



Testing From State Machines

Good Afternoon.

Lecture Objectives:

- 1) Define the concept of a finite state machine.
- 2) Define the terms state, transition, event, and action. *→ Review*
- 3) Explain the concept of a state transition table.
- 4) Given a state diagram, construct a state transition table for the problem.
- 5) Explain how state transition test cases can be constructed from a state machine definition.
- 6) Construct a set of test cases from a state diagram.
- 7) Implement a system which tests a state machine automatically using JUnit.

Lab

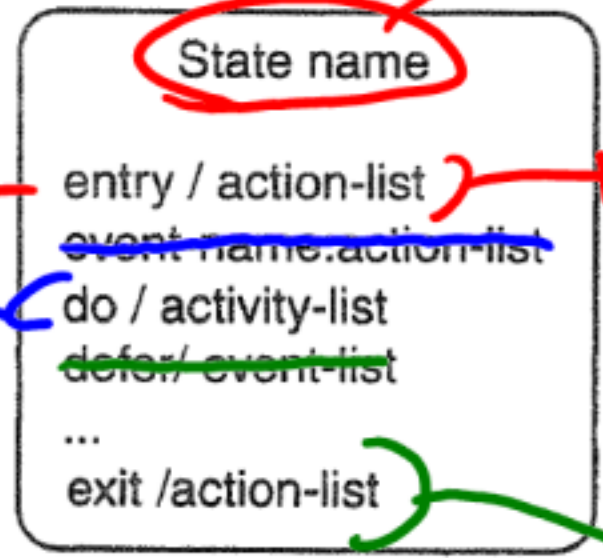
Testing

Definitions

- Finite State Machine (FSM)
 - A Finite State Machine is a graph with nodes that represent states and edges that represent transitions
- UML State chart
 - The UML mechanism for representing finite state machines.


in UML

Light ON



Action which occurs when we enter a state.

turn fan on()

State Chart

State Icon

Things that occur while one is in a state. Monitor bulb temperature) Typically spawn thread.

Actions we take when we leave a state.

State Transitions

Cause vs to, change states.



