

# 10. Diagramma degli stati

Paola Barra  
a.a. 2023/2024

# Diagrammi UML

- **Diagrammi dei casi d'uso**
    - Descrivono il comportamento funzionale del sistema come sono visti dagli utenti
  - **Diagrammi delle classi**
    - Descrivono la struttura statica del sistema: oggetti, attributi, associazioni
  - **Diagrammi delle sequenze**
    - Descrivono il comportamento dinamico tra gli oggetti del sistema
  - **Diagrammi degli stati**
    - Descrivono il comportamento dinamico di un singolo oggetto
  - **Diagrammi delle attività**
    - Descrivono il comportamento dinamico di un sistema, in particolare il flusso di lavoro
-

# Diagramma degli stati

---

# Diagrammi degli stati

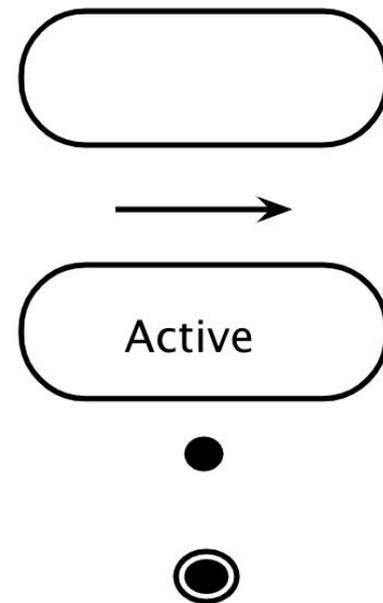
- Descrivono la sequenza di stati di un oggetto in seguito all'occorrenza di eventi esterni
- Sono estensioni del modello della macchina a stati finiti
  - Forniscono le notazioni per innestare stati e macchine degli stati
  - Forniscono notazioni per legare le transizioni agli invii di messaggi e le condizioni sugli oggetti
- Uno stato è una condizione soddisfatta dagli attributi di un oggetto
  - Ad esempio, un oggetto *Incident* in FRIEND può trovarsi in uno di quattro stati: **Active**, **Inactive**, **Closed**, **Archived**
    - Stato **Active**: situazione che richiede una risposta
    - Stato **Inactive**: situazione gestita ma i cui rapporti non sono stati ancora scritti
    - Stato **Closed**: situazione gestita e documentata
    - Stato **Archived**: incidente in stato Closed la cui documentazione è stata archiviata

# La macchina a stati

- Una macchina a stati descrive il comportamento dinamico delle istanze di un classificatore (per esempio degli oggetti istanza di una classe).
- Per costruire una macchina a stati dobbiamo individuare gli stati significativi in cui si può trovare un oggetto durante la sua vita.
- Inoltre dobbiamo descrivere come da ciascuno di questi stati l'oggetto può passare (transire) in un altro.
- Le transizioni avvengono in risposta al verificarsi di un evento. Gli eventi sono tipicamente;
  - messaggi inviati da altri oggetti
  - eventi generati internamente
- Una macchina a stati è rappresentata con un grafo di stati e transizioni, associata a un classificatore

# Transizioni e notazioni

- Una transizione rappresenta un cambiamento di stato attivato da eventi, condizioni o dal tempo
- Uno stato è rappresentato da un rettangolo arrotondato
- Una transizione è rappresentata da frecce che connettono due stati
- Gli stati sono etichettati col proprio nome
- Un circoletto nero rappresenta lo stato iniziale
- Un circoletto che circonda un circoletto nero rappresenta uno stato finale



