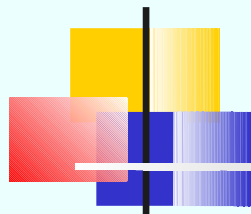




Lập trình mạng

Thread trong Java

Giảng viên: TS. Nguyễn Mạnh Hùng
Học viện Công nghệ Bưu chính Viễn thông (PTIT)



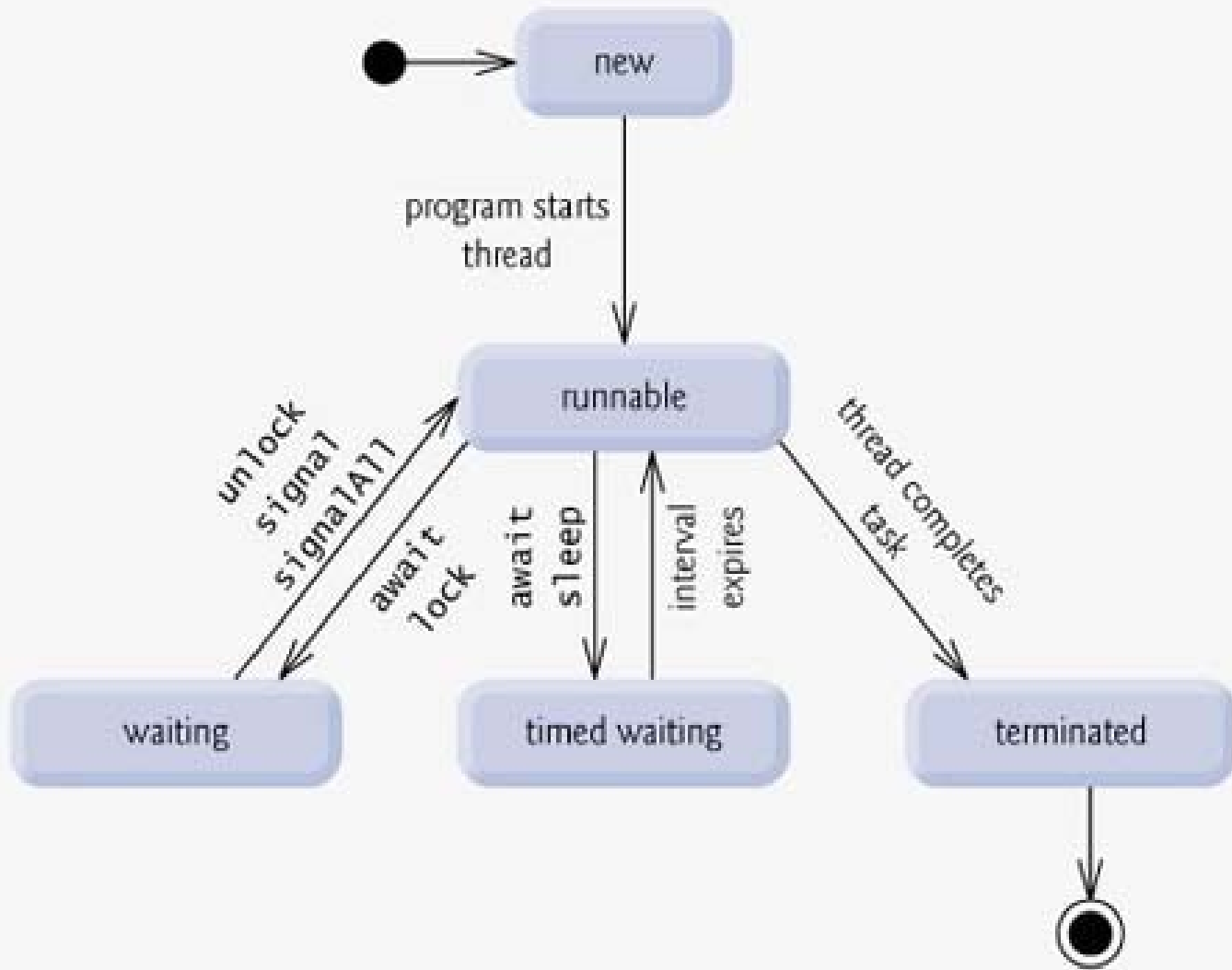
Nội dung

- Mô hình kiến trúc Thread
- Khai báo và phương thức chính
- Trao đổi dữ liệu giữa các thread
- Ví dụ
- Bài tập



