Paper Solution

(a)

types

Mode = $<$working$>$ | $<$idle$>$ | $<$broken$>$

(b)

state IndustrialRobot of
initialMode : Mode

inv mk-InitialRobot (m) $\triangleq$ m = $<$idle$>$ v $<$broken$>$ v $<$working$>$

init mk-IndustrialRobot (m) $\triangleq$ m = $<$idle$>$

end

(c)

Operations

setMode (val : Mode)
ext wr initialMode : Mode
pre True
post initialMode = $<$val$>$

getMode () output : Mode
  ext rd initial Mode : Mode
  pre True
  post   output = initialMode

isIdle () query : B
ext rd initialMode : Mode
pre True
post   query $\iff$ initialMode = <idle>

Question #2

(a)

[Airport] [Aircraft]

| MAX : $\mathbb{Z}$ |

__Airport_____
Permission : $\mathbb{P}$ Aircraft
landed : $\mathbb{P}$ Aircraft
―――――――――――――――
#landed < MAX
landed $\subseteq$ Permission

__InitAirport_____
Airport
―――――――――――――――
Permission = $\emptyset$
landed = $\emptyset$

(b)

```
┌─ givePermission ──────────
│ Δ Airport
│ a? : Aircraft
├──────────────────────────
│ a? ∉ Permission
│ Permission' = Permission ∪ {a?}
│ landed' = landed
└──────────────────────────
```

```
┌─ record landing ──────────
│ Δ Airport
│ l? : Aircraft
├──────────────────────────
│ l? ∈ Permission
│ l? ∉ landed
│ # landed < MAX
│ landed' = landed ∪ {l?}
│ Permission' = permission
└──────────────────────────
```

```
┌─ numberWaiting ──────────
│ Ξ Airport
│ w! : ~~Aircraft~~ Z
├──────────────────────────
│ w! = # (Permission \ landed)
└──────────────────────────
```

Question #3

types

values

    $Limit : Z$

state CarParking of

    $carsCount : Z$

inv $mk-CarParking(r) \triangleq r \leq Limit$

init $mk-CarParking(r) \triangleq r = 0$

end

operations

  Enter ()

ext wr $carsCount : Z$

  pre $carsCount < Limit$

  post $carsCount = carsCount + 1$

  Depart ()

ext wr $carsCount : Z$

  pre $carsCount \geq 0$

  post $carsCount = carsCount - 1$

QuerySpace() space : 2
ent rd CarsCount : 2
pre True
post space = Limit — carsCount

$\longleftrightarrow$

Question #4

[Patient]

| Limit : N

PatientRegister ─────
reg : $\mathbb{P}$ Patient

reg ≤ Limit

Init PatientRegister ─────
PatientRegister

reg = $\emptyset$

add Patient

$$\Delta \text{Patient Register}$$
$$P? : Patient$$

$$P? \notin reg$$
$$\#reg < Limit$$
$$reg' = reg \cup \{P?\}$$

remove Patient

$$\Delta \text{Patient Register}$$
$$P? : Patient$$

$$P? \in reg$$
$$reg' = reg \setminus \{P?\}$$

get Patient

$$\Xi \text{Patient Register}$$
$$total! = \mathbb{P}\, Patient$$

$$total! = reg$$

```
┌─ numberRegistered ──────────────
│ Ξ Patient Register
│ count! : Z
├──────────────────────────────────
│ count! = # reg
└──────────────────────────────────
```

$\longleftrightarrow$

Question #5

```
┌─── Add Room ──────
│ Δ Hotel
│ r? : ROOM
├────────────────────
│ # room < Limit
│ r? ∉ dom room
│ room' = room ∪ {r? ↦ Vacant}
└────────────────────
```

```
┌─── Occupy Room ──────
│ Δ Hotel
│ r? : ROOM
├───────────────────────
│ r? ∈ dom room
│ room(r?) = Vacant
│ room' = room ⊕ {r? ↦ Occupy}
└───────────────────────
```

VacantRoom

Δ Hotel

$r?$: ROOM

$r? \in dom\ room$

$room(r?) = Occupy$

$room' = room \oplus \{r? \rightarrow Vacant\}$