

Traccia di soluzione dell'esercizio del 28 Giugno 2004

1. Modello del sistema

Il casello aggrega almeno una porta di ingresso e una porta di uscita. Il modello del casello è in Figura 1

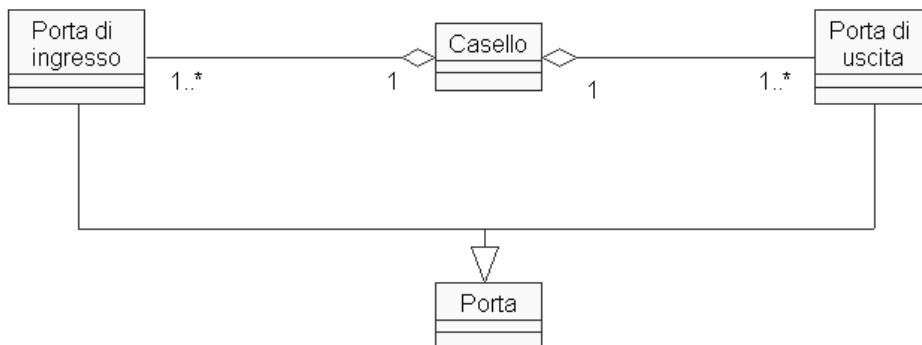


Figura 1: Diagramma UML: Struttura del casello

In Figura 2 viene dato il diagramma a oggetti delle porte di entrata e di uscita. Le relazioni sono di appartenenza dei subordinati alle porte.



Figura 2: Relazioni di appartenenza tra gli oggetti che compongono le porte

Sebbene il testo non specifichi niente al riguardo e si possa assumere che le porte gestiscono direttamente i dispositivi fisici (Pulsanti, Sbarre, LettBiglietto, ecc.), conviene immaginare (come in Figura 2) che a questi ultimi corrispondano oggetti software responsabili della loro gestione. In tal modo l'interazione tra le porte e i dispositivi fisici è mediata da questi oggetti e si limita all'invocazione dei metodi di interfaccia che essi presentano.

2. Porta di ingresso

Il diagramma di stato della porta di ingresso è in Figura 3.¹

La porta di ingresso presenta questi metodi di interfaccia (a parte il costruttore):

- Attiva() per attivare la porta; (porta la luce a verde);
- Disattiva() per disattivarla; (porta la luce a rosso);
- tick() per fare un passo di controllo degli oggetti sottostanti

3. Porta di uscita

Il diagramma di stato della porta di uscita è in Figura 4

La porta di uscita presenta questi metodi di interfaccia (a parte il costruttore):

¹Si è riportata tra parentesi la condizione che fa tornare allo stato di Idle, in quanto non è specificato che ci sia un sensore di chiusura della sbarra. In fase di implementazione l'oggetto che controlla la sbarra può essere fatto in vario modo. In particolare potremmo immaginare che emette un comando ad un motore e che questo opera in modo automatico fino alla chiusura della sbarra. In ogni caso l'oggetto Sbarra avrà il metodo St() che riporta lo stato della sbarra.

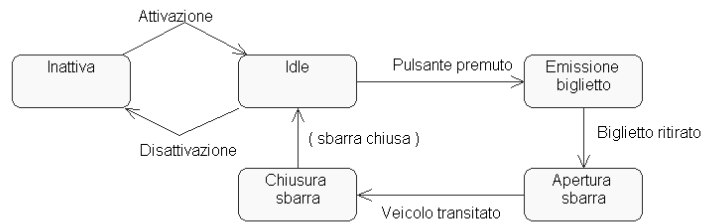


Figura 3: Diagramma di stato dell'oggetto porta di ingresso

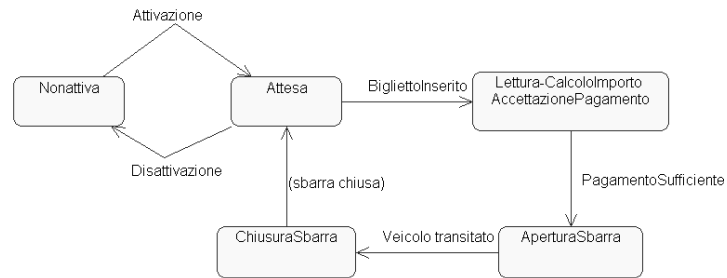


Figura 4: Diagramma di stato dell'oggetto porta di uscita.

- `Attiva()` per attivare la porta; (porta la luce a verde);
- `Disattiva()` per disattivarla; (porta la luce a rosso);
- `tick()` per fare un passo di controllo degli oggetti sottostanti.

