Owner Role

List<Person> tenants;   [DATA]

---

msgHereIsRent(Tenant t, Money m);   [MSG]
msgApplianceBroken(Tenant t, Money m);

---

if ∃ p in tenants ∈ p.owesRent
    collectRent(p);   [SCHED]
if ∃ a in appliances ∈ a.broken
    fixAppliance(a);

```
collectRent(p) {
    p.msgPayRent();                       [ACTIONS]
    p.state = notified;
}
fixAppliance(a) {
    Timer fix
}
```

Tenant Role

float rentOwed;   enum State {
Person owner;   Hungry, Tired
List<Appliance> appliances;   [DATA]

---

msgPayRent(Money m)   [MSG]

---

if rentOwed ≠ 0
    tryToPayRent()
if state = Hungry   [SCHED]
    eat()
if state = Tired
    sleep()

```
eat() {                      sleep() {
    make food                    use bed
    or go out                    sleep
}                            }
tryToPayRent {
    if cash, pay             [ACTIONS]
}
```