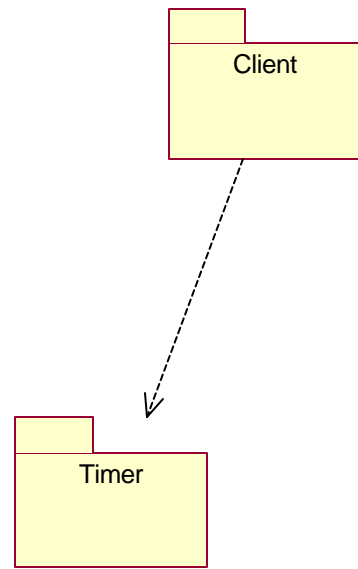
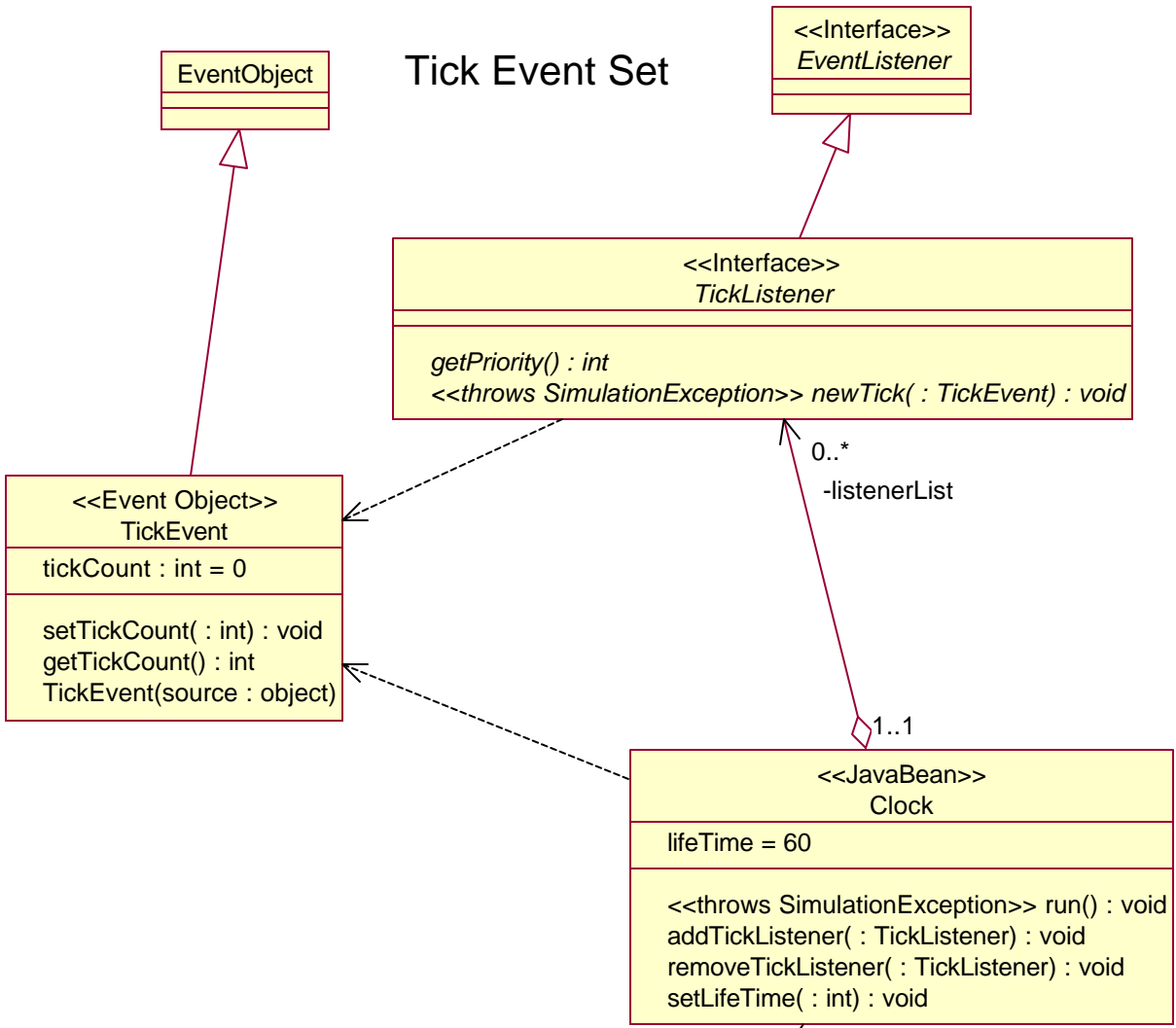


HW3 Solution
EE/CIS 694T
Spring 2000



Tick Event Set

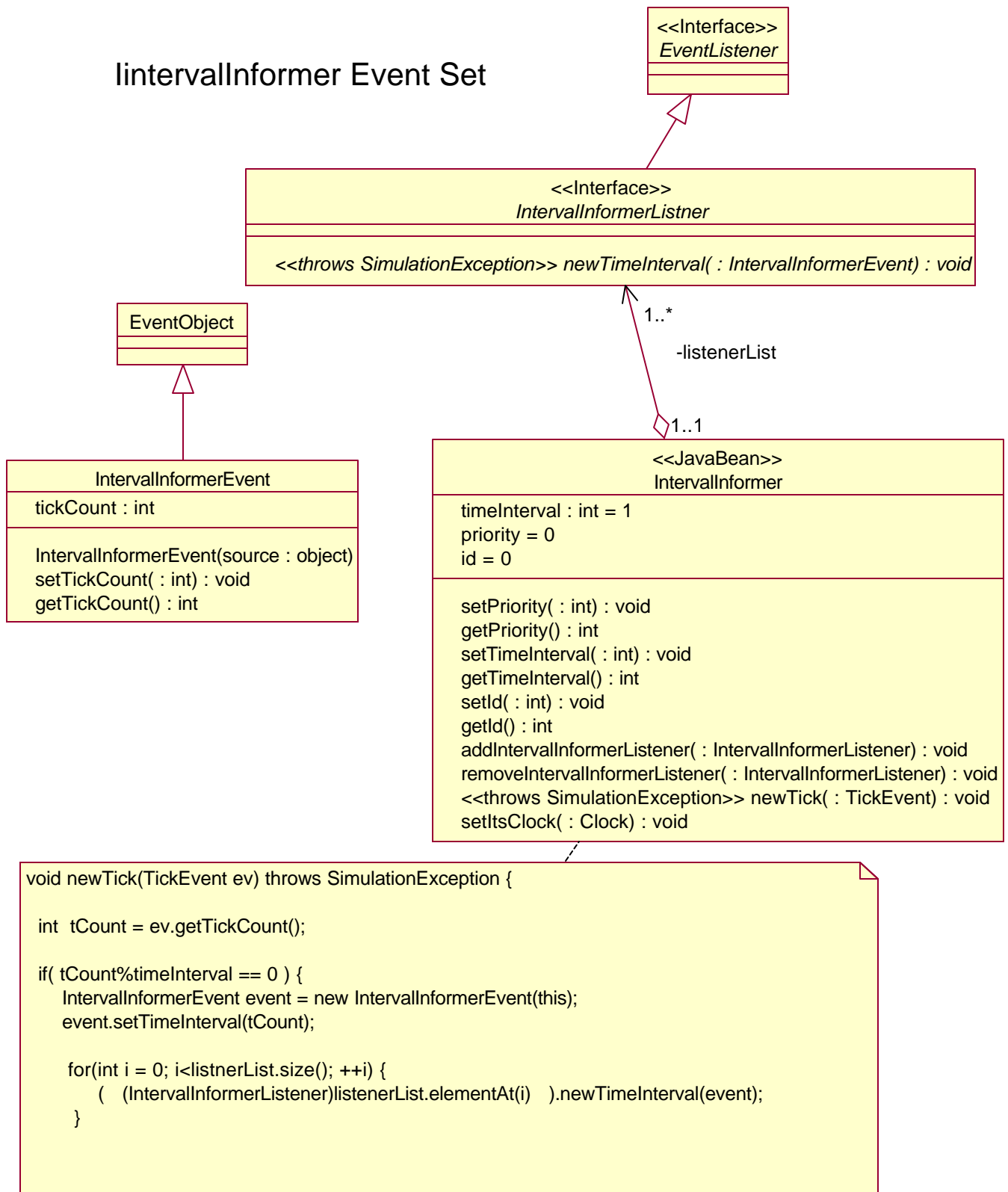