Mô hình Thread

Mô hình Thread





Khai báo

Cách 1:

```
public class LogWriter extends Thread{
```

Cách 2:

```
public class LogWriter implements Runnable{
```



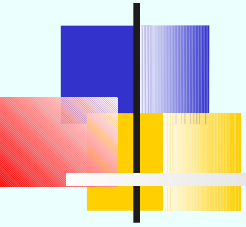
Phương thức chính

```
public class LogWriter extends Thread{
    private String filename;
    private long time;

    public void run(){
        for(int i=0; i<10; i++){
            try{
                Writer wr = new BufferedWriter(new
                    FileWriter(filename,true));
                this.sleep(time);
                wr.append(getName() + "[" +
                    Calendar.getInstance().getTime() + "]: Log-" + i
                    + "\r\n");
                wr.close();

            }catch(Exception e){
                System.out.println(e.getStackTrace());
            }
        }
    }
}
```

**Ví dụ: Ghi file log cho nhiều
người dùng đồng thời**





Lớp LogWriter (1)

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.Writer;
import java.util.Calendar;

public class LogWriter extends Thread{
    private String filename;
    private long time;

    public LogWriter(String name, String filename, long time){
        super(name);
        this.filename = filename;
        this.time = time;
    }
}
```




Lớp LogWriter (2)

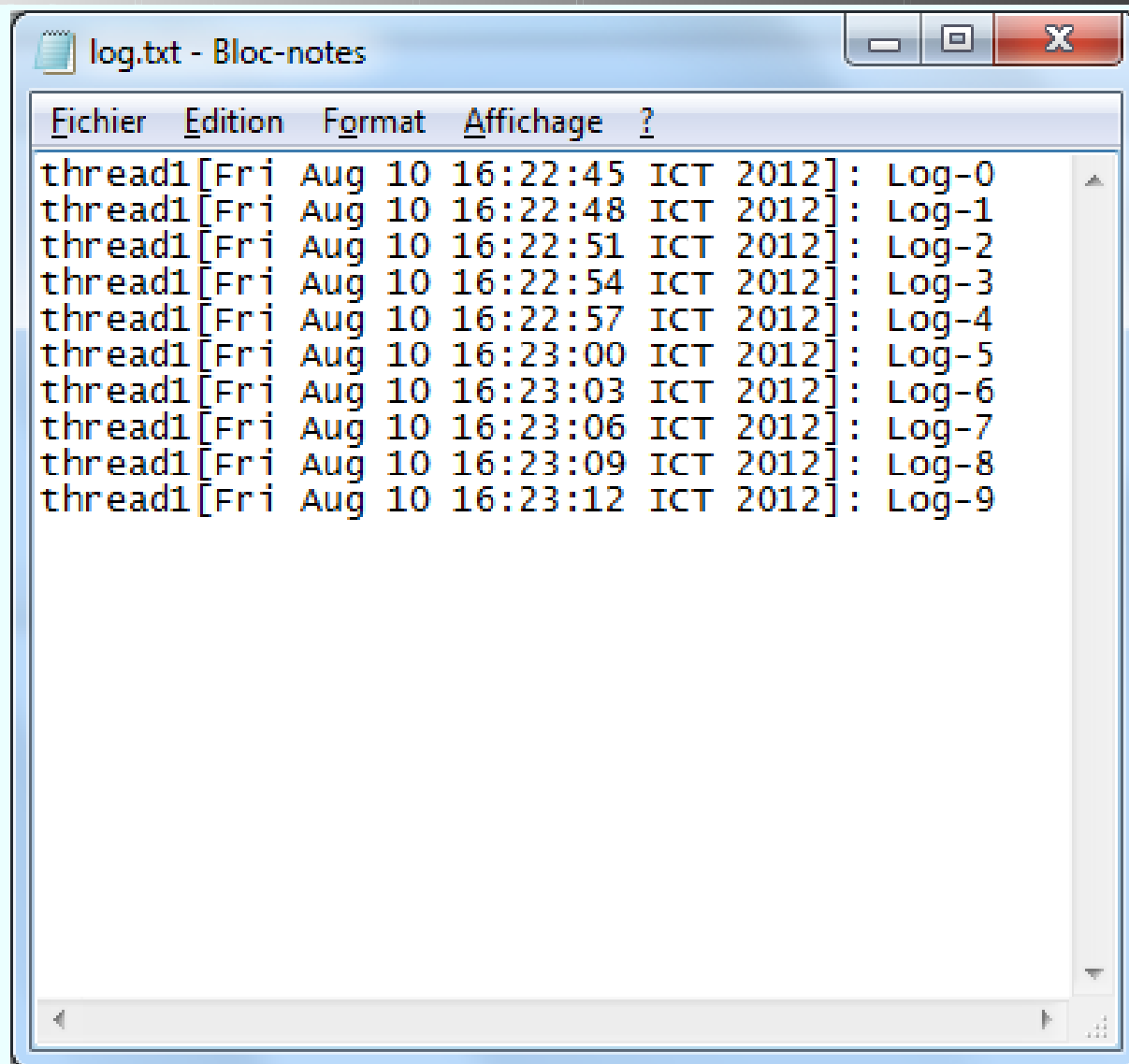
```
public void run(){
    for(int i=0; i<10; i++){
        try{
            Writer wr = new BufferedWriter(new
                FileWriter(filename,true));
            this.sleep(time);
            wr.append(getName() + "[" +
                Calendar.getInstance().getTime() + "]: Log-" +
                i + "\r\n");
            wr.close();
        }catch(Exception e){
            System.out.println(e.getStackTrace());
        }
    }
}
```



Test (1)