## Esempio degli stati della vita di una lampadina

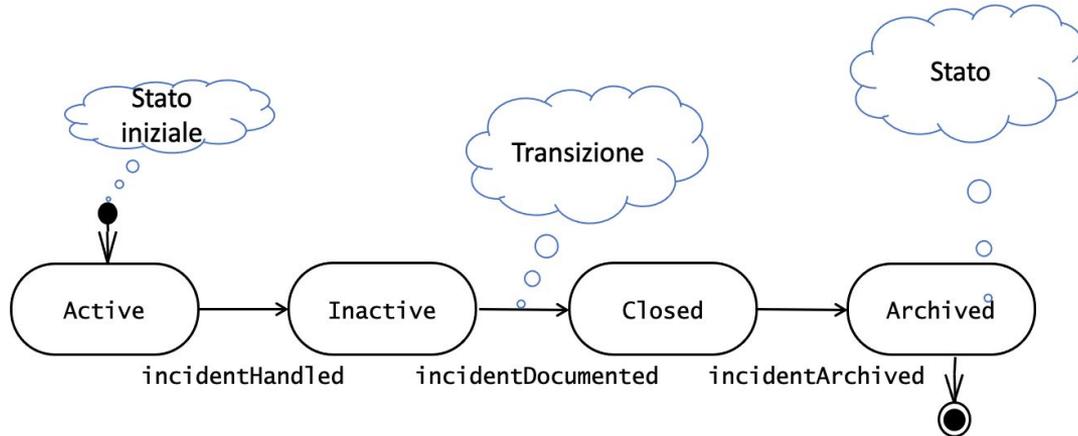
Descriviamo la vita di una lampadina



# Diagramma degli stati per la classe Incident

I quattro stati possono essere rappresentati con un singolo attributo, **status**, nella classe Incident che può assumere uno dei quattro valori {*Active*, *Inactive*, *Closed*, *Archived*}

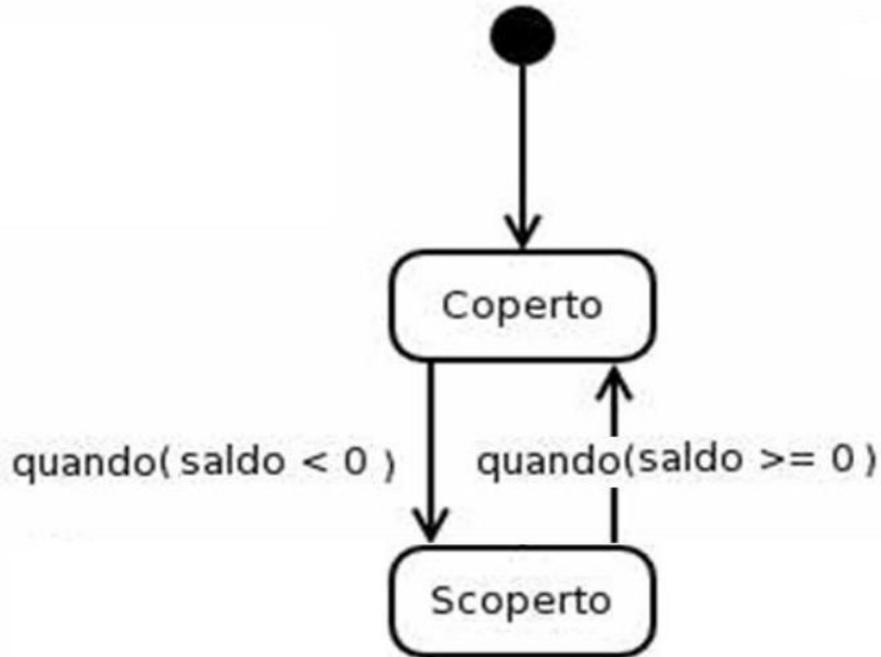
In generale uno stato può essere determinato dai valori più attributi



# Evento

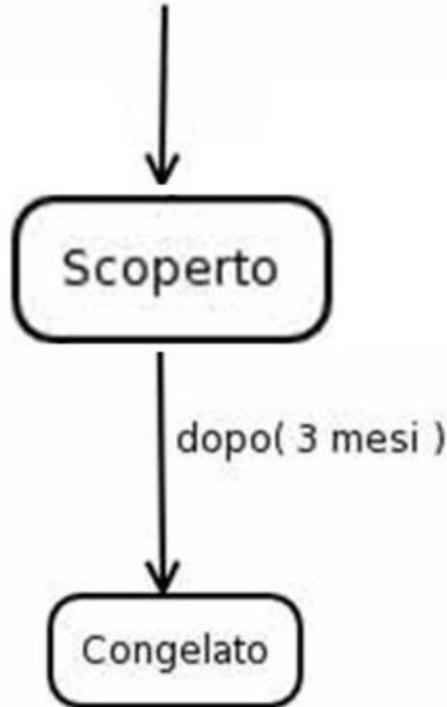
- Un evento è l'occorrenza di un fenomeno collocato nel tempo e nello spazio
- Un evento occorre istantaneamente
- Modellate qualcosa come un evento se ha delle conseguenze
- Gli eventi che arrivano in uno stato per cui non è prevista alcuna transizione etichettata con quell'evento vengono ignorati
- È ammesso il non-determinismo: un evento può fare da trigger a più transizioni:
  - Se le due transizioni escono dallo stesso stato, ne viene scelta una non-deterministicamente