Boolean Condition must Evaluate to TRUE for the Transition to be Taken

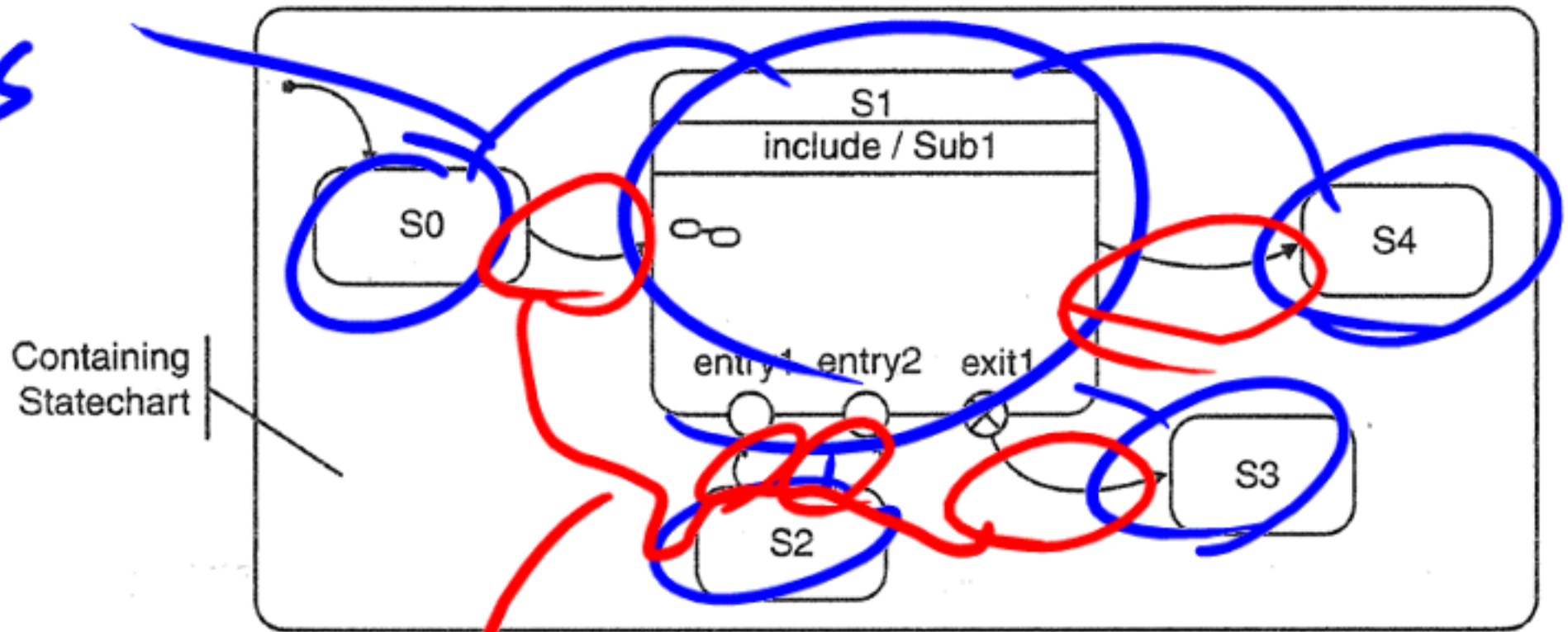