```
public class Test {  
    public static void main(String[] args){  
  
        LogWriter lw1 = new LogWriter("thread1", "log.txt", 3000);  
        lw1.start();  
    }  
}
```

Kết quả (1)



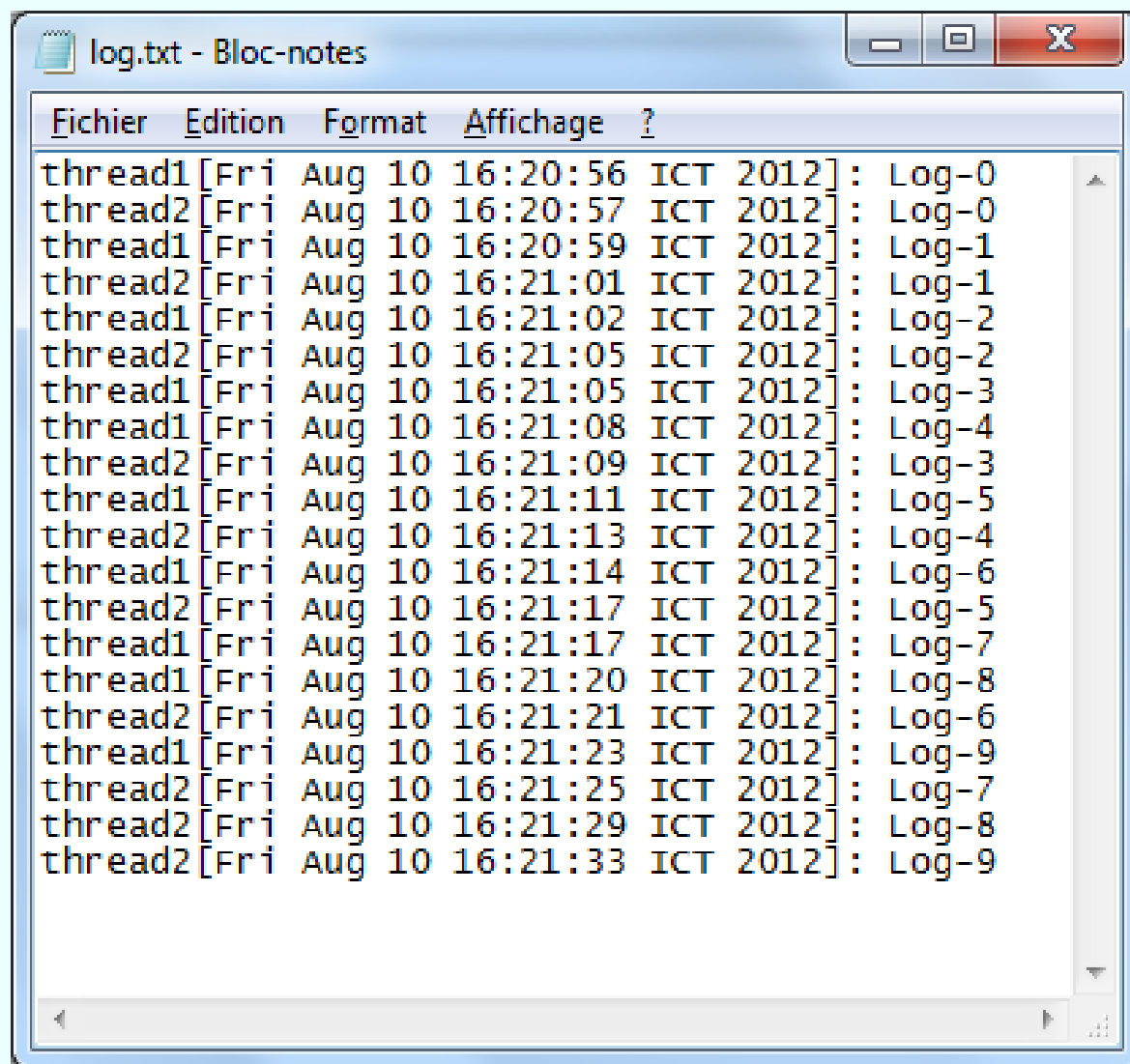
```
log.txt - Bloc-notes
Fichier  Edition  Format  Affichage  ?
thread1[Fri Aug 10 16:22:45 ICT 2012]: Log-0
thread1[Fri Aug 10 16:22:48 ICT 2012]: Log-1
thread1[Fri Aug 10 16:22:51 ICT 2012]: Log-2
thread1[Fri Aug 10 16:22:54 ICT 2012]: Log-3
thread1[Fri Aug 10 16:22:57 ICT 2012]: Log-4
thread1[Fri Aug 10 16:23:00 ICT 2012]: Log-5
thread1[Fri Aug 10 16:23:03 ICT 2012]: Log-6
thread1[Fri Aug 10 16:23:06 ICT 2012]: Log-7
thread1[Fri Aug 10 16:23:09 ICT 2012]: Log-8
thread1[Fri Aug 10 16:23:12 ICT 2012]: Log-9
```



Test (2)

