

```

import java.util.Observer;
import java.util.Observable;

class Chrono {

    public static void main (String ... args) {
        ChronoView view = new ChronoView();
        Counter cents, seconds, minutes;
        minutes = new Counter (0, 60, null, view);
        seconds = new Counter (0, 60, minutes, view);
        cents = new Counter(10, 100, seconds, view);
        view.setCounters(minutes, seconds, cents);
        cents.go();
    }

}

class ChronoView implements Observer{

    private Counter[] counters;

    public ChronoView () {}

    public void setCounters(Counter ... counters){
        this.counters= counters;
    }

    public void renderChronoState() {
        String txt = "";
        for (Counter cnt : counters) txt += cnt.getValue() + " ";
        System.out.println(txt);
    }

    public void update(Observable obs, Object arg) {
        renderChronoState();
    }

}

```

```

class Counter implements Observer {

    private int value = 0;

    private int maxVal;
    private Counter next;
    private Observer view;
    private int pulse;

    class ChangedValueEventObservers extends Observable {
        public void notifyObservers(){
            this.setChanged(); super.notifyObservers();
        }
    }
    class MaxValueReachedEventObservers extends Observable {
        public void notifyObservers(){
            this.setChanged(); super.notifyObservers();
        }
    }

    private ChangedValueEventObservers changedValueEventObservers
        = new ChangedValueEventObservers();

    private MaxValueReachedEventObservers maxValueReachedEventObservers
        = new MaxValueReachedEventObservers();

    public Counter (int pulse, int maxVal,
                    Counter nextCounter, Observer view) {
        this.maxVal = maxVal;
        this.next = nextCounter;
        this.view = view;
        this.pulse = pulse;

        changedValueEventObservers.addObserver(view);

        if (nextCounter != null)
            maxValueReachedEventObservers.addObserver(nextCounter);
    }

    public void go() {
        if (pulse > 0) {
            new Thread() {
                public void run() {
                    while(true) {
                        try {
                            Thread.sleep(pulse);
                            incrementValue();
                        }
                        catch (Exception e) {}
                    }
                }
            }.start();
        }
    }

    public int getValue() {return value;}

    public void update(Observable obs, Object arg) {
        incrementValue();
    }

    public void incrementValue(){
        value = ++ value % (maxVal);
        if (value == 0)
            maxValueReachedEventObservers.notifyObservers();

        changedValueEventObservers.notifyObservers();
    }
}

```