# Eventi variazioni dello stato



- Un evento  
occorre in modo  
istantaneo
- una condizione  
non è istantanea
- è istantaneo il  
momento in cui  
diventa vera

# Eventi temporali

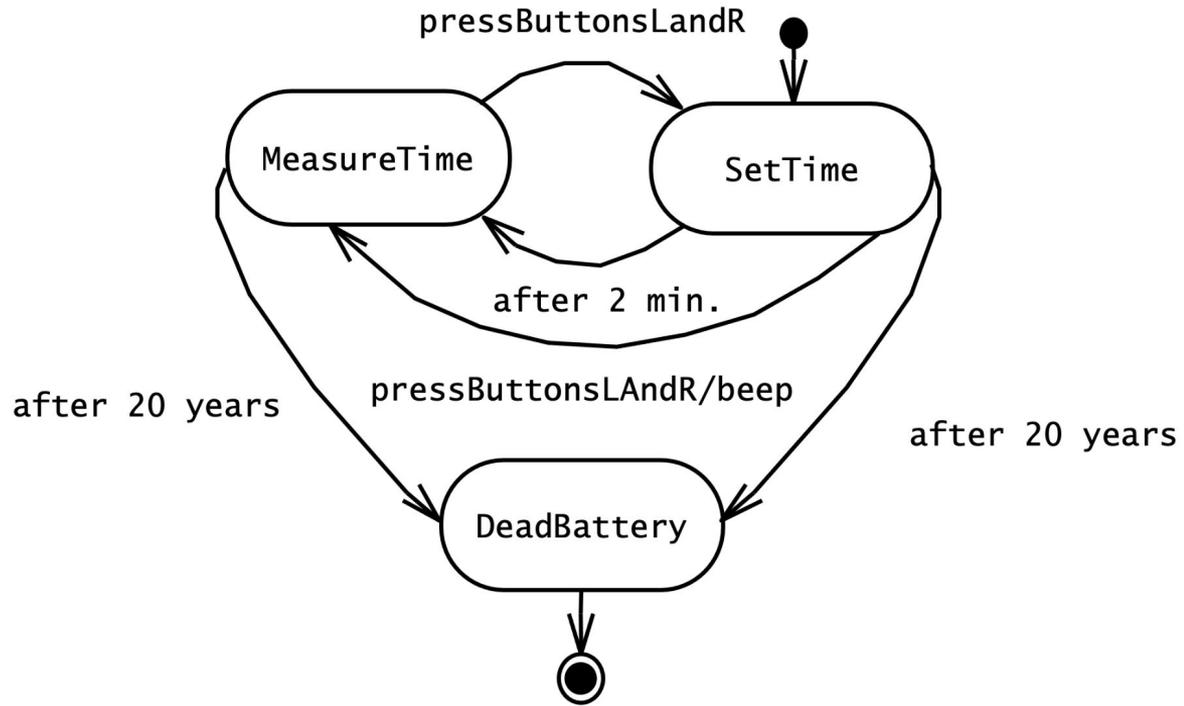


Dopo che l'oggetto è stato 3 mesi nello stato Scoperto, transisce nello stato Congelato

# Esempio 2BWatch

- Al più alto livello di astrazione, 2Bwatch ha due stati, *MeasureTime* e *SetTime*
- 2Bwatch cambia stato quando l'utente preme e rilascia entrambi i pulsanti simultaneamente
- Durante la transizione dallo stato *SetTime* allo stato *MeasureTime*, 2Bwatch emette un beep
  - Indicato dall'azione **/beep** sulla transizione
- Quando 2Bwatch è acceso per la prima volta si trova nello stato *SetTime*
  - Indicato dal cerchietto nero che indica lo stato iniziale
- Quando la batteria si esaurisce, l'orologio non funziona
  - Cerchietto nero circondato da un altro cerchietto, lo stato finale

