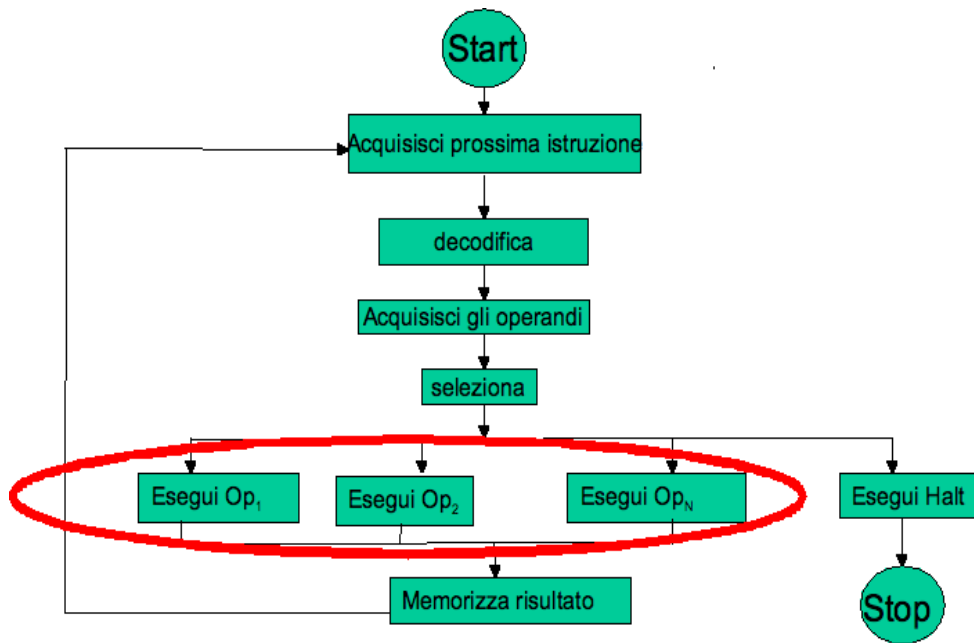


# Soluzioni prova intercorso

27 Aprile 2009

# Esercizio 1

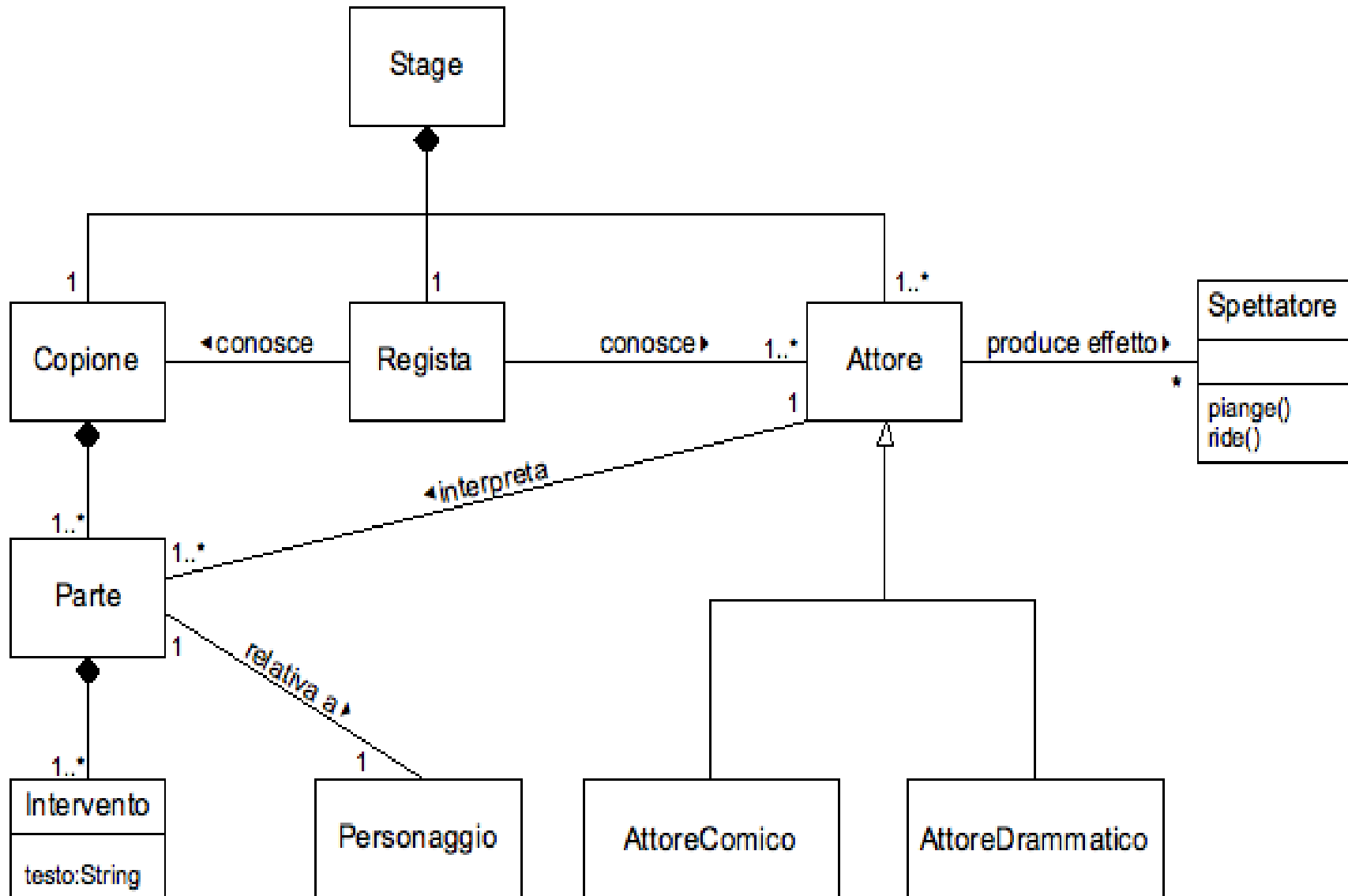


- (a) le operazioni non vengono eseguite in sequenza
- (a) le  $Op_i$  appartengono al linguaggio interpretato

# Esercizio 2

- Se il linguaggio è fortemente tipato
  - il legame di tipo è definito a compile time
  - il controllo di consistenza (in generale) è distribuito tra compile time e run time