Transitions

Limit transitions based on logic

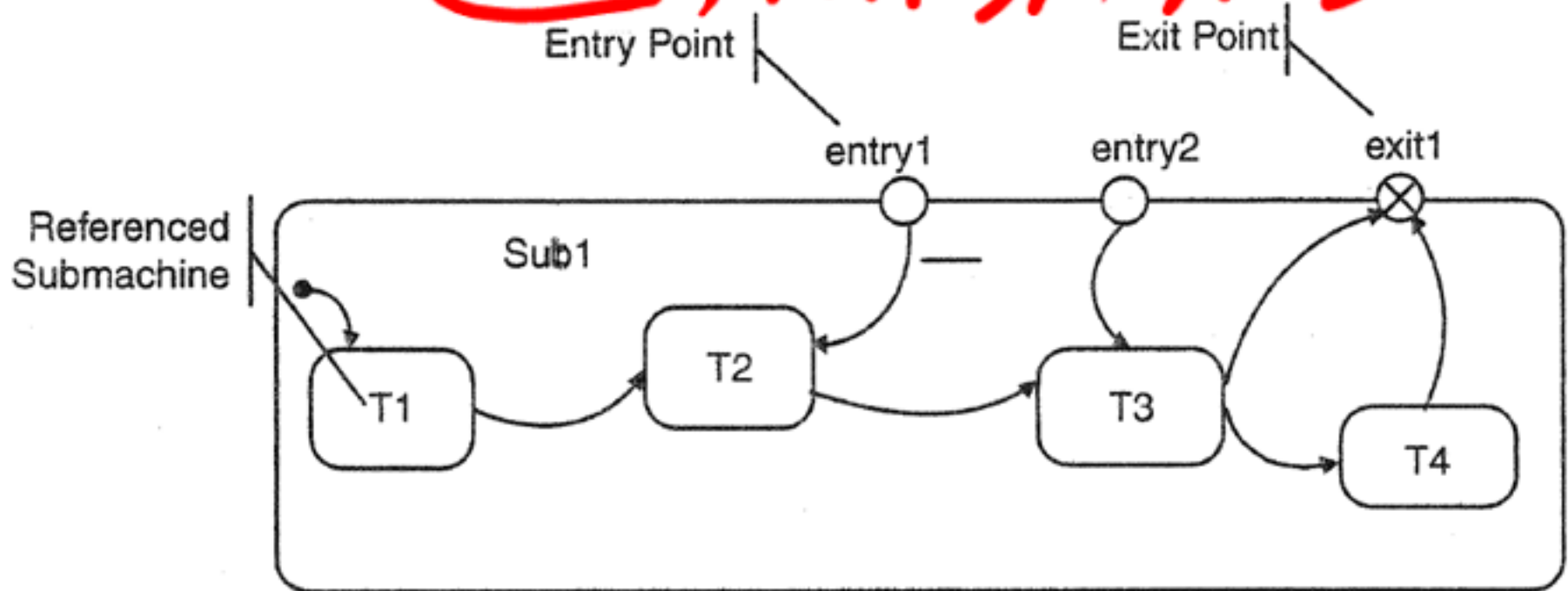
methods invoked on a transition

