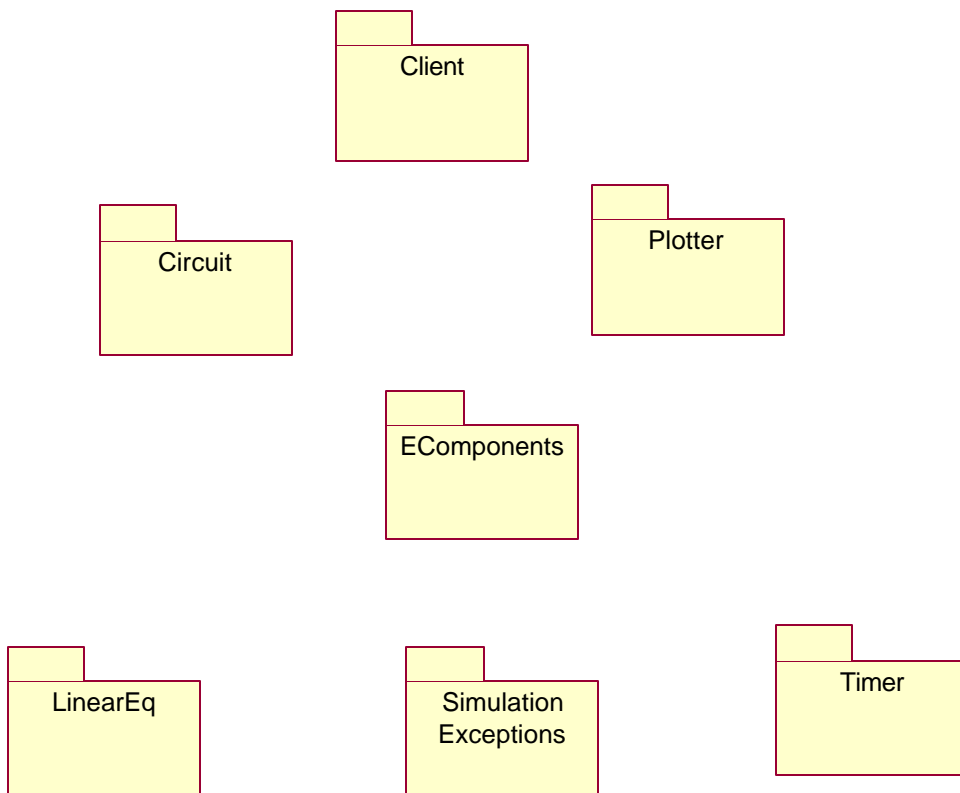


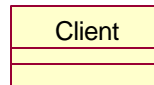
Project
EE/CIS 694T
Spring 2000

See the class diagram of the Client Package (next page) for details about this project

Packages



Package: Client



Client constructs the simulation system, and starts the simulation in its main() method.

All the JavaBean components, are constructed, and wired together in the main method as follows:

A single clock. Each tick is one second. LifeTime of the clock should be 200 seconds.

A series circuit consisting of two voltage sources, and four resistances, R1 = 10 ohm, R2 = 30 ohm, R3 = 5 ohm, and R4 = 20 ohm..

The voltage source and the series circuit listen to IntervallInformer events with timeInterval set to one.

The priority of the voltage source is 0 (highest priority)

The priority of the series circuit is 1 (lower than the souce priority)

One of the voltage sources, V1, is a triangular wave source with amplitude 5 volt, and a period of 50 seconds (phase = 0).