```
public class Test {  
    public static void main(String[] args){  
  
        LogWriter lw1 = new LogWriter("thread1", "log.txt", 3000);  
        LogWriter lw2 = new LogWriter("thread2", "log.txt", 4000);  
        lw1.start();  
        lw2.start();  
    }  
}
```

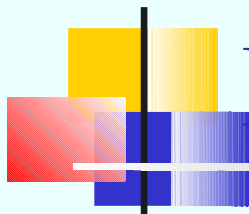
Kết quả (2)



```
log.txt - Bloc-notes
Fichier  Edition  Format  Affichage  ?
thread1[Fri Aug 10 16:20:56 ICT 2012]: Log-0
thread2[Fri Aug 10 16:20:57 ICT 2012]: Log-0
thread1[Fri Aug 10 16:20:59 ICT 2012]: Log-1
thread2[Fri Aug 10 16:21:01 ICT 2012]: Log-1
thread1[Fri Aug 10 16:21:02 ICT 2012]: Log-2
thread2[Fri Aug 10 16:21:05 ICT 2012]: Log-2
thread1[Fri Aug 10 16:21:05 ICT 2012]: Log-3
thread1[Fri Aug 10 16:21:08 ICT 2012]: Log-4
thread2[Fri Aug 10 16:21:09 ICT 2012]: Log-3
thread1[Fri Aug 10 16:21:11 ICT 2012]: Log-5
thread2[Fri Aug 10 16:21:13 ICT 2012]: Log-4
thread1[Fri Aug 10 16:21:14 ICT 2012]: Log-6
thread2[Fri Aug 10 16:21:17 ICT 2012]: Log-5
thread1[Fri Aug 10 16:21:17 ICT 2012]: Log-7
thread1[Fri Aug 10 16:21:20 ICT 2012]: Log-8
thread2[Fri Aug 10 16:21:21 ICT 2012]: Log-6
thread1[Fri Aug 10 16:21:23 ICT 2012]: Log-9
thread2[Fri Aug 10 16:21:25 ICT 2012]: Log-7