# Esempio 2BWatch SetTime



# Azioni

- Le azioni sono piccoli comportamenti atomici eseguiti in punti specifici nella macchina a stati
- Le azioni richiedono per l'esecuzione una breve quantità di tempo e **non possono essere interrotte**
- Le azioni possono verificarsi in tre luoghi
  - durante una transizione
  - quando si entra in uno stato
  - quando si esce da uno stato

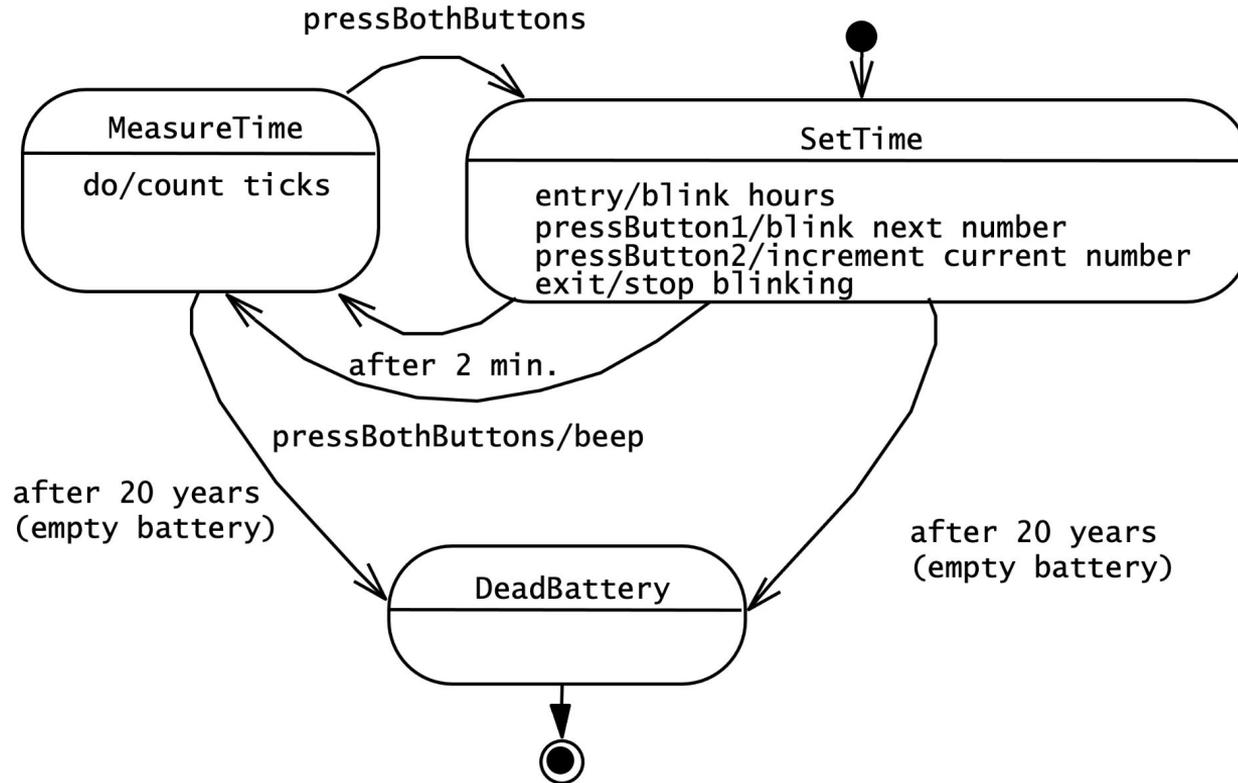
## Transizioni interne

- Durante una transizione, le azioni di uscita dello stato sorgente sono eseguite per prime, poi sono eseguite le azioni associate con le transizioni e poi sono eseguite le azioni di entrata dello stato di destinazione
- Una **transizione interna** è una transizione che non lascia lo stato
  - Sono causate da eventi e possono avere delle azioni associate
  - L'attivazione di una transizione interna non comporta alcuna azione di uscita o entrata

# Attività

- Un'attività è un comportamento che è eseguito fintantoché un oggetto si trova in un dato stato
  - Un'azione è breve e non interrompibile, un'attività può richiedere un certo quantitativo di tempo ed è interrotta quando ha inizio una transizione che esce da uno stato
- Le attività sono rappresentate con l'etichetta *do* posizionata all'interno dello stato in cui è eseguita