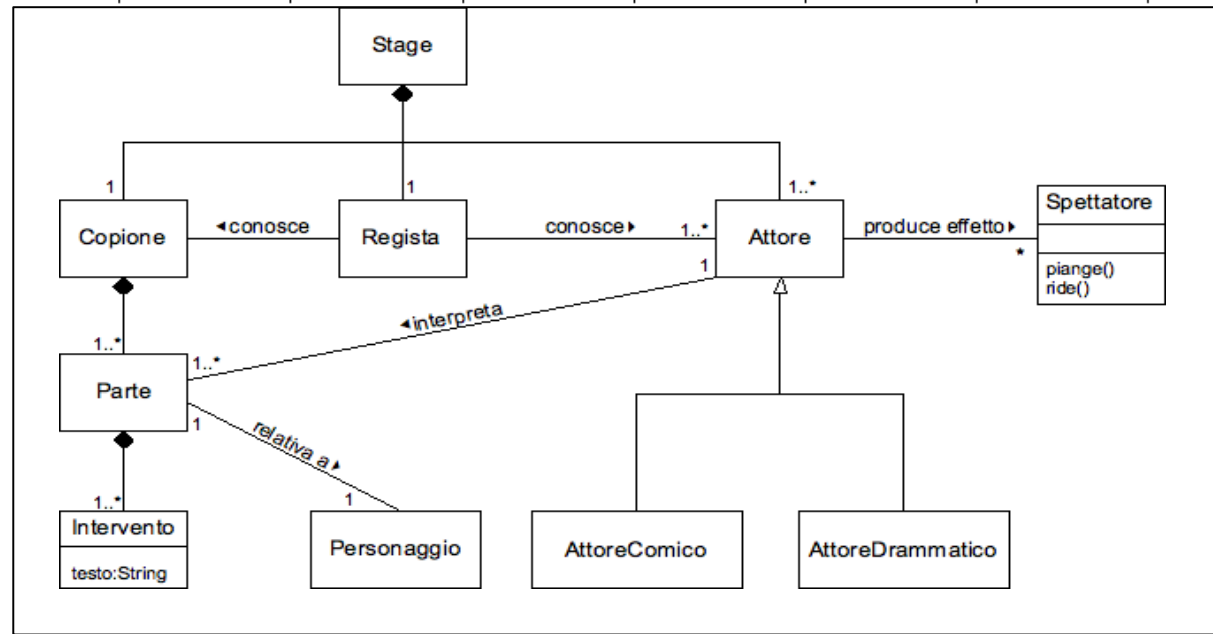
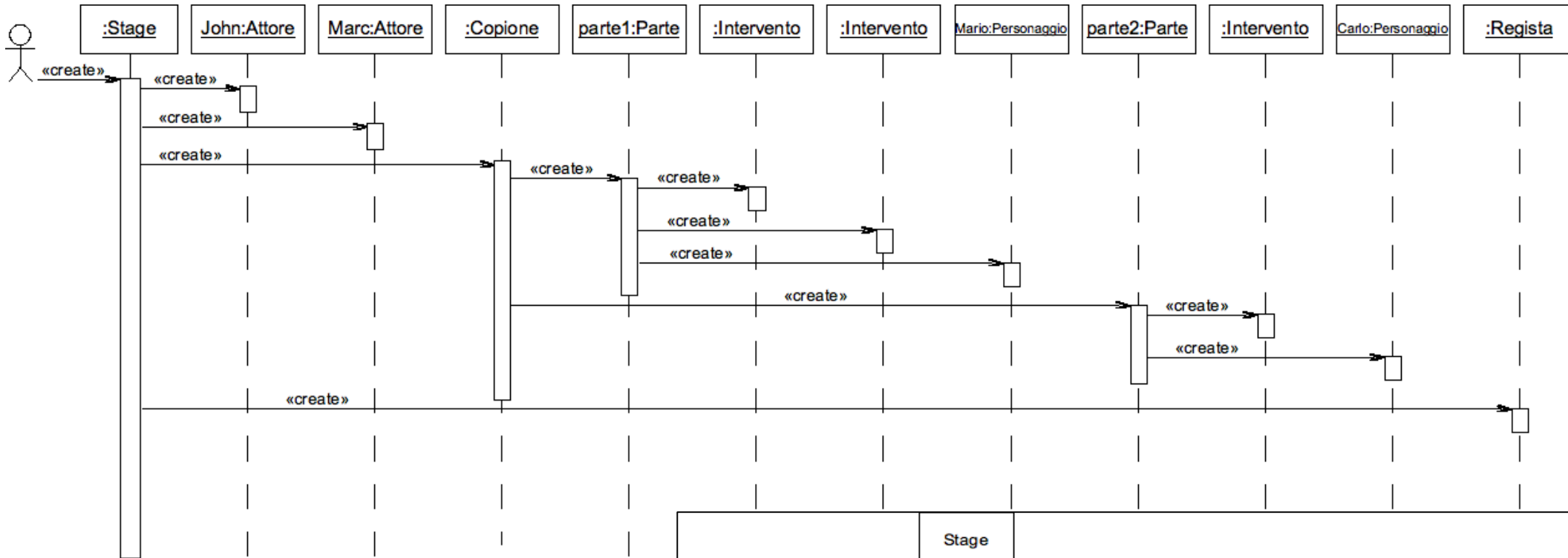
# Esercizio 3.1



