DATA

```
List<MyCheck> checks;

Menu menu;

Timer timer;

class Check {
        Customer cust;
        string choice;
        double amount;

        Cashier cashier;
}

class MyCheck {
        Waiter w;
        Check c;
        checkState state = requested;
}

enum checkState { requested, givenToWaiter, fullyPaid, partiallyPaid, finished  };

List<MyCustomer> customersWhoOweMoney;

class MyCustomer {
        Customer cust;
        double amountOwed;
}
```

MESSAGES

```
ProduceCheck(Waiter w, Customer c, string choice) {

        Check check = new Check(c, choice);
        if c is in customersWhoOweMoney
                then MyCustomer c = customersWhoOweMoney.find(c);
                and check.amount += c.amountOwed;
        checks.add(new MyCheck(w, check));
}
```

```
PayBill(Check check, double money) {
    MyCheck c = checks.find(check);

    if money > c.amount
        then c.state = fullyPaid;
    else if money < c.amount
        amount -= money;
        then c.state = partiallyPaid;
}
```

SCHEDULER

```
if there is a c in checks such that c.state = fullyPaid
    then thankCustomer(c);
if there is a c in checks such that s.state = partiallyPaid
    then addCustomerToOweList(c);
if there is a c in checks such that c.state = requested;
    then giveCheckToWaiter(c);
```

ACTIONS

```
giveCheckToWaiter(MyCheck c) {
    c.w.HereIsCheck(c.c);
    c.state = givenToWaiter;
}

giveChange(MyCheck c) {
    c.state = finished;
}

addCustomerToOweList(MyCheck c) {
    if c.cust is in customersWhoOweMoney
        then add c.amount to c.cust.amountOwed
    else
        customersWhoOweMoney.add(new MyCustomer(c.c.cust, c.c.amount));
    c.state = finished;
}
```