# Transizioni interne associate con lo stato SetTime



## Tipi di eventi

**Signal event** - corresponds to the arrival of an asynchronous message or signal

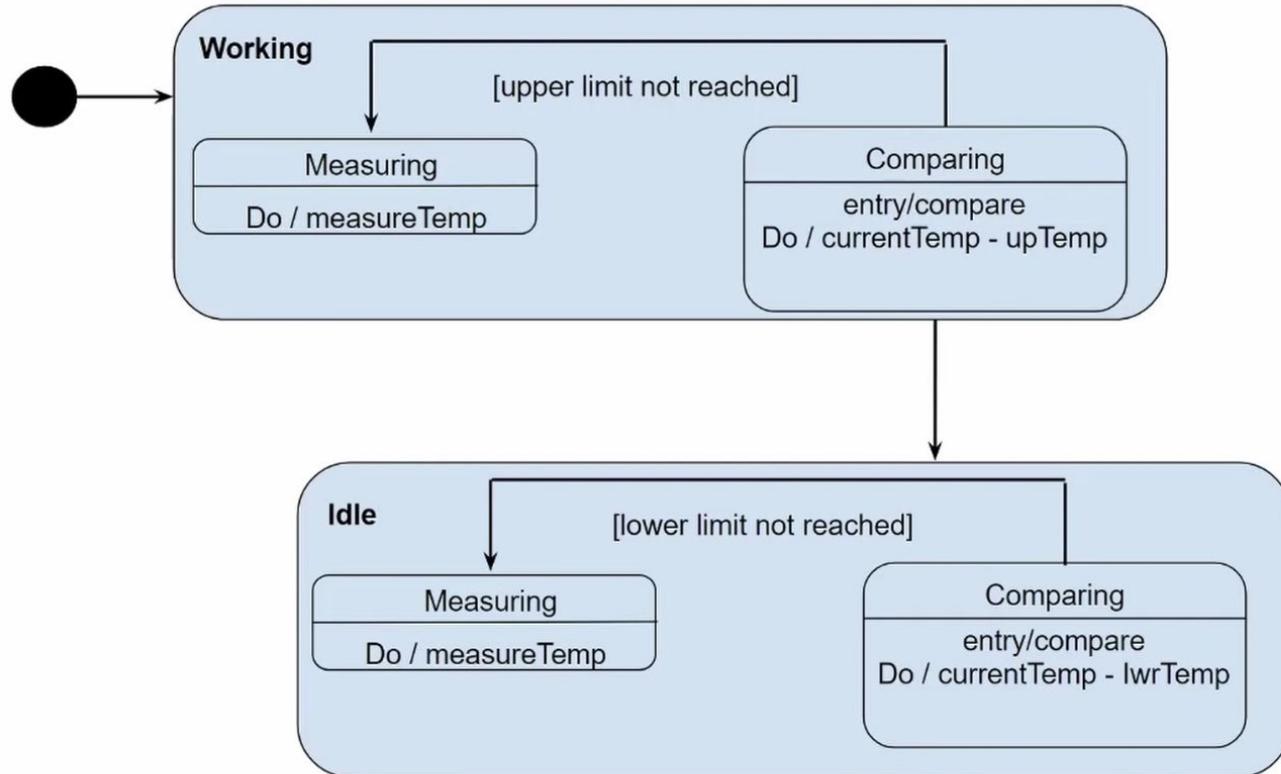
**Call event** - corresponds to the arrival of a procedural call to an operation

