

## monitor-caronte

Caronte, si sa, traghetta le anime dannate verso l'inferno. Le anime sono gravate dal peso dei peccati e la barca di Caronte ha una portata massima MAXCARICO (per non affondare). Caronte attende sulla riva le anime e le imbarca fino a che ammettendo il peccatore successivo non si superi la portata. La barca può quindi partire e portare le anime verso l'inferno. Quando raggiunge l'inferno, i peccatori possono scendere. Quando tutti i peccatori sono scesi, la barca può tornare a riva.

```
shared Traghetto traghetto = new Traghetto
    ();
process Caronte {
    while (true) {
        traghetto.imbarca()
        ... traversata verso l'inferno
        traghetto.sbarca()
        ... traversata da solo di ritorno
    }
}
process dannato[i] {
    defungi;
    L=calcola peso peccati(i);
    traghetto.sali(i,L)
    ... viene traghettato
    traghetto.scendi(i)
    ... vai nel girone opportuno
    while (true)
        contrappasso(i);
}
```

**Scrivere il monitor Traghetto**

Soluzione:

```
monitor Traghetto {

condition caricare;
condition scaricare;
condition salire;
condition scendere;
int posizione = RIVA; // RIVA, FIUME, INFERNO
int dannati = 0;
int pesopeccati = 0;
int Queue coda = new Queue; // metodi di Queue: add, remove, readFirst

void imbarca()
    posizione = RIVA;
    // Invariant: la barca e' vuota, perche' appena arrivata
    // Posso svegliarne uno, perche' per definizione almeno uno puo' salire
    salire.signal();
    if (pesopeccati + coda.readFirst() < MAXCARICO)
        caricare.wait();
    posizione = FIUME;
}

void sbarca()
{
    posizione = INFERNO;
    scendere.signal();
    if (anime > 0) {
        scaricare.wait();
    }
}

int sali(int i, int L)
{
    if (posizione == RIVA && pesopeccati+L > MAXCARICO)
        caricare.signal();
    if (posizione != RIVA || pesopeccati+L > MAXCARICO) {
        queue.add(L);
        salire.wait();
        queue.remove();
    }
    pesopeccati = pesopeccati+L;
    anime = anime+1;
    if (pesopeccati + coda.readFirst() < MAXCARICO)
        salire.signal();
}

int scendi(int i, int L)
{
    if (posizione != INFERNO)
        scendere.signal();
    anime--;
    if (anime > 0)
        scendere.signal();
    else
        scaricare.signal();
}

}
```