thread2[Fri Aug 10 16:21:29 ICT 2012]: Log-8
thread2[Fri Aug 10 16:21:33 ICT 2012]: Log-9
```



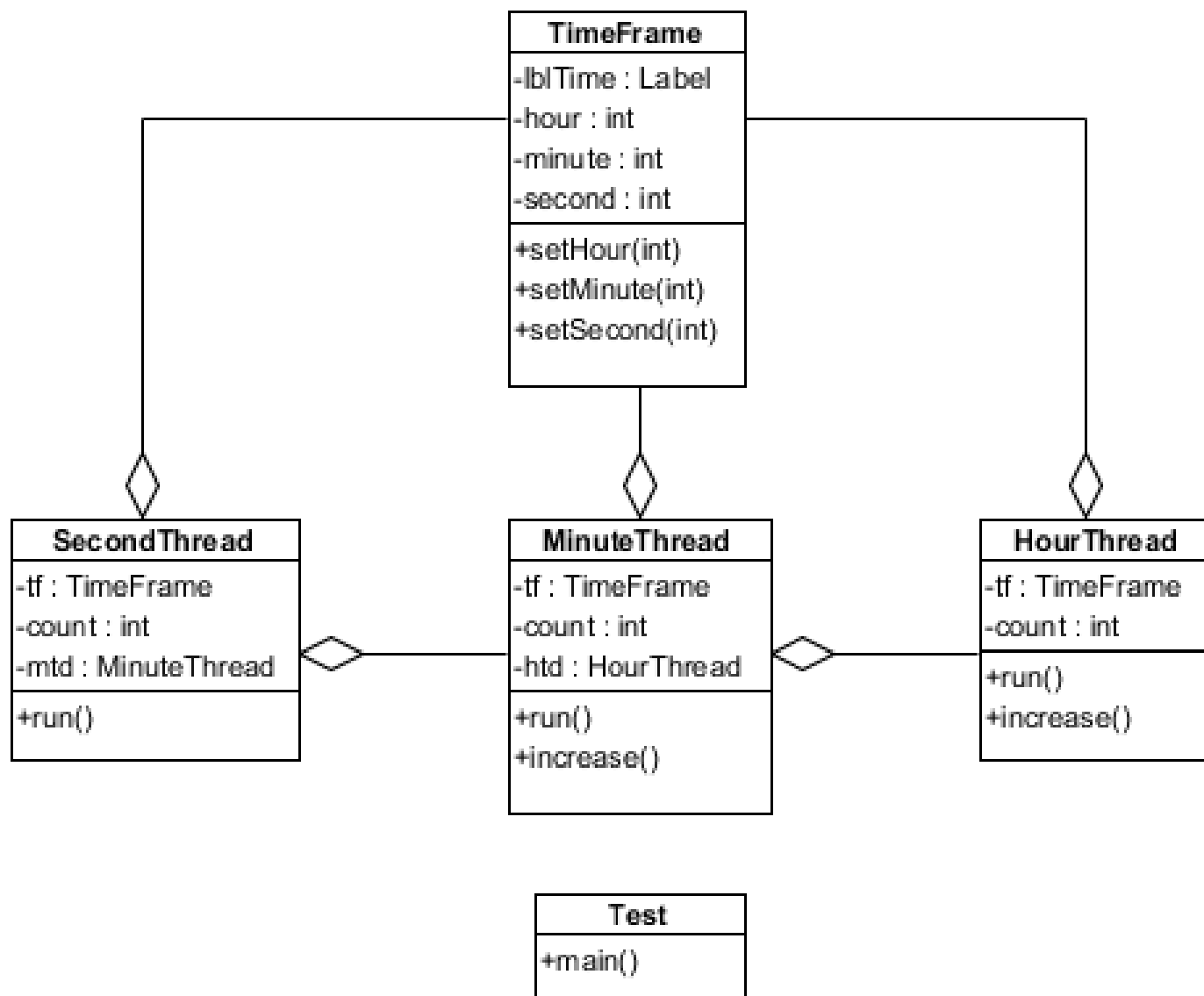
Trao đổi dữ liệu giữa các thread



Bài toán

- Xây dựng bộ đếm thời gian chạy theo giờ, phút, giây
- 3 thread tương ứng với giờ, phút, giây
- Thread giây: mỗi giây đếm tăng 1, đến 60 thì reset về 0 và yêu cầu thread phút tăng 1. Hiển thị giây hiện tại
- Thread phút: khi bộ đếm tăng đến 60 thì reset về 0 và yêu cầu thread giờ tăng lên 1. Hiển thị phút hiện tại
- Thread giờ: hiển thị giờ hiện tại

Sơ đồ lớp





Lớp TimeFrame (1)

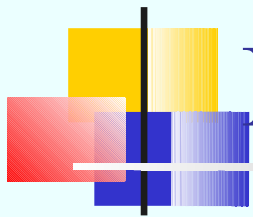
```
import java.awt.Frame;
import java.awt.Label;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

public class TimeFrame extends Frame{
    private Label lblTime;
    private int hour=0, minute=0, second=0;

    public TimeFrame(){
        super("Time Counter");
        this.setSize(250, 50);
        this.setResizable(false);
        this.setLayout(null);
        this.setLocation(400, 400);

        lblTime = new Label(hour+": "+minute+ " "+second);

        lblTime.setAlignment(2);
        lblTime.setSize(50, 20);
        lblTime.setLocation(5,25);
        this.add(lblTime);
    }
}
```



Lớp TimeFrame (2)