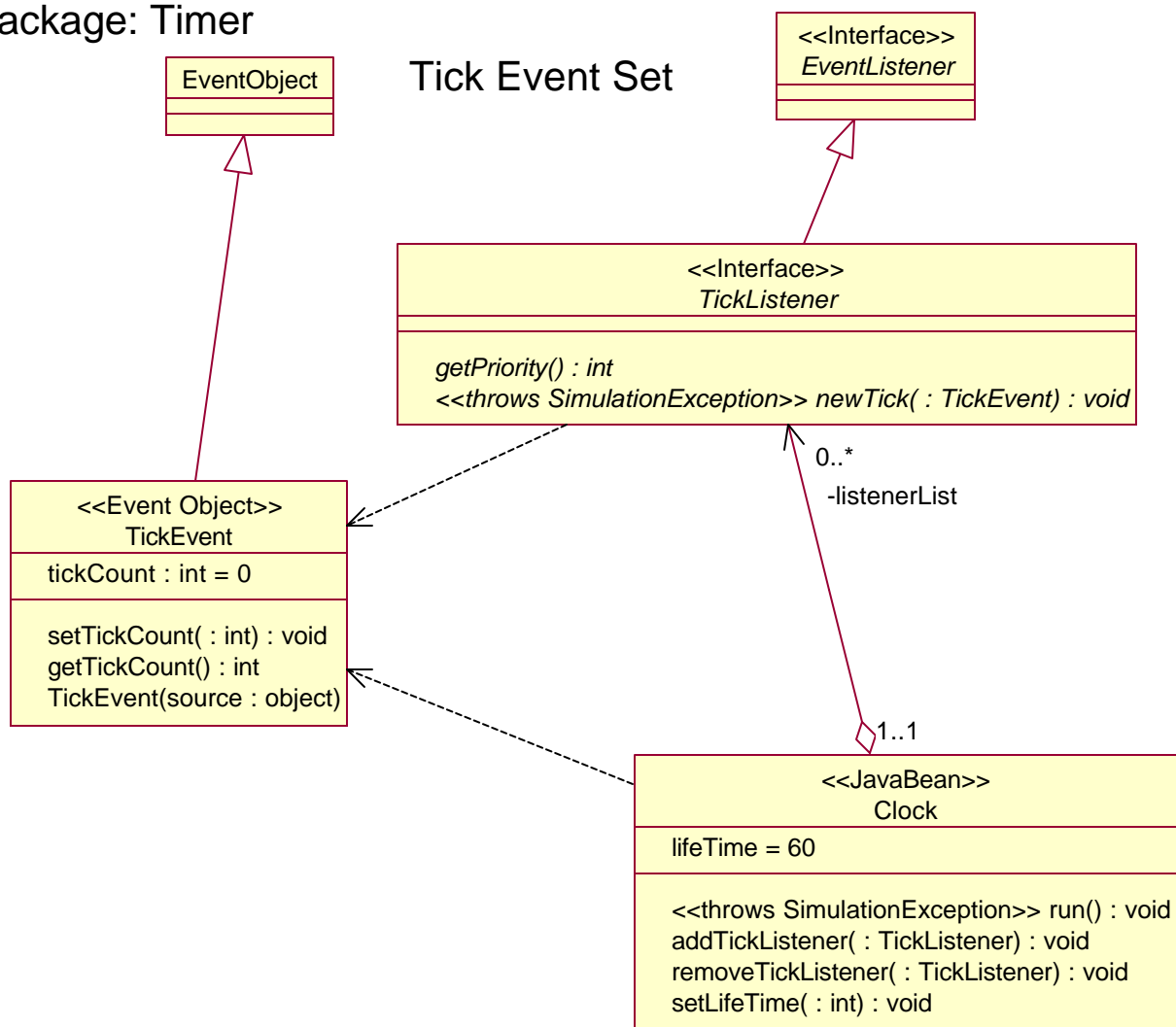
The other voltge source, V2, is a sin wave source, with an amplitude of 10 volts and a time period of 100 seconds (phase = 0).

A PlotToFile plotter, which should print value to a file named "plotterFile". This plotter is a n IntervalEventListener, and listens to timeInterval set to 2.

The Plotter contains a voltage sensor that is attached to the 10 ohm resistor, R1.

The simulation is started when method, run(), in the Clock is invoked, from the main() method of the Client.

