**Key: Model the behavior of each element and each activity of the element (event) that occurs**

Classes Needed

* Stop.java (Bus Stops)
  + For each stop in the route (new instance of Stop)
  + 2 Queues (Q.java)
    - Eastbound
    - Westbound
  + Name of Stop (String)
  + Bus Stops (array) (immutable)
    - University Ave and 27th Street SE: 0
    - Raymond Ave Station: 1
    - University Ave and Fairview Ave: 2
    - University Ave and Snelling Ave: 3
    - University Ave and Lexington Parkway: 4
    - University Ave and Dale Street: 5
    - University Ave and Marion Street: 6
    - **Cedar Street and 5th Street**
    - **Minnesota Street and 4th Street**
    - **Union Depot**
* PQ.java (Event) (Already Provided) (Use Lecture Code and Cite)
  + One Instance (main agenda)
* Event.java (Interface with run() method)
* BusEvent.java (Implements Event and occurs each time a bus arrives at a stop)
  + New instance for every arrival of bus
  + Contains a bus
    - Passenger list
    - Checks each passenger for it’s destination
      * If it’s destination matches the current stop, it removes that passenger (look at lab!)
    - Checks Stop.java’s queues, get the passengers and their destinations
      * If they match, they add to the bus
  + Creates a new BusEvent and schedules it
    - Arrival at the next stop at a time in the future
      * Add time for each passenger that got on + got off
  + If Bus has reached last stop (e.g stop[currentstop] == 9), turn it around (boolean direction = false (vice-versa)
* PassengerEvent.java (Implements Event and occurs each time a Passenger arrives at a stop)
  + Random Distribution of Arrivals
    - Standard: 1 every 120 seconds
    - Look at lab sheet for rest of distributions
    - Downtown stops are more active (50% more often)
  + Made for each stop
  + Rank stops within PassengerEvent (NumPassengers?)
  + Reschedules itself (Look at lab!)
  + Creates a passenger and decide these attributes🡪 Have a Passenger.java class
    - Destination (int) (randomly generated)
      * 2 times as likely for a downtown stop
    - Direction (boolean)
    - Which Queue? (See lab!)
* Statistics.java
  + # of passengers in each bus (int)
  + Maximum time a passenger spends waiting at a stop (update max time each time by comparing each passengers wait time to the current max time)
  + Maximum queue length at a stop (update each max length by comparing each queue’s length at each BusEvent when it checks the queues)
  + Tbc…
* Timing
  + 180 seconds between each stop
  + 15 second wait between each stop
  + 2 seconds to get on and 3 seconds to get off
* Other variables
  + 18 busses
  + Average inter-arrival rate
  + Bus size