**Time event** - occurs after a specified time has elapsed

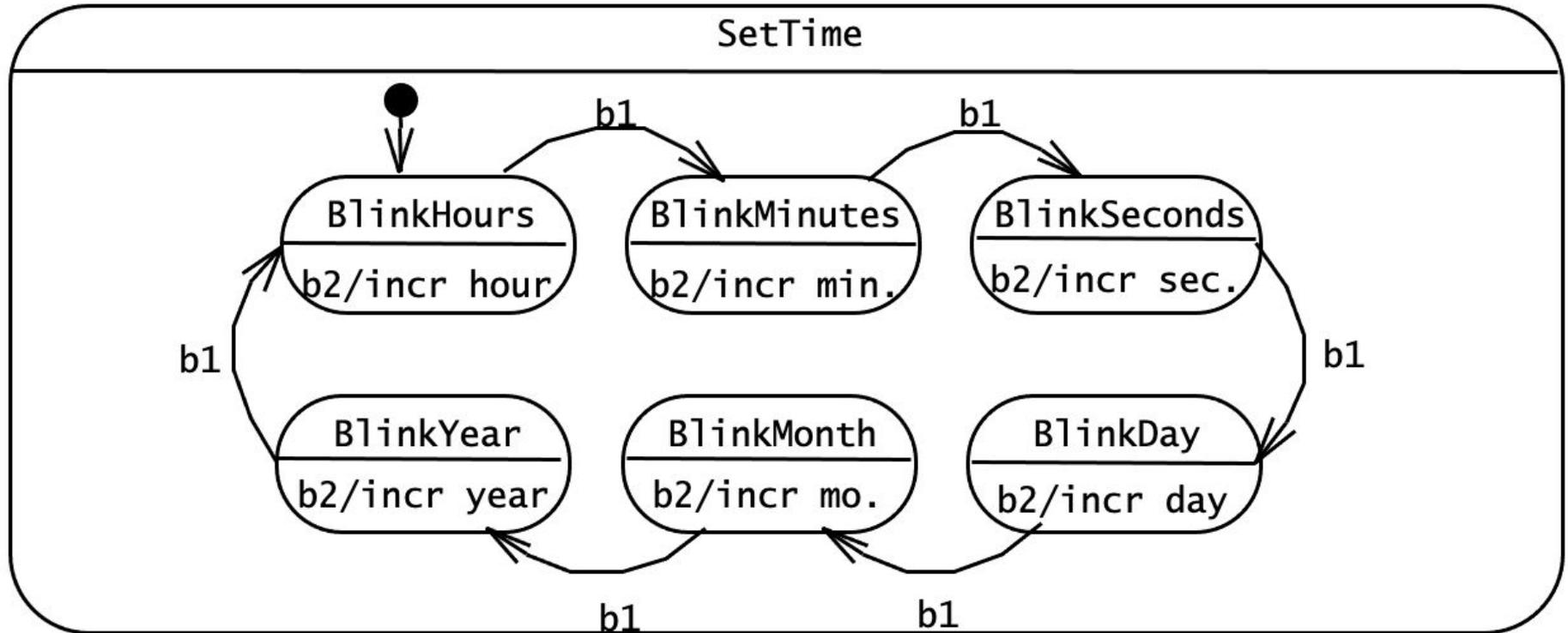
**Change event** - occurs whenever the specified condition is met

# Stati annidati

Gli stati possono essere annidati a qualsiasi livello.