Package: Timer



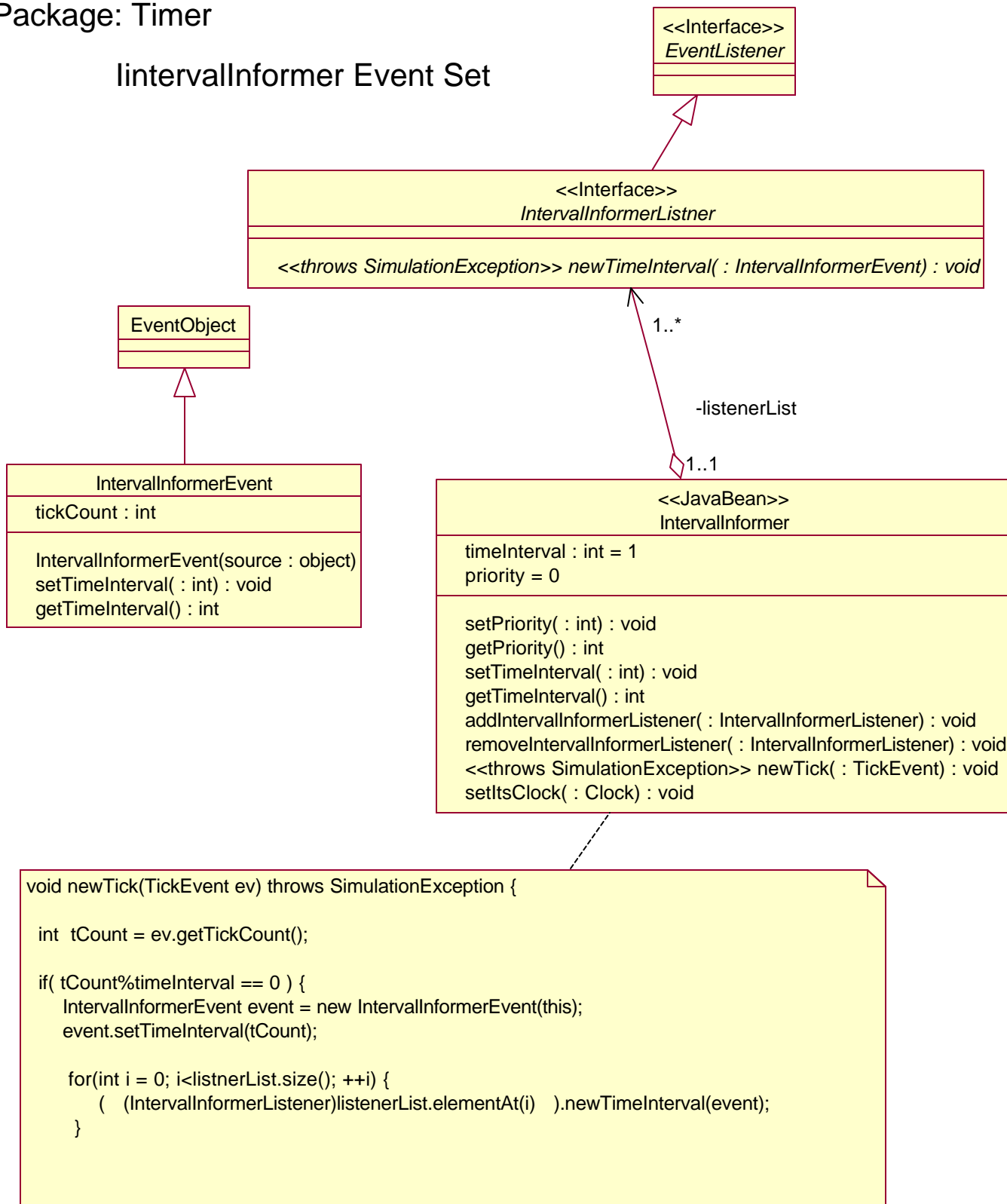
```

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);
        }
    }
}
  