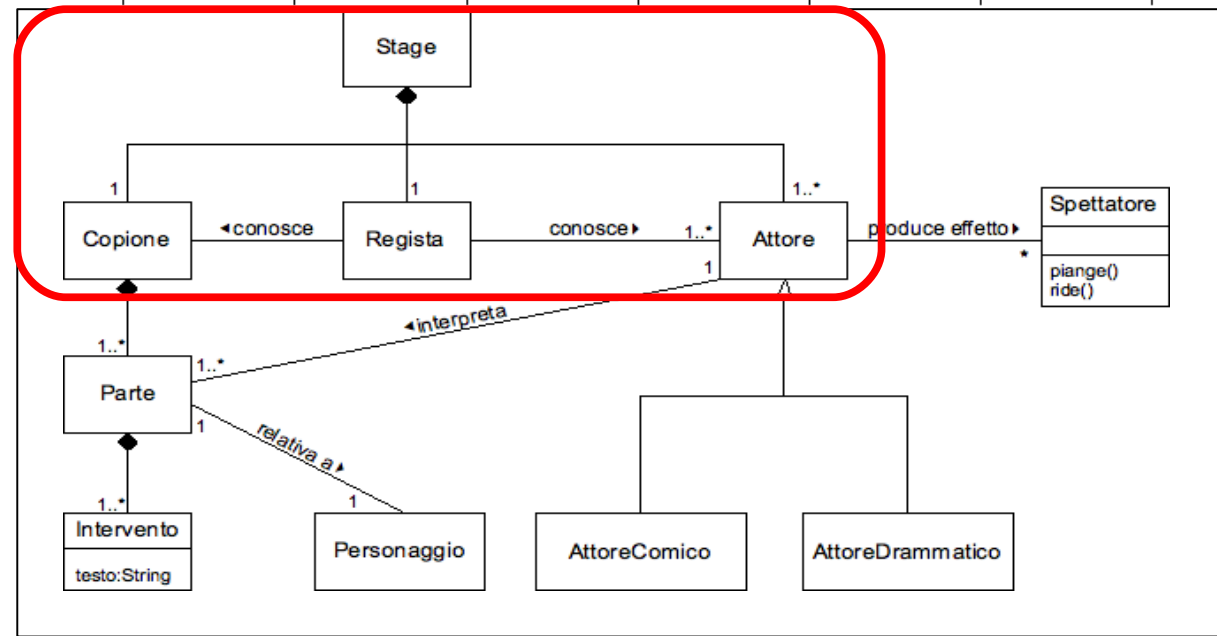
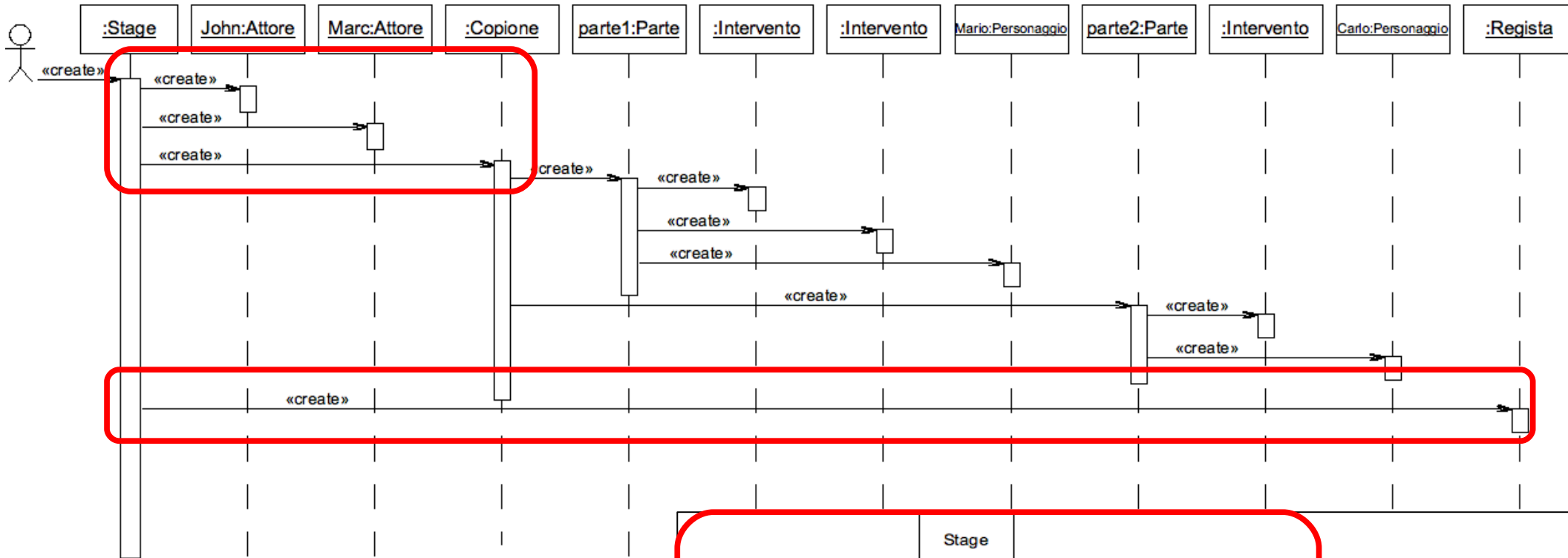
# Esercizio 3 (cont.)