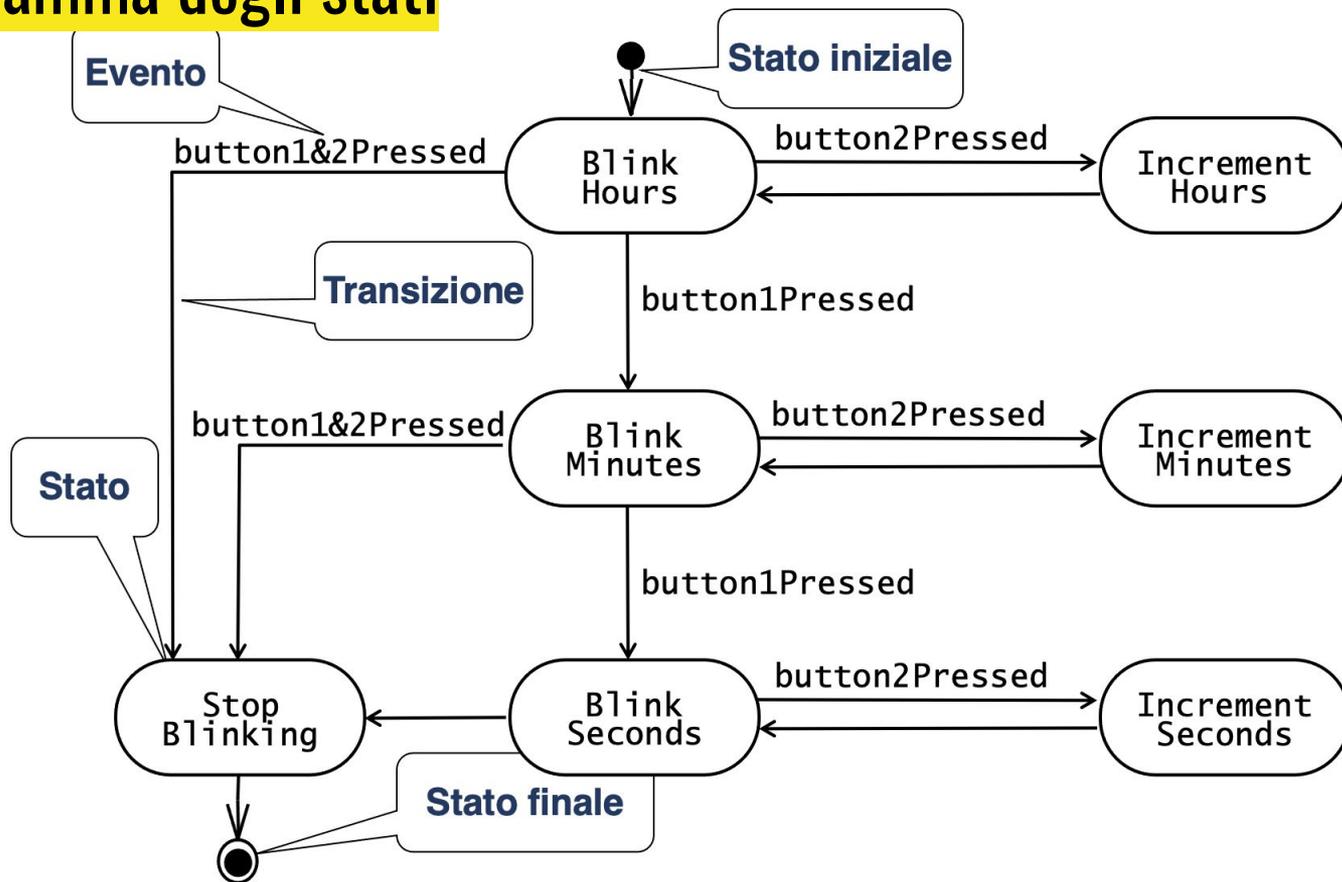
# Diagrammi di stato annidati



# Applicazioni diagrammi di stato

- Usati per rappresentare comportamenti non banali di un sottosistema o un oggetto
- A differenza dei diagrammi di interazione, che si focalizzano sugli eventi che influenzano il comportamento di un insieme di oggetti, i diagrammi di stato mettono a fuoco quale o quali attributi influenzano il comportamento di un singolo oggetto
- Usati per identificare attributi di oggetti e per rifinire tutte le descrizioni dei comportamenti di un oggetto
  - Invece, i diagrammi di interazione sono usati per identificare gli oggetti partecipanti ed i servizi che forniscono

# Diagramma degli stati



Rappresenta il comportamento di un singolo oggetto con un comportamento dinamico interessante