```

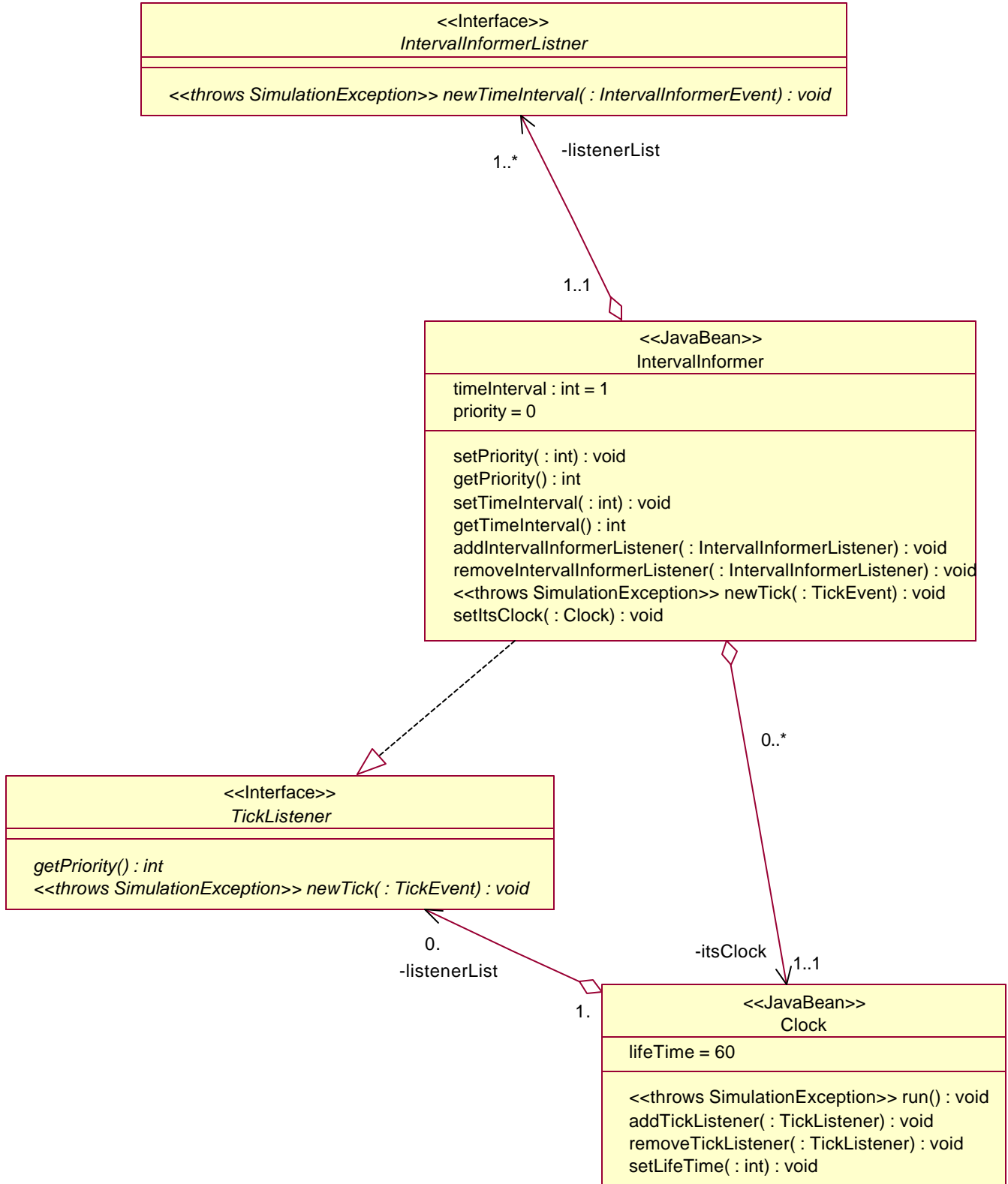
Package: Timer

IntervallInformer Event Set



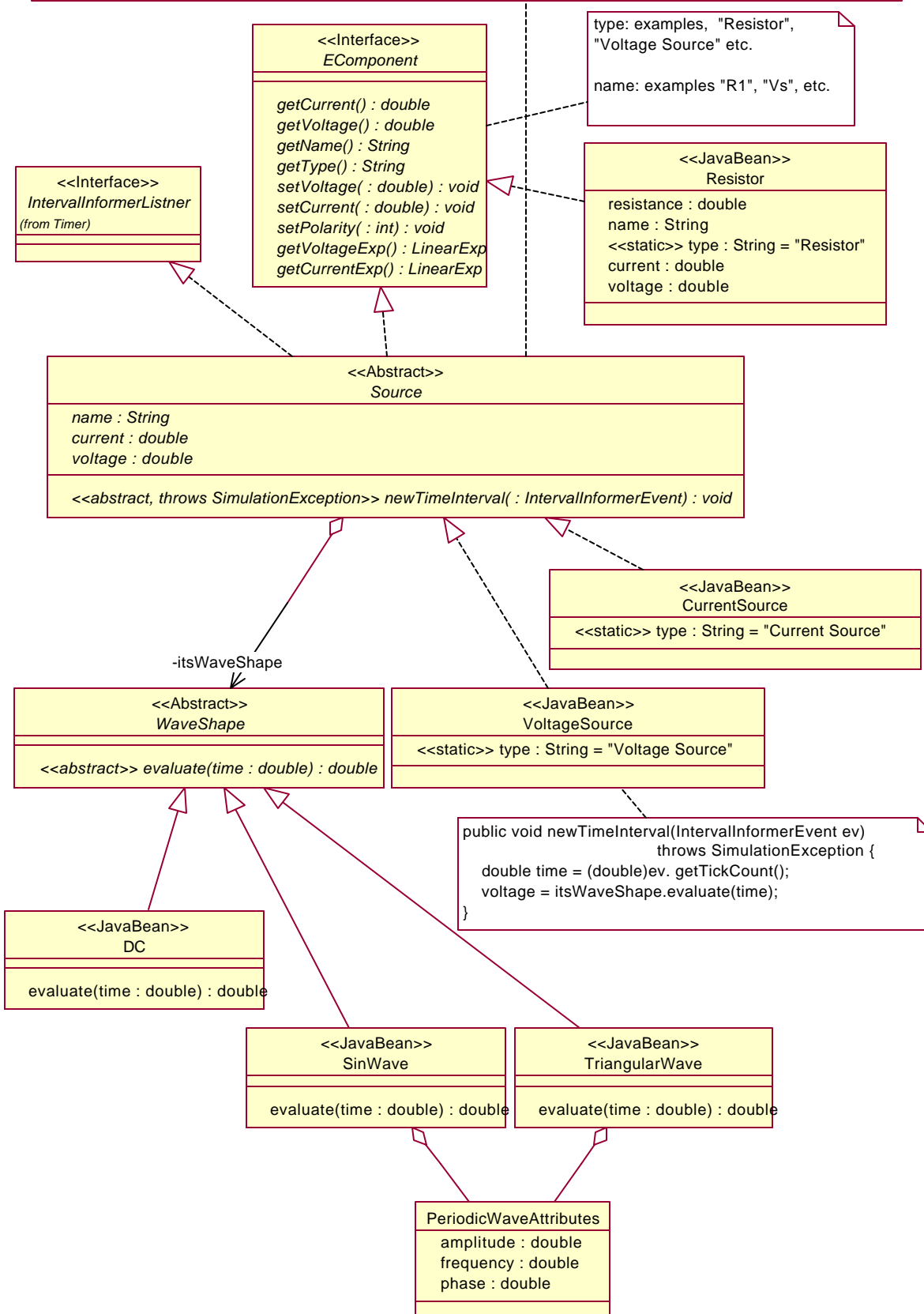
Package: Timer

IntervallInformer
listenes to Tick events



Package: EComponents

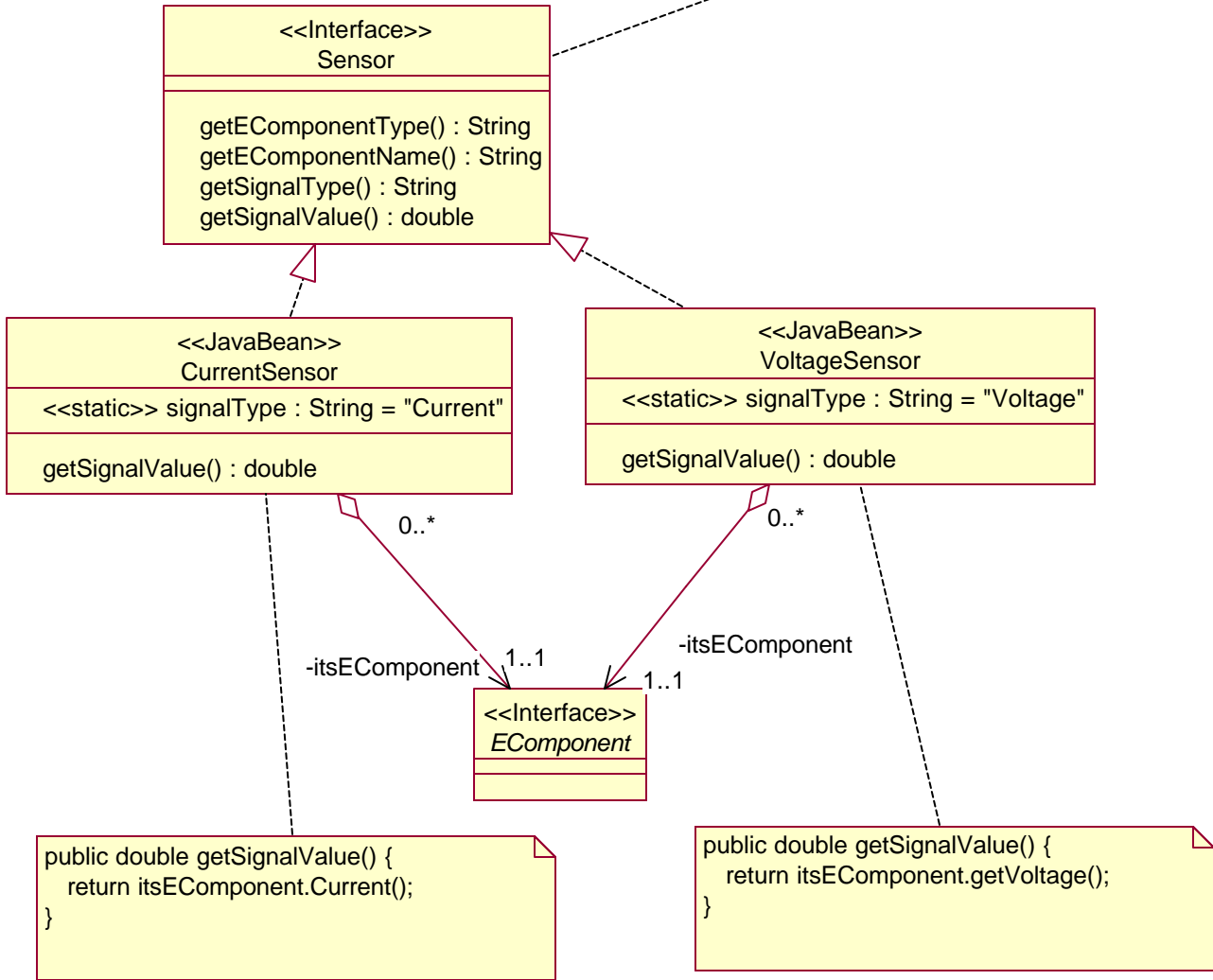
A Source registers itself with an IntervallInformer and listens to IntervallInformer Events. When it receives this event it asks its WaveShape to evaluate a wave shape value depending on the time it extracts from the event object, and sets its voltage or current to this evaluated value. The priority of this IntervallInformerListener should be the highest (higher than the priority of the Circuit, i.e., Sources should set their values before the Circuit solves itself).