- 3.2 Se introduciamo altri due attori Francesca e Marta:
  - aggiungiamo degli *oggetti*
  - potrebbero quindi cambiare i diagrammi che contengono oggetti:
    - degli oggetti, sequence, collaboration
- 3.3 Se introduciamo attori che possono suscitare applausi
  - aggiungiamo tipi e funzionalità nuovi
  - cambia certamente il diagramma delle classi

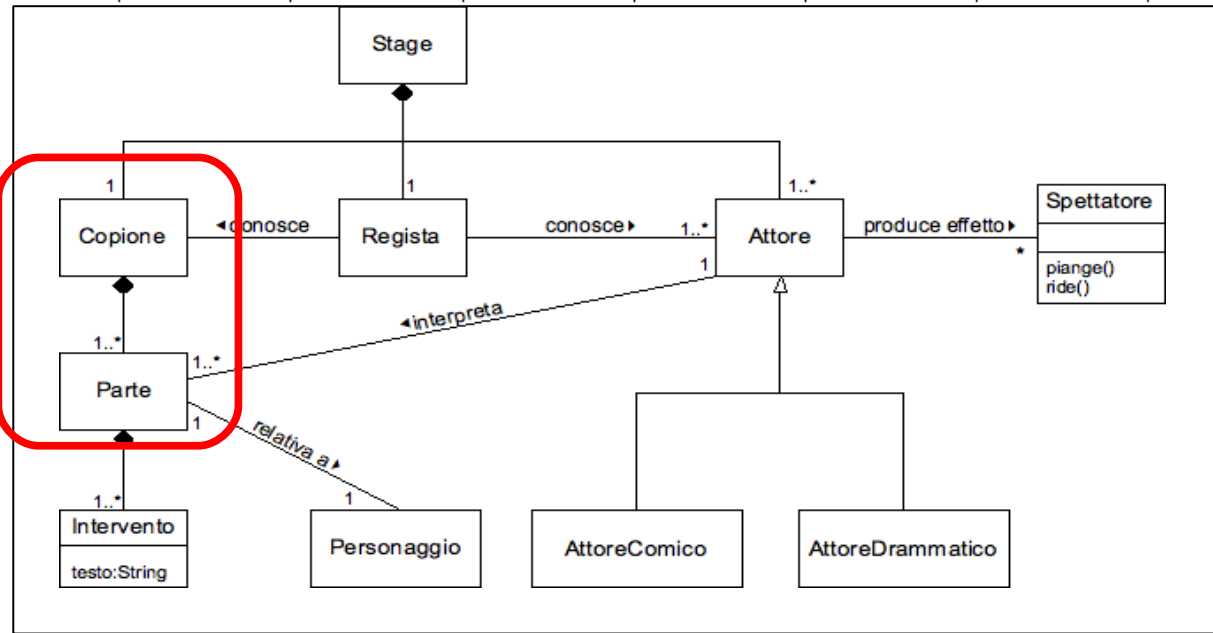
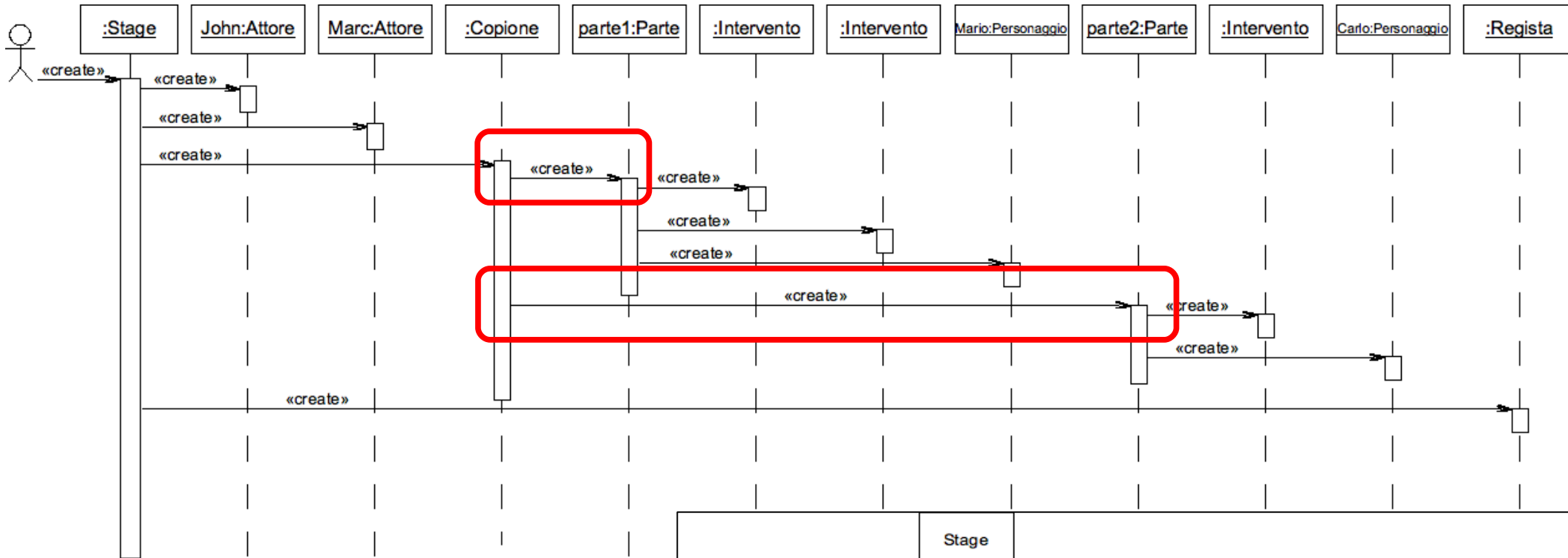
# Esercizio 4.1



