

## 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

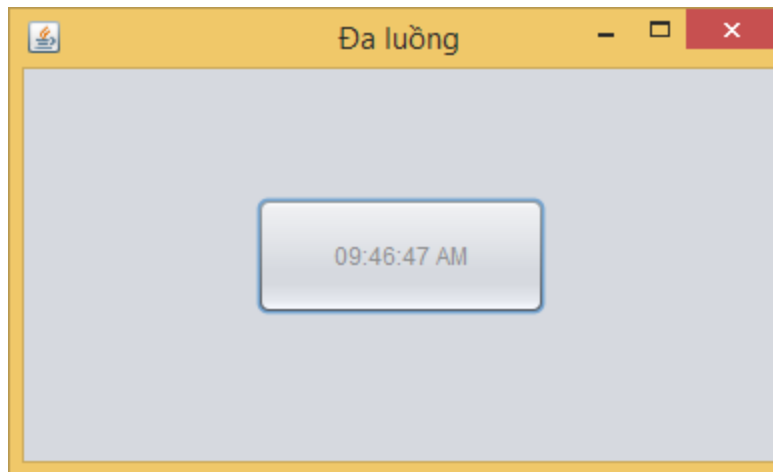
#### BÀI 1 (2 ĐIỂM)

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.

#### 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 theo interface Runnable
  - ClockJFrame implements Runnable
  - Viết mã cho phương thức run()
 

```
while (true) {
    try {
        // ... mã lấy thời gian hệ thống hiển thị lên nút
        Thread.sleep(1000);
    } catch (Exception e) {
        break;
    }
}
```
  - Lấy thời gian hệ thống và hiển thị lên nút btnClock
 

Sử dụng SimpleDateFormat để chuyển đổi thời gian hiện hành sang chuỗi dạng hh:mm:ss aa sau đó hiển thị lên nút

```
Date now = new Date();
SimpleDateFormat formater = new SimpleDateFormat();
formater.applyPattern("hh:mm:ss aa");
String time = formater.format(now);
btnClock.setText(time);
```
- ✓ Viết mã cho nút btnClock

Tạo một Thread với Runnable là đối tượng hiện tại, gọi start() để bắt đầu chạy sau đó vô hiệu hóa nút

```
Thread t1 = new Thread(this);
t1.start();
btnClock.setEnabled(false);
```

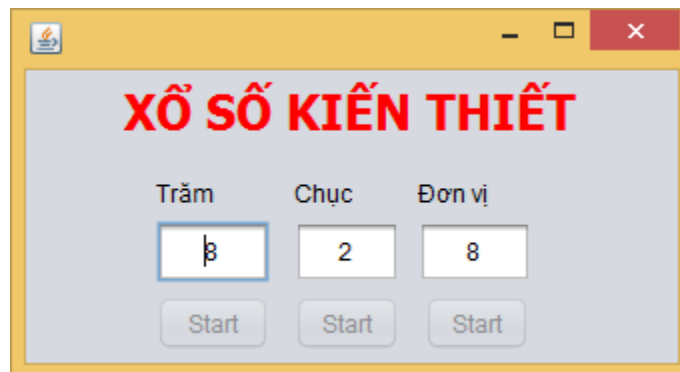
## PHẦN II

### BÀI 3 (2 ĐIỂM)

- 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.
- 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.

### BÀI 4 (2 ĐIỂM)

Viết chương trình số 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
  - Sử dụng kỹ thuật lớp nặc danh để tạo và start thread như sau

```
new Thread(){
    @Override
    public void run() {...}
}.start();
btnTram.setEnabled(false);
```
  - Bổ sung mã cho phương thức 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 ex) {
        break;
    }
}
```

**BÀI 5 (2 ĐIỂM)**

Giảng viên cho thêm