Package: EComponents

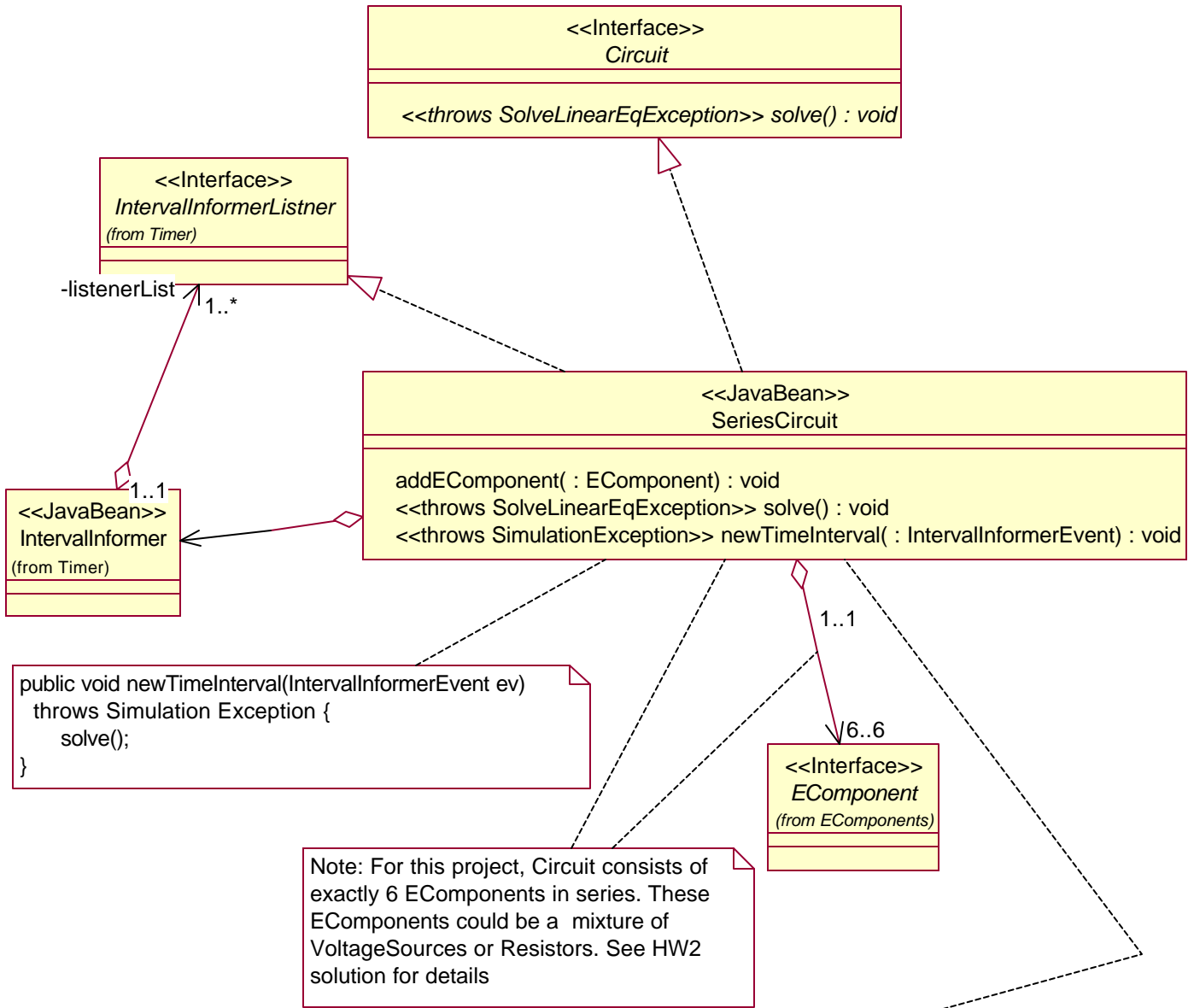
Sensors

EComponent Type: "Resistor", "Voltage Source", etc.
EComponent Name: "R1", "Vs", etc.
SignalType: "Voltage", "Current"
SignalValue: voltage or current value of the EComponent