# Esercizio 4.1

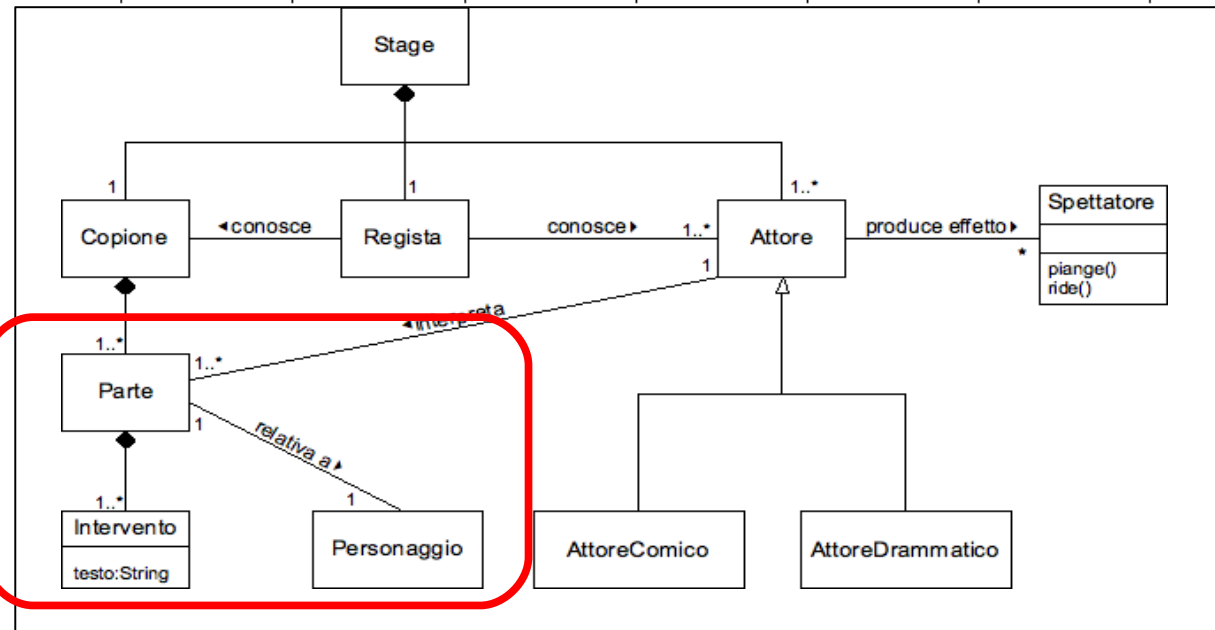
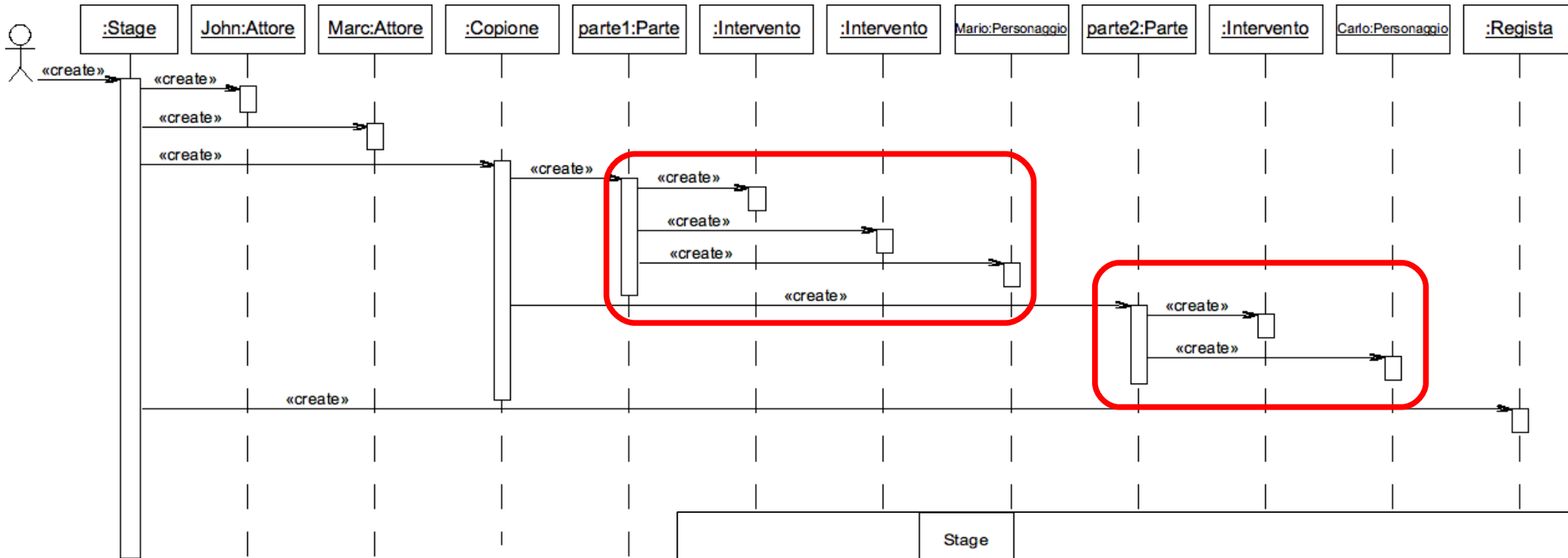


