

# Progettazione del Software

Giuseppe De Giacomo

*Dipartimento di Informatica e Sistemistica*

*SAPIENZA Università di Roma*

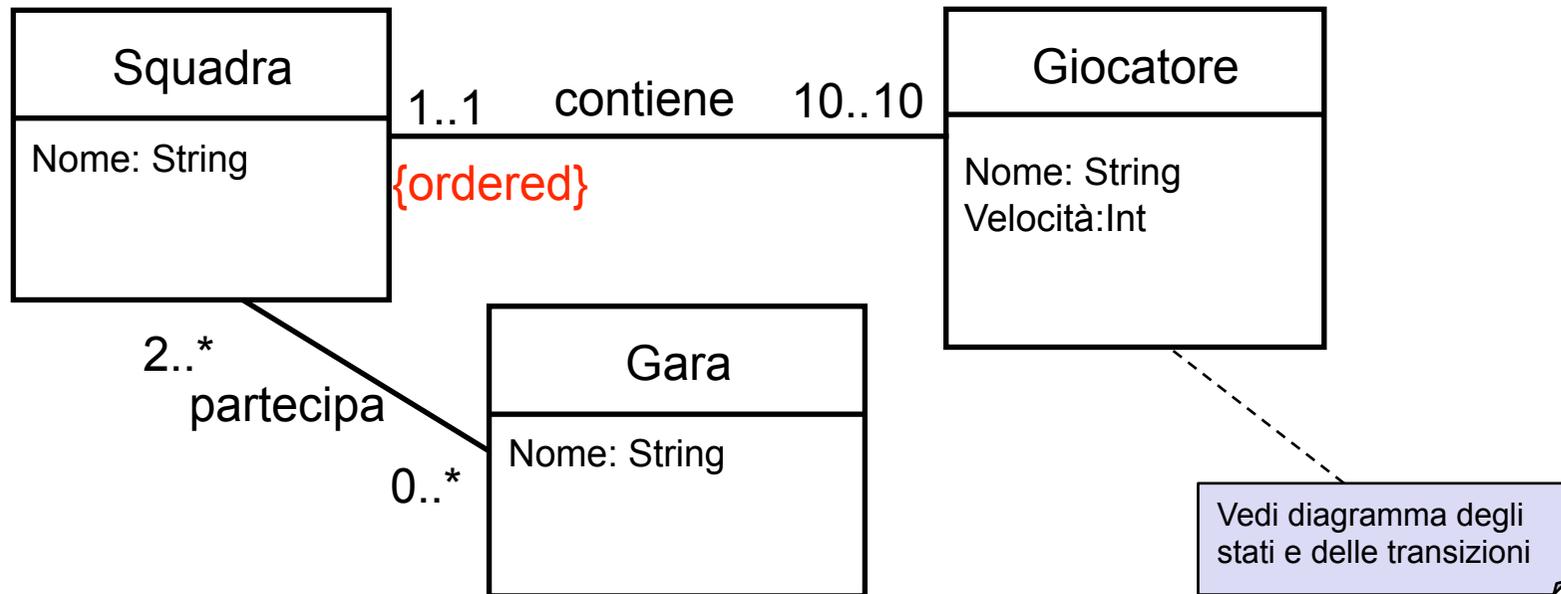
<http://www.dis.uniroma1.it/~degiacomo>

**Diagramma degli stati e delle transizioni**

# 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 0 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

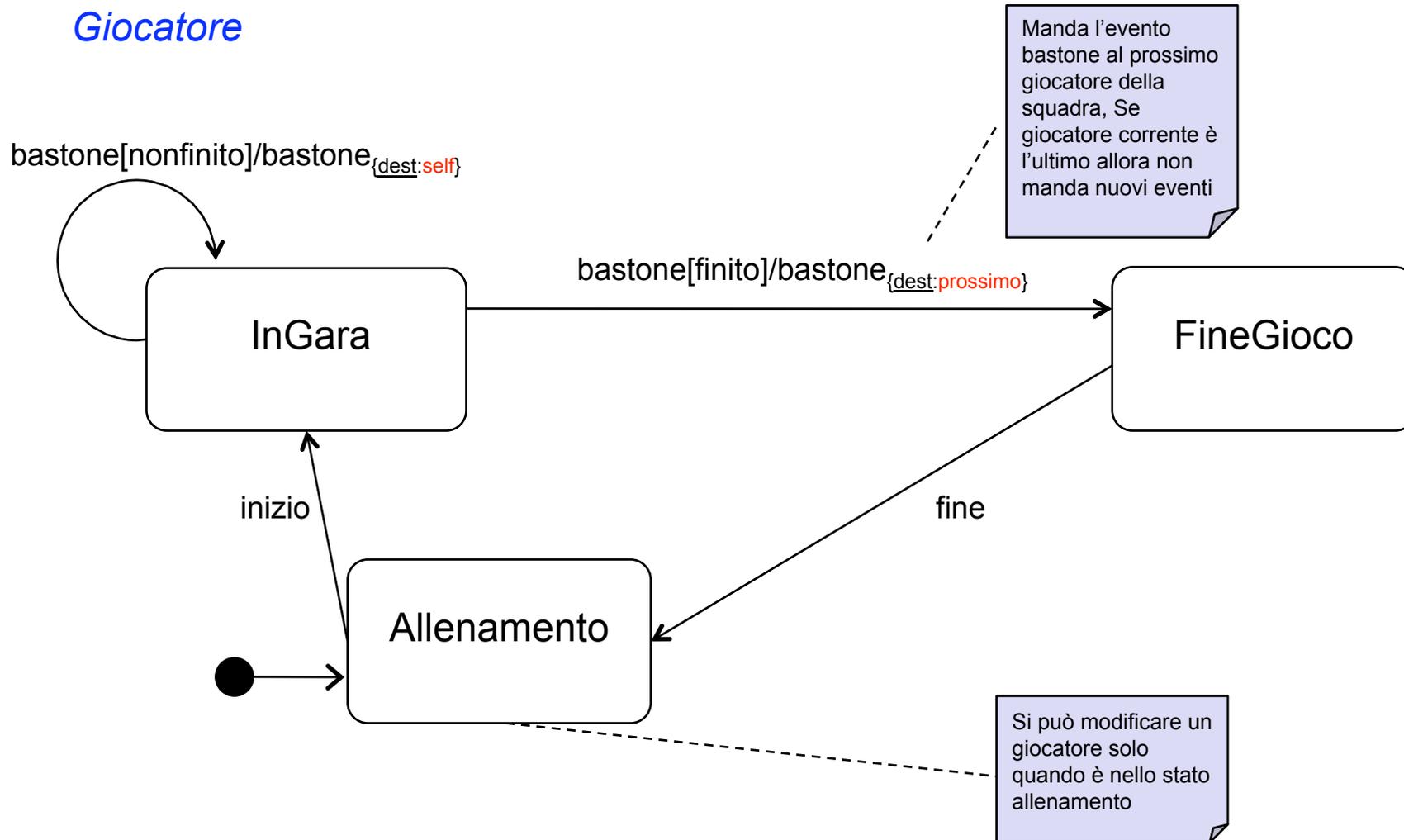


# Responsabilità sulle associazioni

- contiene:
  - Squadra (molteplicità, operazioni)
  - Giocatore (molteplicità, operazioni)
- partecipa:
  - Gara (molteplicità, operazioni)

# Diagrammi degli stati e delle transizioni

## *Giocatore*



# Specifica degli stati di Giocatore

InizioSpecificaStatiClasse Giocatore

Stato: {Allenamento, InGara, FineGioco}

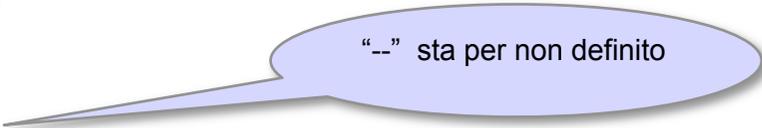
Variabili di stato ausiliarie:

trattopercorso: reale

Stato iniziale:

stato = Allenamento

trattopercorso= --



--" sta per non definito

FineSpecifica

# Specifica delle transizioni di Giocatore

InizioSpecificaTransizioniClasse Giocatore

Transizione: Allenamento → InGara  
inizio

Evento: inizio

Condizione: --

Azione:

pre: evento.dest = self

post: trattopercorso = 0

...

FineSpecifica

# Specifica delle transizioni di Giocatore

InizioSpecificaTransizioniClasse Giocatore

...

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

Evento: bastone

Condizione: trattopercorso < 100

Azione:

pre: evento.dest = self

post: nuovoevento =

bastone{mitt = self, dest = self} and  
self.trattopercorso = trattopercorso@pre+rand[1,101)

...

FineSpecifica

# Specifica delle transizioni di Giocatore

InizioSpecificaTransizioniClasse Giocatore

...

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

Evento: bastone

Condizione: trattopercorso >= 100

Azione:

pre: evento.dest = self

post: let indiceprossimo = self.squadra.indexOf(self)+1 in  
indiceprossimo < 10 implies

let prossimo = self.squadra[indiceprossimo] in

nuovoevento = bastone{mitt = self, dest = prossimo}

...

FineSpecifica

# Specifica delle transizioni di Giocatore

InizioSpecificaTransizioniClasse Giocatore

Transizione: FuoriGioco → Allenamento  
fine

Evento: fine

Condizione: --

Azione:

pre: evento.dest = self

post: --

...

FineSpecifica