```

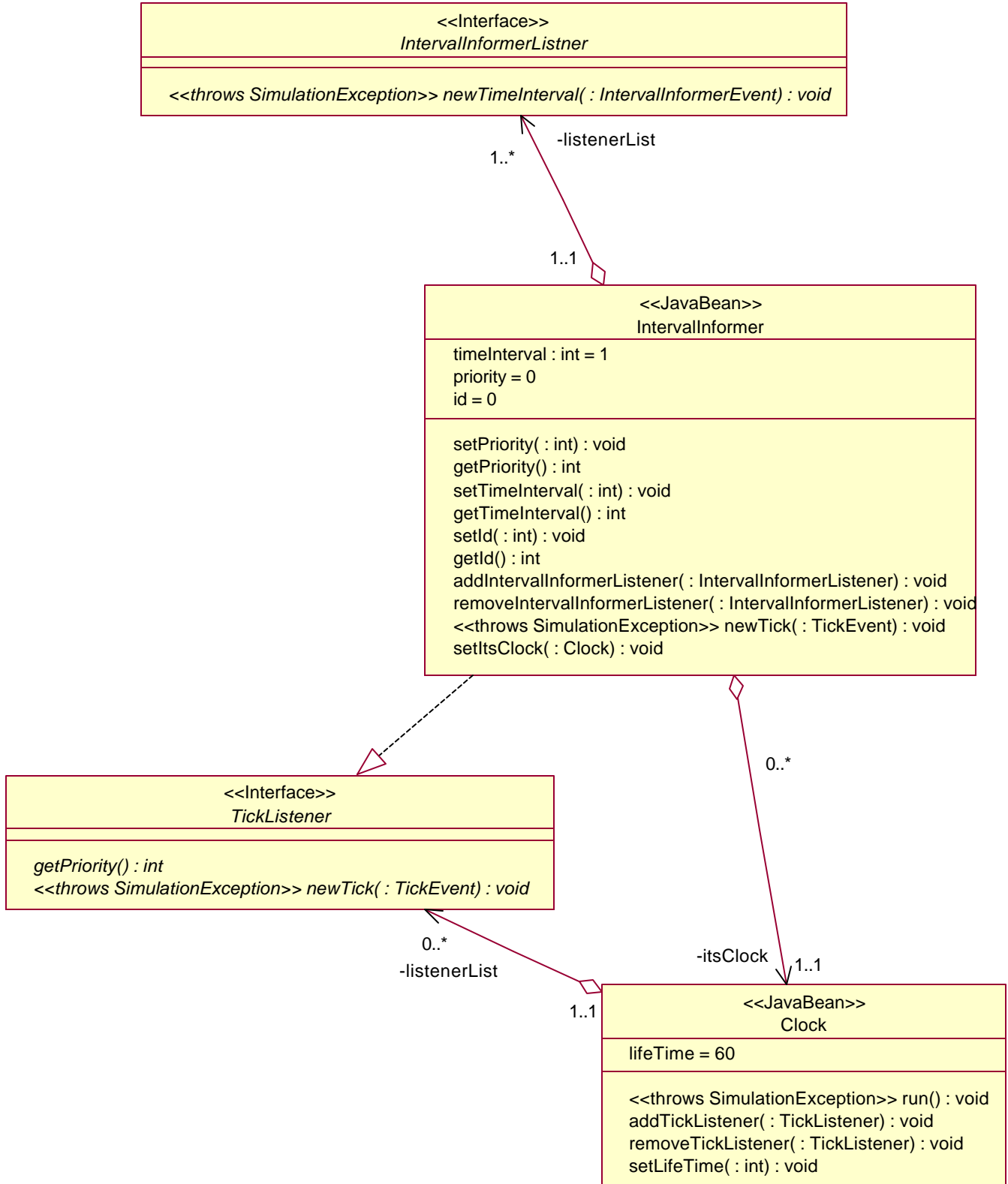
public void run() throws SimulationException {
    for(int i = 0; i<listenerList.size(); ++i) {
        invoke, priority = (TickListener)listenerList.elementAt(i) .getPriority(), and
        rearrange all the elements in the listenerList according to priority.