5 states

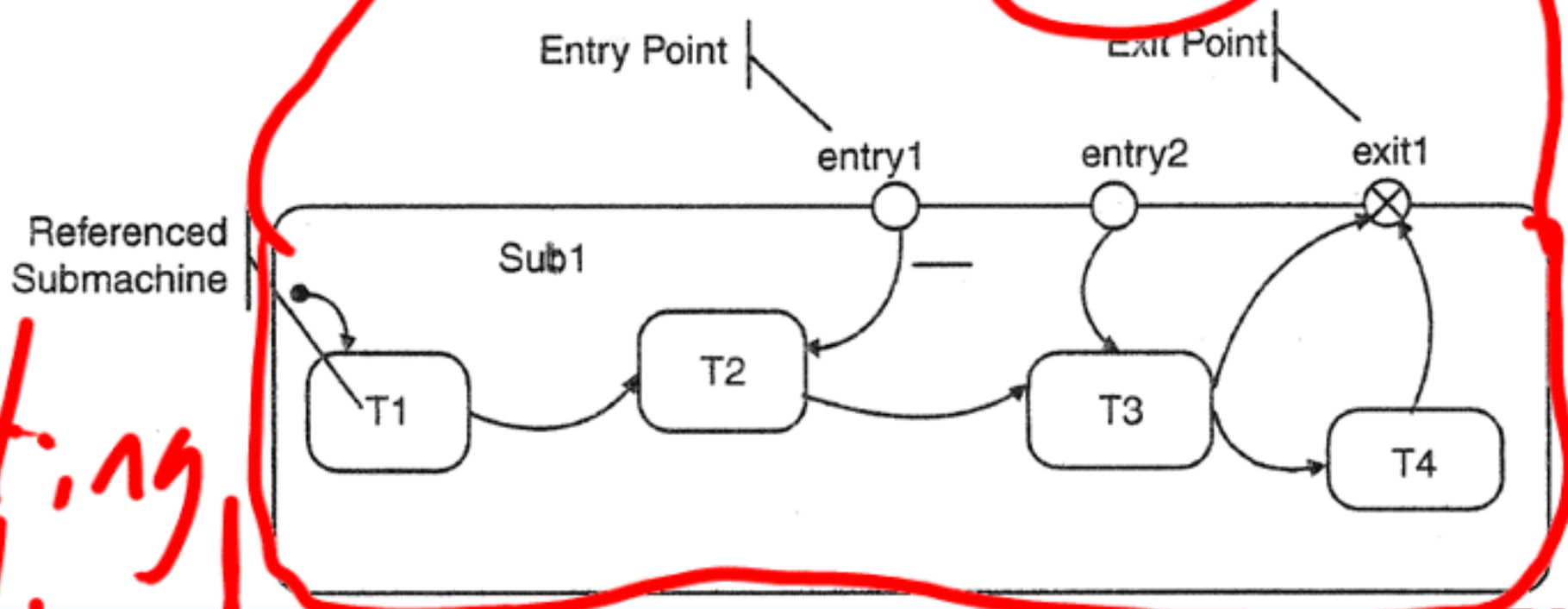
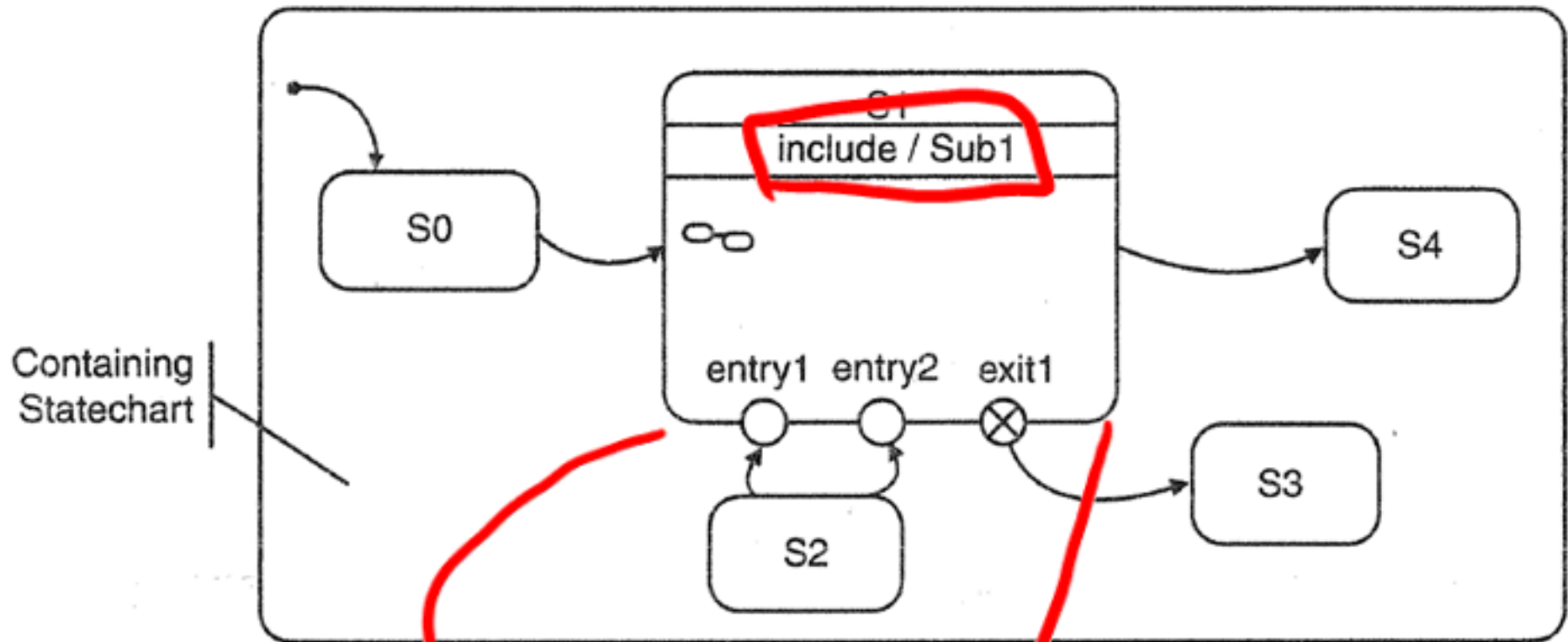
State Charts



