD> t =new ArrayList<FindD>();
BufferedReader bf=new BufferedReader(new FileReader("d://domin.txt"));
String line;
while((line=bf.readLine())!=null)
as.add(line);
for (int i = 0; i < as.size(); i++)
t.add(new FindD(as.get(i)));
ExecutorService service = Executors.newFixedThreadPool(as.size());
for (int i = 0; i < as.size(); i++)
f.add(service.submit(t.get(i)));
for (int i = 0; i < as.size(); i++)
System.out.println(f.get(i).get());
service.shutdown();
```

```
}
}
```

Lecture8: 15-12-2021

```
package javaapplication1;
import java.io.*;
import java.util.*;
import java.net.*;
public class JavaApplication1 {
public static void main(String[] args) throws Exception {
byte addr[]=new byte[]{(byte)192,(byte)168,1,38};
NetworkInterface ni=NetworkInterface.getByInetAddress(InetAddress.getByAddress(addr));
//185.37.12.3
Enumeration <NetworkInterface>interfaces=NetworkInterface.getNetworkInterfaces();
NetworkInterface inter;
while (interfaces.hasMoreElements()){
inter=interfaces.nextElement();
System.out.println(inter.getDisplayName()+"\n-----------------------------\n");
Enumeration <InetAddress> addresses = inter.getInetAddresses();
while (addresses.hasMoreElements()){
InetAddress add = addresses.nextElement();
System.out.println(add.getHostAddress());
}
}
}
}
```

Lecture9: 16-12-2021

```
package javaapplication10;
import java.net.*;
import java.io.*;
import java.util.*;
import java.util.concurrent.*;
class FindInterface implements Callable <NetworkInterface>
{
String ipName;
public FindInterface(String name) {
this.ipName = name;
}
public NetworkInterface call()
{
NetworkInterface nI=null;
String s[]=ipName.split("\\.");
byte b[]=new byte[s.length];
for (int i = 0; i <s.length; i++) {
b[i]=(byte)Integer.parseInt(s[i]);
}
try{
InetAddress addrs=InetAddress.getByAddress(b);
nI=NetworkInterface.getByInetAddress(addrs);
}catch(Exception e){e.printStackTrace();}
return nI;
}
}
public class JavaApplication10 {
public static void main(String[] args) throws Exception{
```

```
ArrayList <String> ips =new ArrayList<String>();
ArrayList <Future> f =new ArrayList<Future>();
ArrayList <FindInterface> t =new ArrayList<FindInterface>();
BufferedReader bf=new BufferedReader(new FileReader("d://ips.txt"));
String line;
while((line=bf.readLine())!=null)
ips.add(line);
for (int i = 0; i < ips.size(); i++)
t.add(new FindInterface(ips.get(i)));
ExecutorService service = Executors.newFixedThreadPool(ips.size());
for (int i = 0; i < ips.size(); i++)
f.add(service.submit(t.get(i)));
for (int i = 0; i < ips.size(); i++)
System.out.println(f.get(i).get());
service.shutdown();
}
}
```

Lecture10: 22-12-2021

```
package javaapplication14;
import java.net.*;
import java.util.concurrent.*;
import java.io.*;
import java.util.ArrayList;
class GetNetworkInterface implements Callable<String>
{
String ip;
public GetNetworkInterface (String ip)
```

```
{
this.ip=ip;
}
public String call(){
try{
String temp[]=ip.split("\\.");
byte address[]=new byte[4];
for (int i=0;i<temp.length;i++)
address[i]=(byte)(Integer.parseInt(temp[i]));
InetAddress w=InetAddress.getByAddress(address);
NetworkInterface in= NetworkInterface.getByInetAddress(w);
return in.getDisplayName();
}
catch(Exception e)
{
return null;
}
}
}
public class JavaApplication14 {
public static void main(String[] args) throws Exception{
GetNetworkInterface t1 =new GetNetworkInterface ("180.145.2.3");
//t1.call();
BufferedReader buf=new BufferedReader (new FileReader("e:\\q.txt"));
ArrayList <String> ipAddress = new ArrayList<String>();
String data;
while((data=buf.readLine())!=null)
ipAddress.add(data);
buf.close();
System.out.println(ipAddress);
ArrayList <Future> results=new ArrayList <Future>();
ArrayList<GetNetworkInterface> objects =new
ArrayList<GetNetworkInterface>();
for(int i=0;i<ipAddress.size();i++){
objects.add(new GetNetworkInterface(ipAddress.get(i)));
// Future r=service.submit(temp);
// results.add(r);
}
ExecutorService service=Executors.newFixedThreadPool(ipAddress.size());
for (int i=0;i<ipAddress.size();i++){
results.add(service.submit(objects.get(i)));}
for (int i=0;i<results.size();i++)
System.out.println((String)results.get(i).get());
}
}
```

Lecture11: 23-12-2021