# Esercizio 4.1

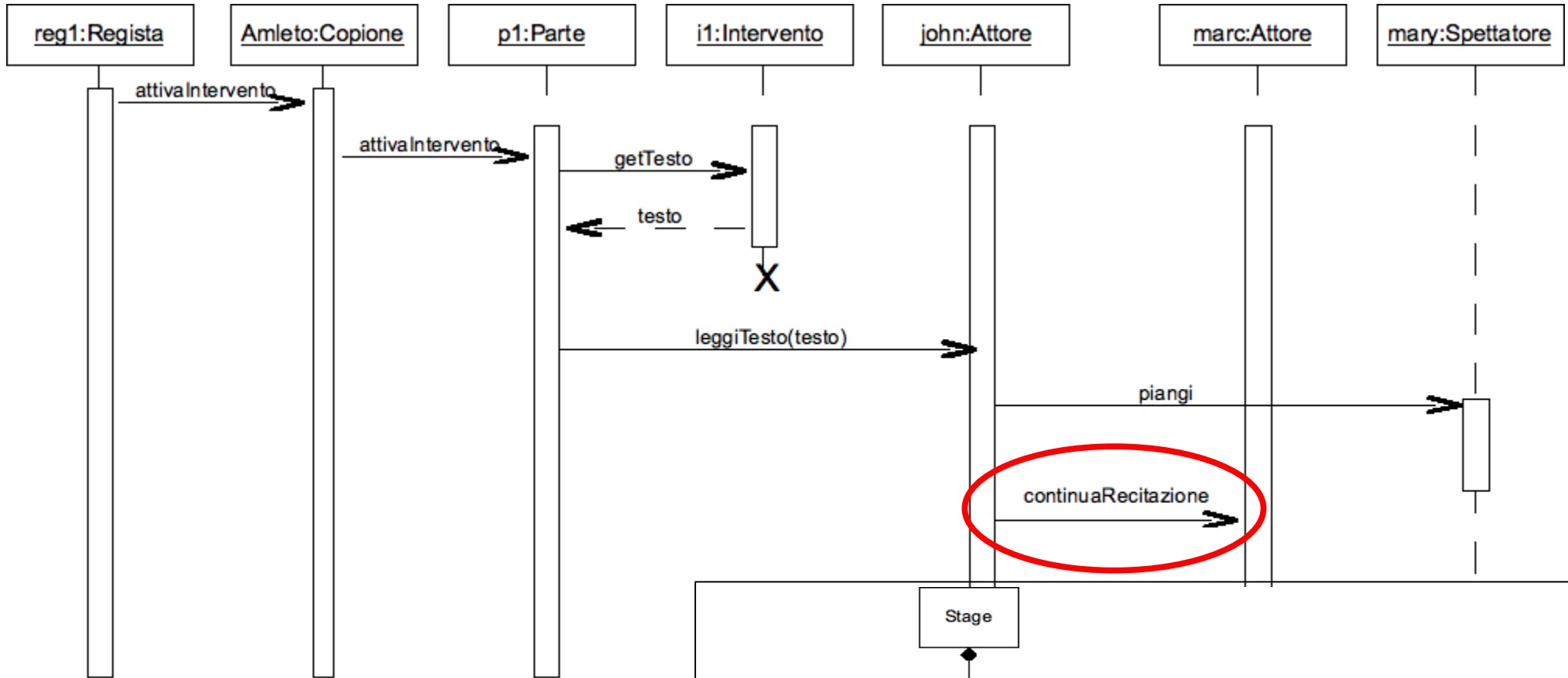




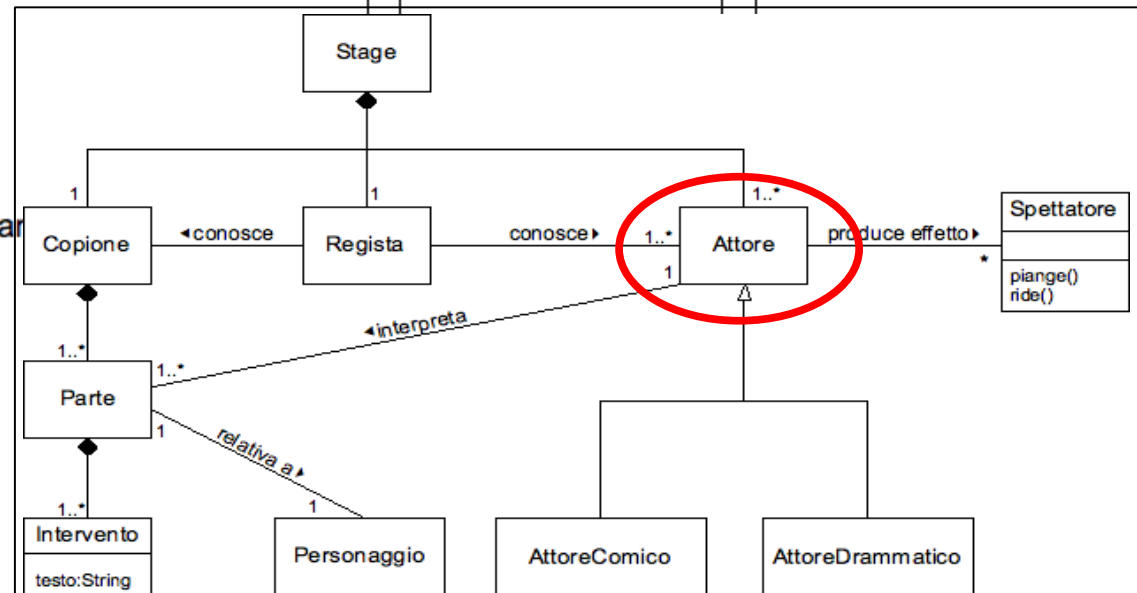
# Esercizio 4.1



# Esercizio 4.1



Sequence diagram



# Esercizio 5

- se in proc2 x è IN per indirizzo
  - ERRORE:  $x := x * y + 1;$
- se in proc2 x è OUT per indirizzo
  - ERRORE:  $x := x * y + 1;$
- se in proc2 y è OUT per indirizzo
  - ERRORE:  $x := x * y + 1;$

# Esercizio 5 (cont.)

- se in `proc2` `x` è IN OUT per indirizzo e `y` *non* è OUT: **1 10 20**
  - (nessuna modifica ai valori iniziali)
  - l'unica variabile non locale in `proc1` e `proc2` è `z`, che non compare mai a sinistra di un assegnamento nè come parametro OUT o IN OUT, quindi non viene modificata
  - **in `proc1`** `x == z+y` ( $\neq z$ ) quindi si esegue sempre il ramo `else` (`proc2(x,y)`) che può modificare `x` (locale a `proc1`, quindi nessun effetto all'esterno) ma non `y` (`proc2` non modifica il suo secondo parametro)