Transitions

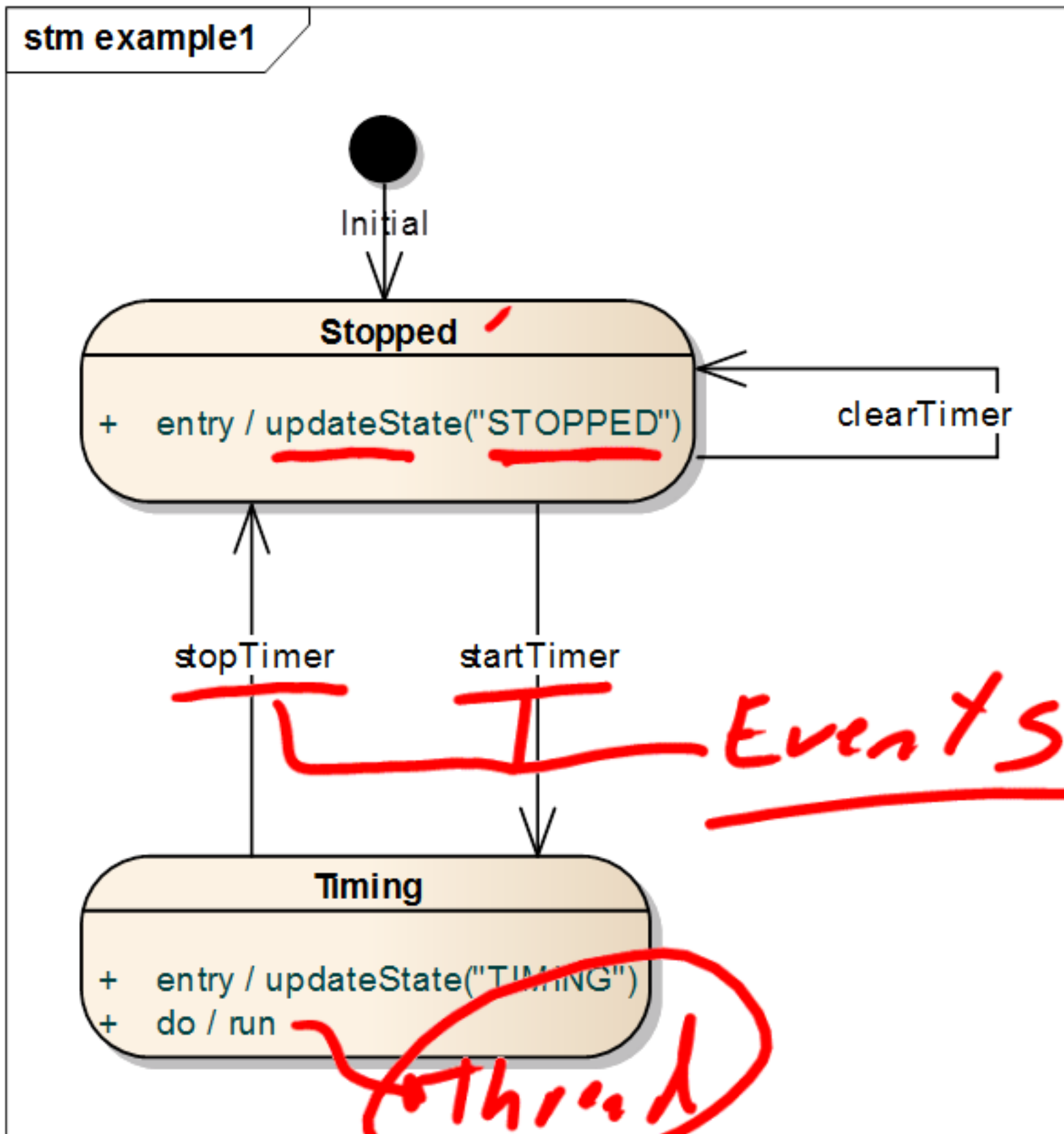


State Charts



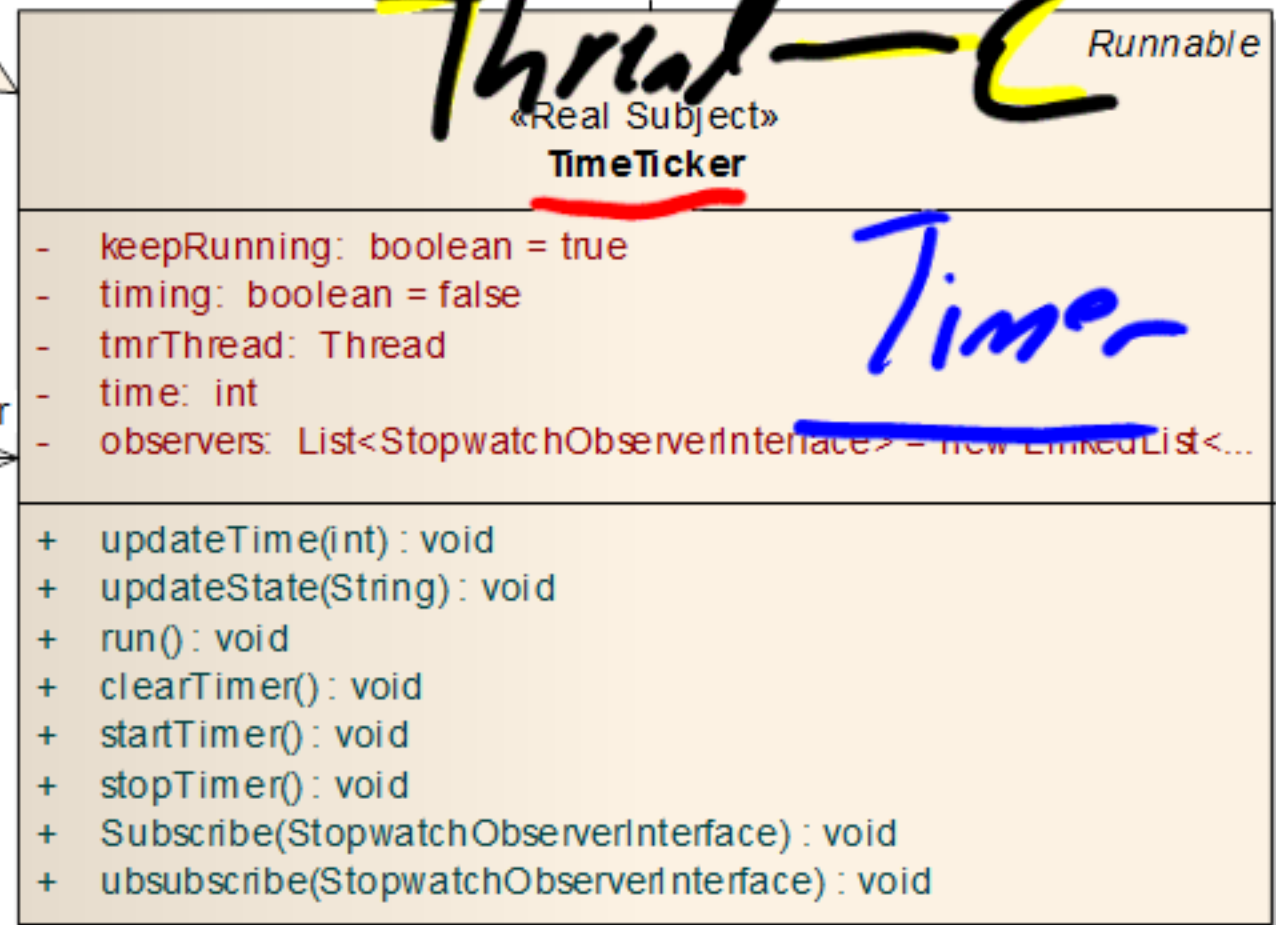
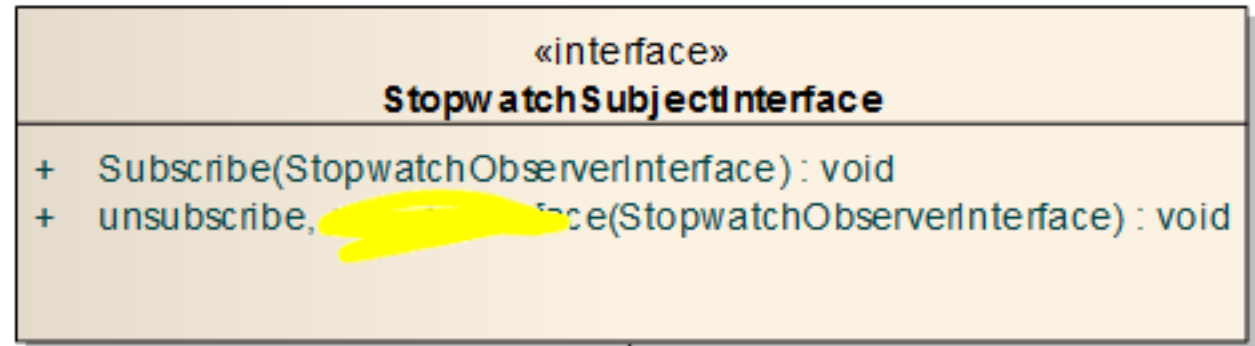
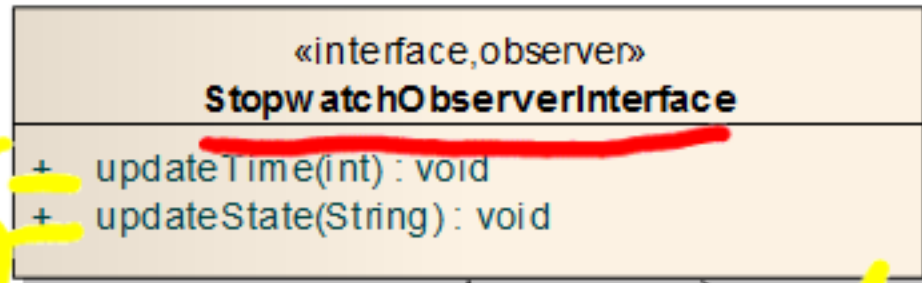
Nesting is allowed

A simple stopwatch



Observer Pattern

class example1



observer relation

Thread - Runnable

Timer

State Table

State	Event	Guard Condition	Destination State
Stopped	clear_time	—	Stopped
	start	—	Timing
Timing	stop_timer	—	Stopped

Method invoked

update state (STOPPED)
update state (Timing)

update state (STOPPED)



Lets take a look at the code

Testing