HD LAB 6: ĐA LUỒNG

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Tạo lớp Thread bằng cách kế thừa lớp Thread
- ✓ Tạo và start thread bằng cách sử dụng interface Runnable
- ✓ Sử dụng độ ưu tiên của Thread
- ✓ Đồng bộ các thread

PHẦN I

Bài 1 (2 điểm)

Tạo file MyThread.java thực thi interface Runnable và thực hiện các công việc sau:

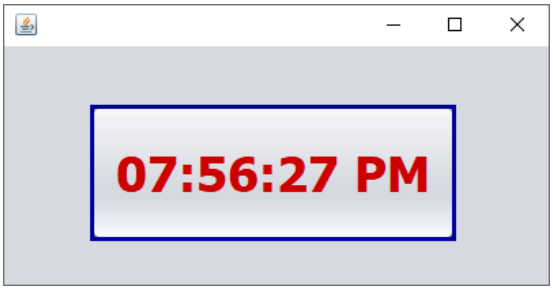
- ✓ Trong phương thức run() chứa một vòng lặp in ra 10 số tự nhiên đầu tiên và
- ✓ mỗi lần in cách nhau 500 milliseconds.
- ✓ Tạo 2 object từ class MyThread là Thread1 và Thread2 với thứ tự ưu tiên tương ứng mà MAX_PRIORITY và MIN_PRIORITY.
- ✓ Gọi phương thức start() cho cả 2 thread

```
public class MyThread implements Runnable{
    @Override
    public void run() {
        for(int i=1;i<=10;i++){
            System.out.println(Thread.currentThread().getName() +i);
            try{
                Thread.sleep(500);
            }catch(InterruptedException e){
                System.out.println(e);
            }
        }
    }
```

```
    public static void main(String[] args) {
        MyThread ob1 = new MyThread();
        MyThread ob2 = new MyThread();
        Thread th1 = new Thread(ob1);
        Thread th2 = new Thread(ob2);
        th1.setName("Thread 1: ");
        th2.setName("Thread 2: ");
        th1.setPriority(Thread.MAX_PRIORITY);
        th2.setPriority(Thread.MIN_PRIORITY);
        th1.start();
        th2.start();
    }
}
```

Bài 2 (2 điểm)

Hiển thị đồng hồ hệ thống lên nút có định dạng hh:mm:ss aa như hình sau. Đồng hồ bắt đầu hiển thị khi click vào nút. Khi đồng hồ đã chạy thì làm vô hiệu hóa nút.



HƯỚNG DẪN

- ✓ Thiết kế giao diện như hình trên với nút btnClock và cửa sổ ClockJFrame
- ✓ Thực thi cửa sổ ClockJFrame thực thi interface Runnable
- ✓ Viết mã cho nút btnClock và phương thức run()

```
    private void btnClockActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Thread t1 = new Thread(this);
        t1.start();
        btnClock.setEnabled(false);
    }
```

```
    @Override
    public void run() {
        while(true){
            try {
                Date now = new Date();
                SimpleDateFormat sdf = new SimpleDateFormat();
                sdf.applyPattern("hh:mm:ss aa");
                String time = sdf.format(now);
                btnClock.setText(time);
                Thread.sleep(1000);
            } catch (Exception e) {
                break;
            }
        }
    }
}
```

PHẦN II

Bài 3 (2 điểm)

1. Tạo lớp **OddThread** và **EvenThread** kế thừa từ **Thread** và thực hiện theo yêu cầu:

   - ✓ Viết run() của OddThread sao cho xuất các số lẻ từ 1 đến 10, các số được xuất ra cách nhau 10 mili giây.
   - ✓ Viết run() của EvenThread sao cho xuất các số chẵn từ 1 đến 10, các số được xuất ra cách nhau 15 mili giây.

```
public class OddThread extends Thread{
    @Override
    public void run() {
        for(int i=1;i<=10;i++){
            if(i%2 != 0){
                System.out.print(i+" ");
            }
            try{
                Thread.sleep(10);
            }catch(InterruptedException e){
                break;
            }
        }
    }
}
```

```
public class EvenThread extends Thread{
    @Override
    public void run() {
        for(int i=1;i<=10;i++){
            if(i%2 == 0){
                System.out.print(i+" ");
            }
            try{
                Thread.sleep(15);
            }catch(InterruptedException e){
                break;
            }
        }
    }
}
```

2. Tạo lớp **TestThread** chứa phương thức main() sau đó tạo 2 đối tượng từ 2 lớp OddThread và EvenThread, start() các thread này.

- ✓ Sử dụng join() để cho phép xuất các số lẻ trước mới đến các số chẵn.

```
public class TestThread {
    public static void main(String[] args) {
        try {
            OddThread t1 = new OddThread();
            EvenThread t2 = new EvenThread();
            t1.start();
            t1.join();
            t2.start();
        } catch (Exception ex) {
            System.out.println("Error: "+ex);
        }
    }
}
```

Bài 4 (2 điểm)

Viết chương trình sổ số 3 số (trăm, chục và đơn vị) như giao diện sau.

Yêu cầu: Khi nhấp nút Start của số nào thì bắt đầu sinh 1000 lần số ngẫu nhiên từ 0 đến 9 và đặt số đó lên ô tương ứng. Các số được hiển thị cách nhau 10 mili giây.

Vô hiệu hóa nút đã được click.

HƯỚNG DẪN:

- ✓ Thiết kế giao diện như trên
- ✓ Viết mã cho nút Start hàng trăm, hàng chục, hàng đơn vị

```
private void btnTramActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new Thread(){
        @Override
        public void run(){
            for(int i=0;i<1000;i++){
                try{
                    int so = (int) Math.round(Math.random()*9);
                    txtTram.setText(String.valueOf(so));
                    Thread.sleep(10);
                }catch(InterruptedException e){
                    break;
                }
            }
        }
    }.start();

}
```

```
private void btnChucActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new Thread(){
        @Override
        public void run(){
            for(int i=0;i<1000;i++){
                try{
                    int so = (int) Math.round(Math.random()*9);
                    txtChuc.setText(String.valueOf(so));
                    Thread.sleep(10);
                }catch(InterruptedException e){
                    break;
                }
            }
        }
    }.start();
}
```

```
private void btnDonviActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new Thread(){
        @Override
        public void run(){
            for(int i=0;i<1000;i++){
                try{
                    int so = (int) Math.round(Math.random()*9);
                    txtDonvi.setText(String.valueOf(so));
                    Thread.sleep(10);
                }catch(InterruptedException e){
                    break;
                }
            }
        }
    }.start();
}
```

Hết