```
package javaapplication2;
import java.io.*;
import java.net.*;
import java.util.ArrayList;
public class JavaApplication2 {
public static void main(String[] args) throws Exception{
/*
URL url = new
URL("https://ahmad@google.com:8750/news/sport/view.aspx?n=4&g=w");
System.out.println(url.getAuthority());
System.out.println(url.getHost());
System.out.println(url.getPath());
System.out.println(url.getPort());
System.out.println(url.getUserInfo());
System.out.println(url.getQuery());
URL url2=new URL(url,"exams/midterm/nwp.doc");
System.out.println(url2);
URL url3=new URL("http","www.yahoo.com","/news/view.aspx.v=4");
System.out.println(url3);
URL url4=new URL("http","www.yahoo.com",21,"/news/view.aspx.v=4");
System.out.println(url4);
URL u=new URL("https://www.aaup.edu") ;
/* BufferedInputStream bufs = new BufferedInputStream(u.openStream());
byte data[]=new byte[1024] ;
while ((bufs.read(data))!=-1)
System.out.println(new String(data));
bufs.close();
*/
ArrayList<String>links=new ArrayList<String>();
ArrayList <URL> urls = new ArrayList<URL>();
ArrayList<BufferedReader>buffers = new ArrayList<BufferedReader>();
try{
BufferedReader buf = new BufferedReader(new
FileReader("D://domains.txt"));
String line;
while((line=buf.readLine())!=null)
{
links.add(line);
urls.add(new URL(line));
}
for (int i = 0; i < urls.size(); i++) {
String data;
buffers.add(new BufferedReader(new
InputStreamReader(urls.get(i).openStream())));
while((data=buffers.get(i).readLine())!=null)
{ System.out.println(data); }
buffers.get(i).close();
}
}
catch(Exception e){}
}
}
```

Lecture12: 30-12-2021

```
package javaapplication1;
import java.awt.image.BufferedImage;
import java.io.*;
import java.net.*;
import javax.imageio.ImageIO;
public class JavaApplication1 {
static String getRate(String c1, String c2)throws Exception{
URL url = new URL("https://www.x-rates.com/calculator/?from="+c1+"&to="+c2+"&amount=1");
BufferedReader buf = new BufferedReader(new InputStreamReader(url.openStream()));
String line;
while ((line=buf.readLine())!=null)
if (line.contains("ccOutputRslt"))
return line.substring(line.indexOf(">")+1,line.indexOf("<span class=\"ccOutputTrail"));
return null;
}
public static void main(String[] args) throws Exception{
//System.out.println(getRate("EUR","ILS"));
URL url = new URL("https://www.aaup.edu/.../slide.../public/AAUJ-to-AAUP-en.jpg");

if (url.getContent().getClass().getName().contains("Image"))

{

BufferedImage img= ImageIO.read(url);

if (ImageIO.write(img, "jpg", new File ("e:\\aaup.jpg")))

System.out.println("DONE..");

else

System.out.println("unknown error!");

}

}

}
```

Lecture13: 5-1-2022

```
package javaapplication5;

import java.io.*;

import java.net.*;

public class JavaApplication5 {

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

URL url = new URL("https://file-examples-com.github.io/.../file_example_MP4...");

BufferedInputStream bufs = new BufferedInputStream(url.openStream());

BufferedOutputStream bufw = new BufferedOutputStream(new FileOutputStream("e:\\samplev.mp4"));

System.out.println(url.getContent().getClass().getName());

byte arr[]=new byte[1024];

int data;

while ((data=bufs.read(arr))!=-1)

{

bufw.write(arr, 0, data);
}
bufw.close();
}
}
```

Lecture14: 6-1-2022

```
package javaapplication2;
import java.awt.image.BufferedImage;
import java.io.*;
import java.net.*;
import java.util.ArrayList;
import java.util.concurrent.Callable;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.Future;
import javax.imageio.ImageIO;
class Find implements Callable <String>{
String data;
public Find(String data){
this.data=data;
}
public String call() throws Exception{
URL url = new URL(data);
if(url.getContent().getClass().getName().contains("Image")){
BufferedImage m= ImageIO.read(url);
ImageIO.write(m, "jpg", new File("d:\\image"));
}
if(url.getPath().contains(".mp4")){
BufferedInputStream bufs = new BufferedInputStream(url.openStream());
```

```
BufferedOutputStream bufw = new BufferedOutputStream(new FileOutputStream("d:\\samplev.mp4"));
byte arr[]=new byte[1024];
int data;
while ((data=bufs.read(arr))!=-1)
{
bufw.write(arr, 0, data);
}
bufw.close();
}
return null;
}
}
public class NewClass {
public static void main(String[] args) {
try{
ArrayList <String> link=new ArrayList<String>();
ArrayList<Find> give=new ArrayList<Find>();
ArrayList <Future> a1=new ArrayList<Future>();
BufferedReader buf=new BufferedReader(new FileReader("d:\\links.txt"));
String data;
while((data=buf.readLine())!=null){
link.add(data);
// System.out.println(data);
}
for(int i=0;i<link.size();i++){
give.add(new Find(link.get(i)));
}
ExecutorService service=Executors.newFixedThreadPool(link.size());
for (int i=0;i<link.size();i++){
a1.add(service.submit(give.get(i)));
}
service.shutdown();
}catch(Exception e){e.printStackTrace();}
}
}
```

Lecture15:

Lecture16:

Lecture17:

Lecture18:

Lecture19:

Lecture20: