

# Progettazione del Software

Giuseppe De Giacomo

*Dipartimento di Informatica e Sistemistica  
SAPIENZA Università di Roma*

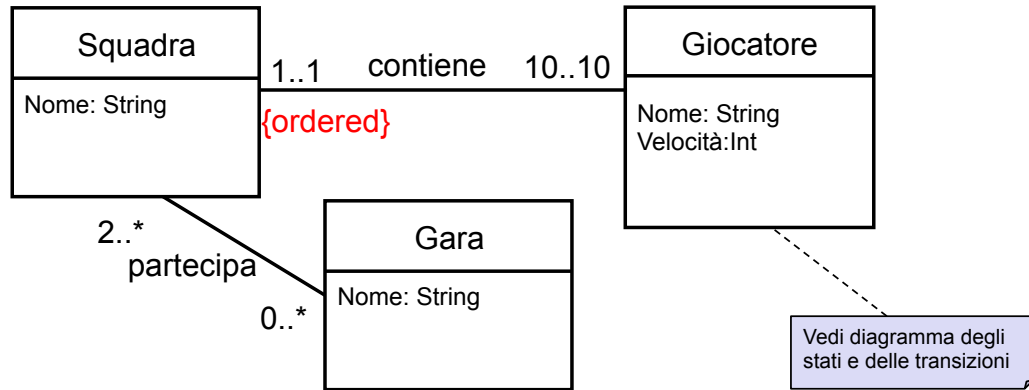
<http://www.dis.uniroma1.it/~degiacomo>

## Diagramma degli stati e delle transizioni: esempio Staffetta

## Requisiti

- Consideriamo la seguente applicazione relativa a gare di staffetta: Ogni squadra (con nome) è formata da 10 giocatori ciascuno con un proprio ordine di gara. Ogni giocatore (che appartiene a esattamente una squadra) è un 100 metrista con nome e parametro intero che indica la sua velocità nominale sui 100 metri. Le gare (con nome) consistono in percorrere 1km usando ogni giocatore per 100 metri. Alle gare partecipano almeno 2 squadre.
- Un giocatore è inizialmente “in allenamento”. Quando inizia una gara passa allo stato “in gara”. Ciascun giocatore che è in gara quando riceve il bastone corre percorrendo un tratto compreso tra 1 e 100 metri (il numero effettivo è random). Alla fine del tratto percorso se non ha superato i 100 corre per un altro tratto. Altrimenti, smette di giocare e passa il bastone al successivo giocatore della squadra (secondo l’ordine di squadra). Quando la gara è finita torna in allenamento. Il giocatore può essere modificato solo quando è in allenamento.

# Diagramma delle classi



## Responsabilità sulle associazioni

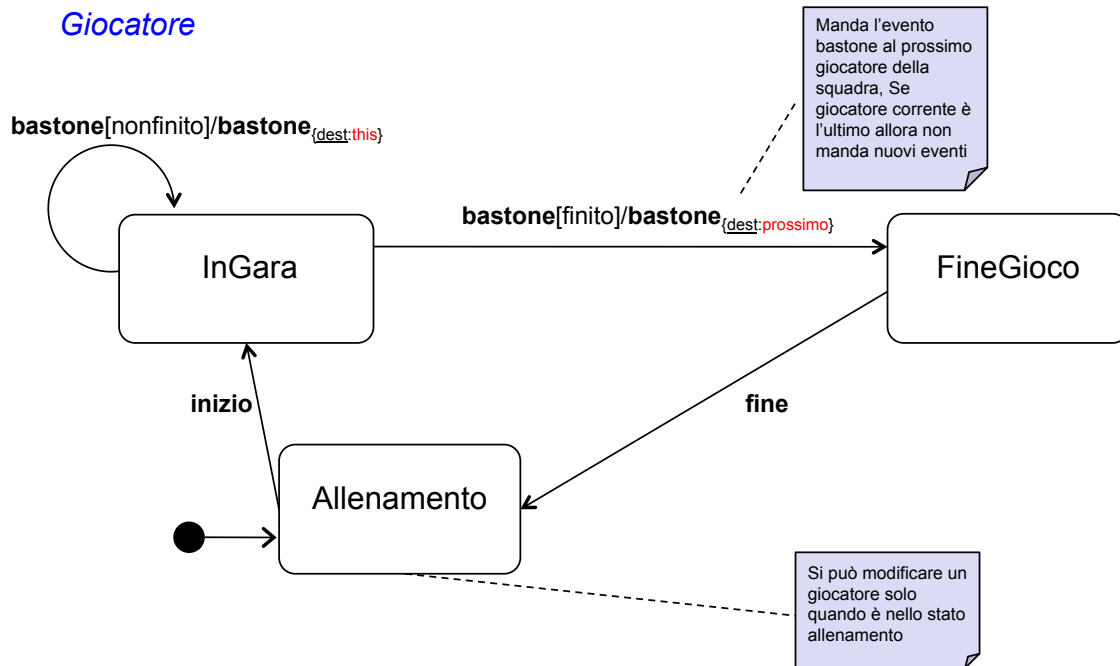
Classi responsabili della associazione **contiene**:

- **Squadra** (molteplicità, operazioni)
- **Giocatore** (molteplicità, operazioni)

Classi responsabili della associazione **partecipa**:

- **Gara** (molteplicità, operazioni)

# Diagrammi degli stati e delle transizioni



## Specifica degli stati di Giocatore

### Inizio Specifica Stati Classe Giocatore

Stato: {Allenamento, InGara, FineGioco}

Variabili di stato ausiliarie:

trattopercorso: reale

Stato iniziale:

*statoCorrente* = Allenamento

trattopercorso = --

### Fine Specifica

# Specifica delle transizioni di Giocatore

## InizioSpecificaTransizioniClasse Giocatore

Transizione: Allenamento → InGara  
inizio

Evento: inizio

Condizione: nessuna

Azione:

pre: nessuna //naturalmente deve essere: *evento.dest = this*  
post: *this.trattopercorso = 0*

...

## FineSpecifica

# Specifica delle transizioni di Giocatore

## InizioSpecificaTransizioniClasse Giocatore

...

Transizione: InGara → InGara  
bastone[nonarrivato]/bastone<sub>{dest:this}</sub>

Evento: bastone

Condizione: *this.trattopercorso < 100*

Azione:

pre: nessuna  
post: *nuovoevento = bastone{mitt = this, dest = this}* and  
*this.trattopercorso = Pre(this.trattopercorso)+rand[1,101)*

...

## FineSpecifica

## Specifica delle transizioni di Giocatore

InizioSpecificaTransizioniClasse Giocatore

...

Transizione: InGara → FuoriGioco  
bastone[arrivato]/bastone<sub>{dest:prossimo}</sub>

Evento: bastone

Condizione: *this*.trattopercorso >= 100

Azione:

pre: nessuna

post: sia

s tale che <*this*,s> in squadra //NB s è unico!

ip = indexOf(squadra(*this*,s))+1

se ip < 10 allora

sia prossimo tale che squadra(squadra(prossimo,s)) = ip

*nuovoevento* = bastone{mitt = *this*, dest = prossimo}

...

FineSpecifica

## Specifica delle transizioni di Giocatore

InizioSpecificaTransizioniClasse Giocatore

...

Transizione: FuoriGioco → Allenamento  
fine

Evento: fine

Condizione: nessuna

Azione:

pre: nessuna

post: --

FineSpecifica