```
        this.addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e){
                System.exit(0);
            }
        });
    }

    public void setHour(int hour) {
        this.hour = hour;
        lblTime.setText(hour+":"+minute+ ":"+second);
    }

    public void setMinute(int minute) {
        this.minute = minute;
        lblTime.setText(hour+":"+minute+ ":"+second);
    }

    public void setSecond(int second) {
        this.second = second;
        lblTime.setText(hour+":"+minute+ ":"+second);
    }
}
```



Lớp HourThread

```
public class HourThread extends Thread{
    private TimeFrame tf;
    private int count;

    public HourThread(TimeFrame tf){
        super();
        this.tf = tf;
        count = 0;
    }

    public void increase(){
        count++;
        tf.setHour(count);
    }

    public void run(){
        while(true){
            try{
                tf.setHour(count);
            }catch(Exception e){System.out.println(e.getStackTrace());}
        }
    }
}
```



Lớp MinuteThread (1)

```
public class MinuteThread extends Thread{
    private TimeFrame tf;
    private HourThread htd;
    private int count;

    public MinuteThread(TimeFrame tf, HourThread htd){
        super();
        this.tf = tf;
        this.htd = htd;
        count = 0;
    }

    public void increase(){
        count++;
        if(count == 60){
            htd.increase();
            count = 0;
        }
        tf.setMinute(count);
    }
}
```



Lớp MinuteThread (2)

```
public void run(){
    while(true){
        try{
            tf.setMinute(count);
        }catch(Exception e){
            System.out.println(e.getStackTrace());
        }
    }
}
```



Lớp SecondThread (1)

```
public class SecondThread extends Thread{  
    private TimeFrame tf;  
    private MinuteThread mtd;  
    private int count;  
  
    public SecondThread(TimeFrame tf, MinuteThread mtd){  
        super();  
        this.tf = tf;  
        this.mtd = mtd;  
        count = 0;  
    }  
}
```



Lớp SecondThread (2)

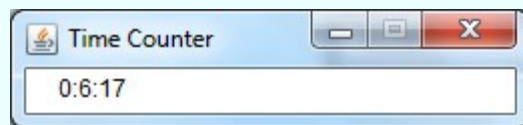
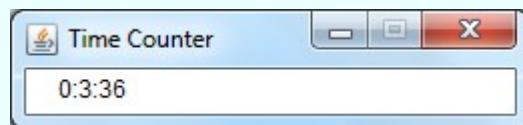
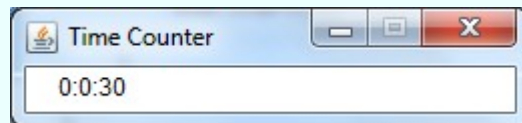
```
public void run(){
    while(true){
        try{
            this.sleep(1000);
            count++;
            if(count == 60){
                count = 0;
                mtd.increase();
            }
            tf.setSecond(count);
        }catch(Exception e){
            System.out.println(e.getStackTrace());
        }
    }
}
```

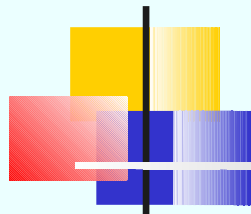


Lớp Test

```
public class Test {  
    public static void main(String[] args){  
  
        TimeFrame tf = new TimeFrame();  
        tf.setVisible(true);  
        HourThread htd = new HourThread(tf);  
        MinuteThread mtd = new MinuteThread(tf, htd);  
        SecondThread std = new SecondThread(tf, mtd);  
        htd.start();  
        mtd.start();  
        std.start();  
    }  
}
```


Kết quả





Bài tập

- Cài đặt lại ví dụ trong bài theo mô hình MVC
- Viết chương trình copy toàn bộ 1 thư mục với nhiều file, mỗi thread chỉ copy một file, tiến hành copy song song



Questions?