    }

    for(int t = 0; t < lifeTime; ++t) {
        TickEvent ev = new TickEvent(this);
        ev.setTickCount(t);
        for(int i = 0; i<listenerList.size(); ++i) {
            ( (TickListener)listenerList.elementAt(i) ).newTick(ev);
        }
    }
}
    
```

IntervallInformer Event Set



IntervallInformer listenes to Tick events



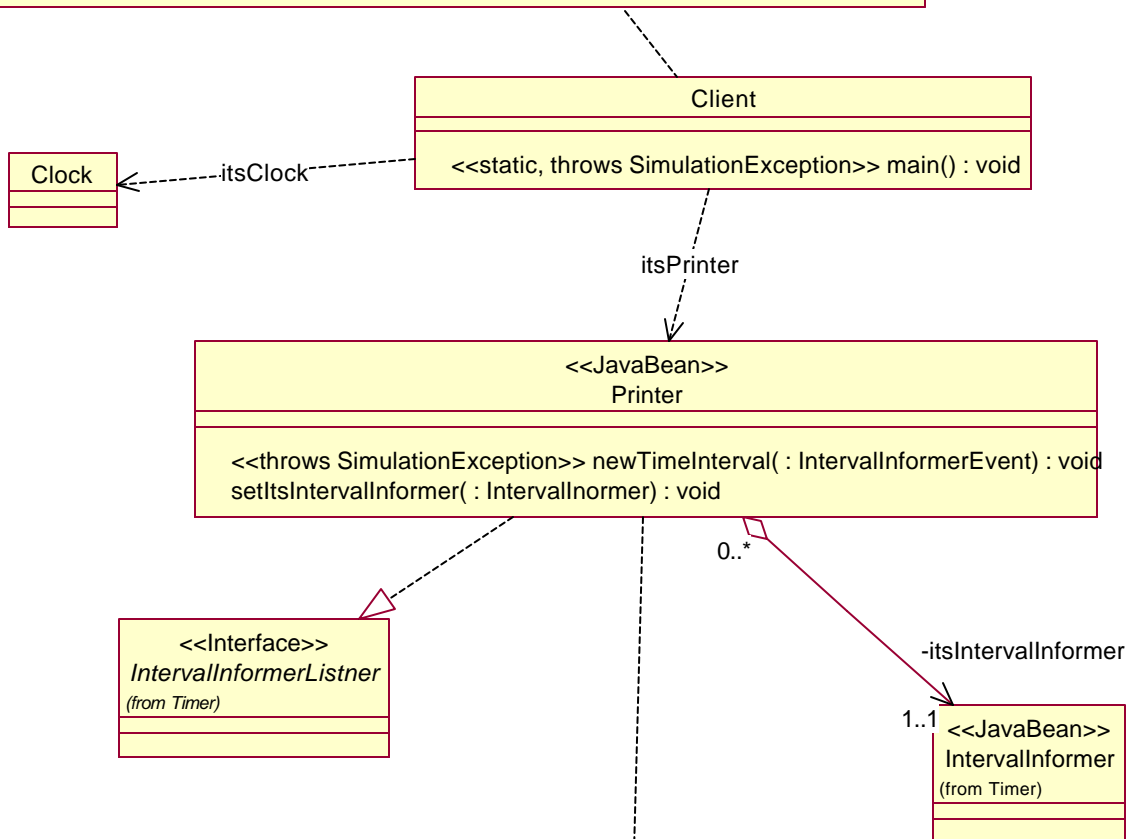
Client

```
public static main(String[] str) throws SimulationException {
```

```
    Construct a clock, with lifeTime set to 1200 seconds.
    Construct three Interval, with timeIntervals set to 4, 60, and 100 ticks respectively. Set
    their priority to 0.
    Connect the IntervalInformers to the clock, to listen to Tick events.
    Construct one printer
    Connect this printer to the three IntervalInformers to listen to IntervalInformer events
```

```
    itsClock.run();
```

```
}
```



```
void newTimeInterval(IntervalInformerEvent ev) throws SimulationException {
```

```
    int tickCount = ev.getTickCount();
    int ID = ( (IntervalInformer)ev.getSource()).getId();
    System.out.println("time: " + tickCount+" , ID: "+ ID);
```

```
}
```