Package: Circuit

See HW2 solutions for details



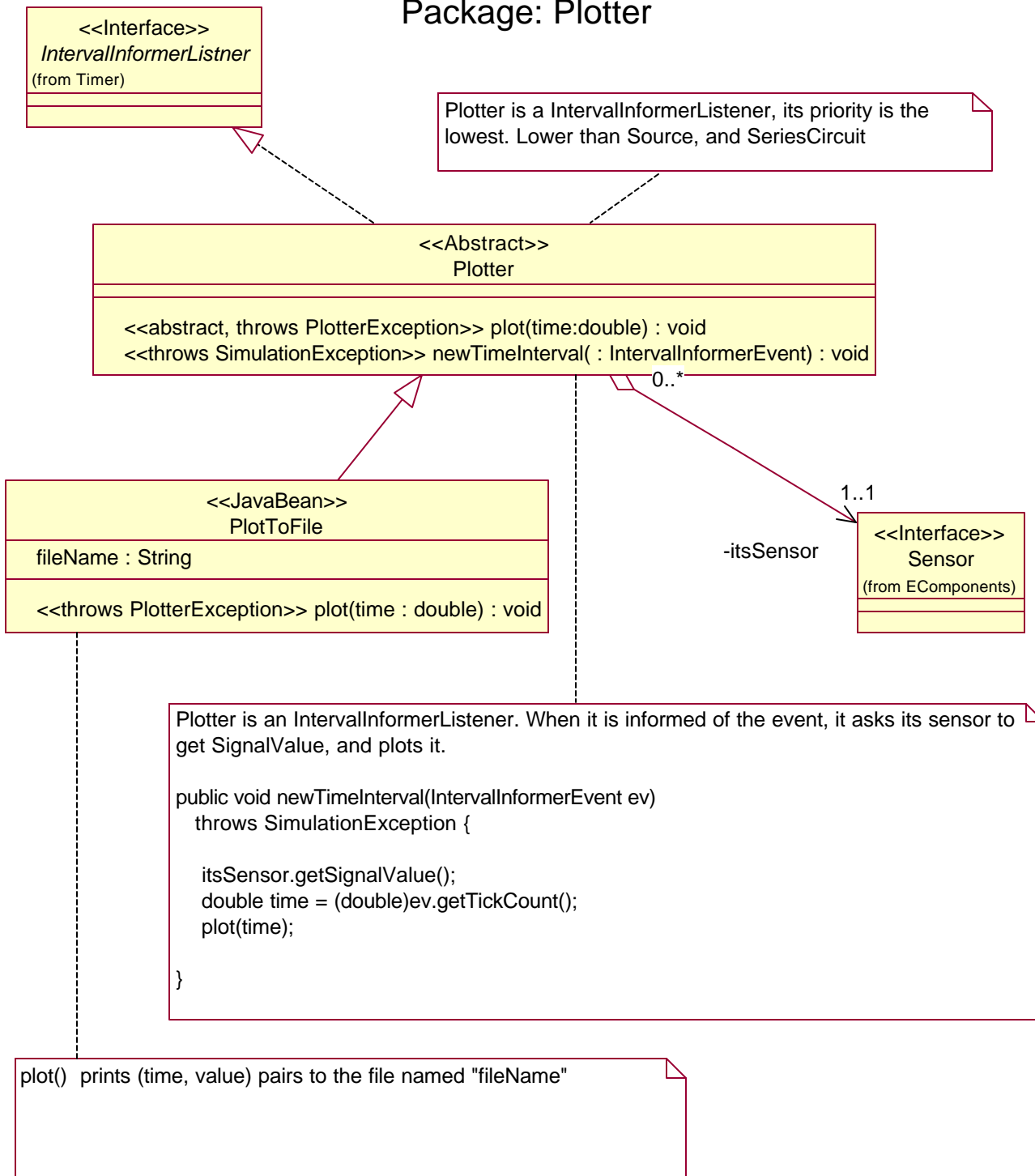
```

public void newTimeInterval(IntervalInformerEvent ev)
throws SimulationException {
    solve();
}
    
```

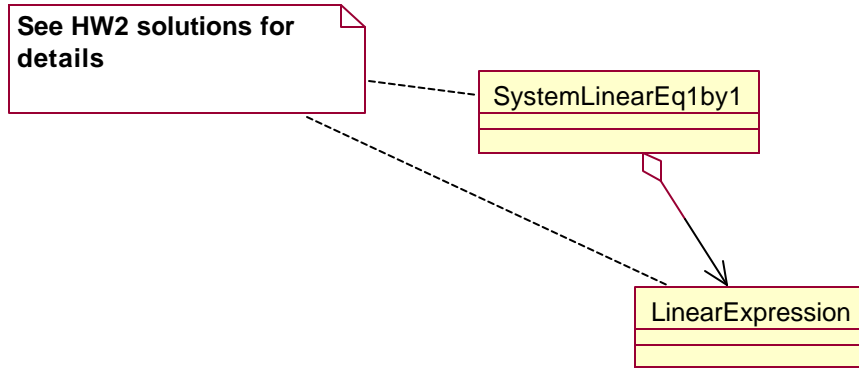
Note: For this project, Circuit consists of exactly 6 EComponents in series. These EComponents could be a mixture of VoltageSources or Resistors. See HW2 solution for details

The SeriesCircuit registers itself with an IntervalInformer and listens to IntervalInformer Events. When it receives this event it solves the circuit, and sets the voltage and current values of all the EComponents in it. The priority of this IntervalInformerListener should be lower than the priority of a Source, which is also an IntervalInformerListener. i.e. sources should set their values before the circuit solves itself.

Package: Plotter



Package: LinearEq



Exceptions

