Doreen Hakimi
CS201 Restaurant V2.1
Fall 2013

DATA

```
List<Customer> waitingCustomers
List<Table> tables

class Table {
        int tableNum;
        Customer c= null;
}

List<Waiter> waiters

Customer custLeavingWaitlist=null;
Customer sendFullMsgTo=null;
int customersInRST=0;

boolean KitchenReadyForOpen=false;
```

MESSAGES

```
KitchenIsReady() {
        KitchenReadyForOpen=true;
}
IWantFood(Customer c) {
        if(customersInRst>=NTABLES)
                sendFullMsgTo = cust;
        waitingCustomres.add(c);
}

IDontWantToWait(Customer cust) {
        if there exists CustomerAgent c in waitingCustomers such that c=cust
                then custLeavingWaitliste=cust;
}

AddWaiter(Waiter w) {
        waiters.add(w);
}

TableIsClear(Table t) {
        if there exists Table t1 in tables such that t1=t
                then t.c=null;
}
```

```
GoOnBreakPlease(Waiter w) {
        if there exists Waiter wa such that wa = w
                then wa.requestedBreak=true;
}

BackToWork(Waiter w) {
        //stateChanged();
}
```

SCHEDULER

```
if(cusLeavingWaitlist!=null)
        then RemoveCustomerFromList();
if(sendFullMsgTo!=null)
        then RestaurantIsFull();
if (!waiters.empty())
        waiter=w in waiters with minimum customers.size() value with w.isOnBreak=false
        if there exists table t in tables such that t.c==null
                then AssignToTable(waitingCustomers.get(0), waiter, table)
        if there exists Waiter wa in waiters such that wa.requestedBreak=true
                then AnswerWaiterBreakRequest(wa)
```

ACTIONS

```
RestaurantIsFull() {
        sendFullMsgTo.NoRoomForYou();
        sendFullMsgTo=null;
}

RemoveCustomerFromList() {
        waitingCustomers.remove(custLeavingWaitlist);
        custLeavingWaitlist=null;
}

AnswerWaiterBreakRequest(Waiter w)
        int waitersOnDuty=0;
        if there exists a waiter in waiters such that !waiter.onBreak;
                waiters.OnDuty++;
        if(waitersOnDuty>1) {
                w.BreakReply(true);
        }
        else
```

```
                w.BreakReply(false);
}

AssignToTable(Customer cust, Table t, Waiter w) {
        t.occupiedBy=cust;
        waitingCustomers.remove(cust);
        w.SitAtTable(t, cust);
}

AnswerWaiterBreakRequest(Waiter wa) {
        if (waiters.size()>1)
                then wa.BreakReply(true);
                        wa.requestedBreak=false;
        else
                wa.BreakReply(false);
                wa.requestedBreak=false;
}
```