Si noti che i metodi della porta di ingresso e della porta di uscita sono gli stessi. I primi due svolgono identiche funzioni; le differenze sono nel terzo. Conviene definirli nella classe `Porta` e derivarli (eventualmente modificandoli).

4. Algoritmo di controllo della porta di ingresso

il metodo `tick()` fornisce il modo far percorrere alla porta il suo diagramma di stato. Ad ogni chiamata viene percorso il ramo corrispondente allo stato, leggendo lo stato degli oggetti coordinati e comandando attraverso di essi gli attuatori.

```

void tick() {
  switch (STATO)
  {
  case Idle: if (Pulsante.st() == Premuto)
              {Emittitore.go(); STATO= EmissioneBiglietto}; break;
  case EmissioneBiglietto: if (Emittitore.st() == Ritirato)
                          {Sbarra.open(); STATO= AperturaSbarra}; break;
  case AperturaSbarra: if (SensorePassaggio.st() == Transitato)
                       {Sbarra.close(); STATO= ChiusuraSbarra}; break;
  case ChiusuraSbarra: if (Sbarra.st() == Chiusa) STATO= Idle; break;
  }
}
  
```

Si noti che gli oggetti `Pulsante`, `Emittitore`, `SensorePassaggio` e `Sbarra` hanno il metodo `st()` che riporta il relativo stato. L'emittitore ha il metodo `go()` per emettere il biglietto. La Sbarra ha i metodi `open()` e

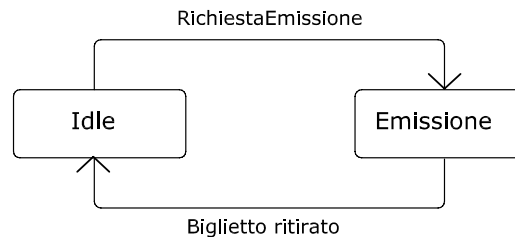


Figura 5: Diagramma di stato dell'oggetto Emittitore di biglietti.

`close()` per aprire e chiudere la sbarra. I dettagli di gestione dei dispositivi restano incapsulati negli oggetti che li rappresentano.

Ad esempio, l'oggetto Emittitore di biglietti avrà un diagramma di stato come in Figura 5

Corrispondentemente l'Emittitore ha questi due metodi:

```

void go() {
    if (ST == Idle) {ST= inEmissione;
                    comando al dispositivo all'emissione;}
}
  
```

```

Stato st() {
    if (ST == inEmissione & BigliettoRitirato) ST= Idle; return ST;
}
  
```

Il primo fa passare allo stato `inEmissione` e avvia l'emissione. Il secondo riporta allo stato `Idle` solo se il biglietto è stato estratto. La condizione `BigliettoRitirato` viene valutata guardando il dispositivo fisico.

5. Algoritmo di controllo della porta di uscita

Mutatis mutandis è molto simile a quello della porta di ingresso.

6. Conteggio ingressi

Il conteggio degli ingressi può essere effettuato dalle porte di entrata. Il diagramma di stato ci garantisce che ogni volta che si esce dallo stato di idle viene effettuato un percorso che riporta allo stato stesso. Il conteggio si riduce a tener conto di quante volte si esce dallo stato di idle (o da qualunque altro stato). Il ramo relativo dello statement `switch` si modifica in questo modo:

```

case Idle: if (Pulsante.st() == Premuto)
            {Emittitore.go(); STATO= EmissioneBiglietto;
             cont= cont+1;}; break;
  
```

dove `cont` è una variabile interna. Occorre aggiungere due metodi all'interfaccia:

- `getcont()` che restituisce il valore del contatore;
- `resetcont()` che azzerava il contatore.

L'oggetto Casello deve prevedere la lettura e l'azzeramento (eventuale) del contatore:

```

if (tempo == "mezzanotte")
    {nIngressi = Porta_di_Ingresso.getcont();
     Porta_di_Ingresso.resetcont();}
  
```

`tempo` è una variabile di sistema che tiene l'ora.

Alternativamente, il conteggio poteva essere effettuato da `SensorePassaggio`. Alla mezzanotte il Casello chiede ad ogni porta di ingresso di fornire il conteggio e questa gira la domanda al sensore.²

²Come ulteriore approfondimento si invita a dare la specifica delle interfacce relativamente a questa soluzione e scrivere l'algoritmo (seguito dal metodo principale) del sensore.

7. Calcolo incassi

In modo analogo si tratta il calcolo degli incassi. Ogni porta di uscita tiene una variabile nella quale conteggia l'incasso. La porta fornisce i metodi per leggere ed eventualmente azzerare l'incasso corrente. Il Casello chiama ai tempi dovuti tali metodi.