## Diagramma degli stati in breve

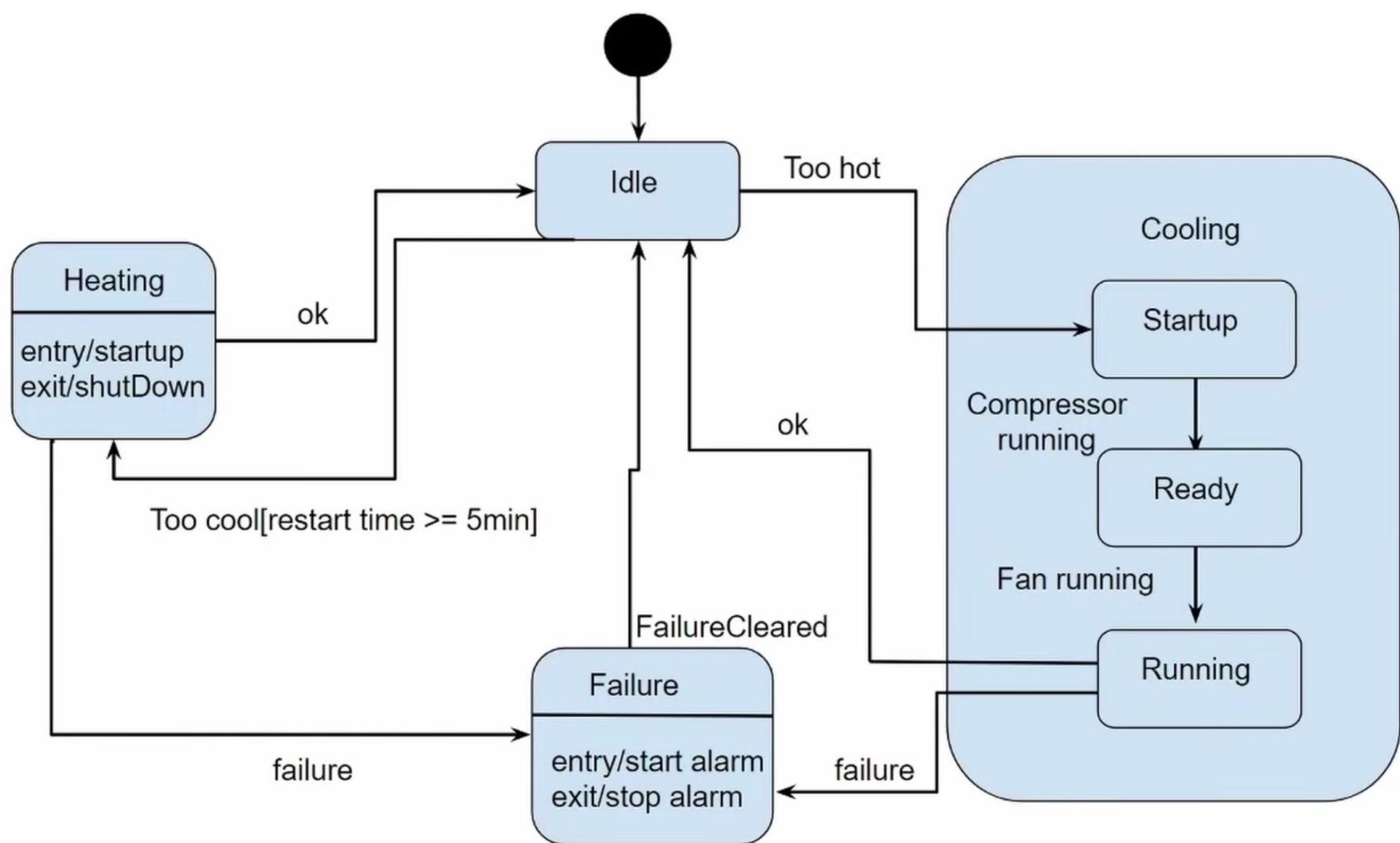
Il comportamento di un'entità è non solo direttamente il risultato di un input ma dipende anche dal comportamento degli stati precedenti.

Il diagramma degli stati mostra i vari stati delle entità.

Il diagramma degli stati può mostrare anche come reagisce un'entità ai vari eventi muovendosi da uno stato all'altro e può essere usato per modellare la natura dinamica del sistema.

Il diagramma degli stati è tipicamente usato per descrivere il comportamento di un oggetto che dipende dallo stato in cui si trova.

Si usa in congiunzione con il diagramma delle sequenze.

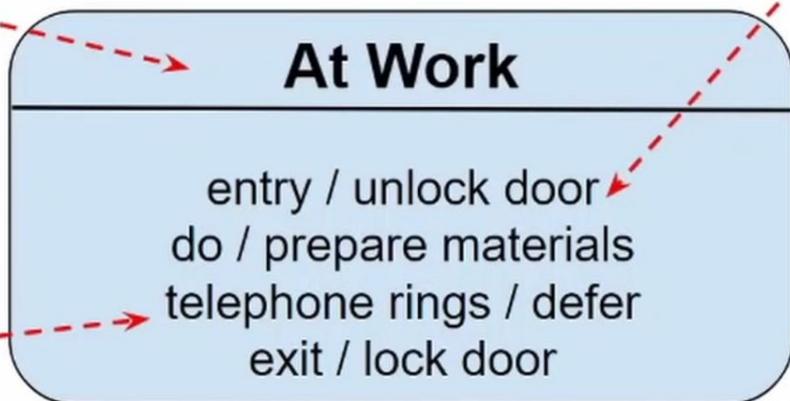


Name of the State

Entry Activity

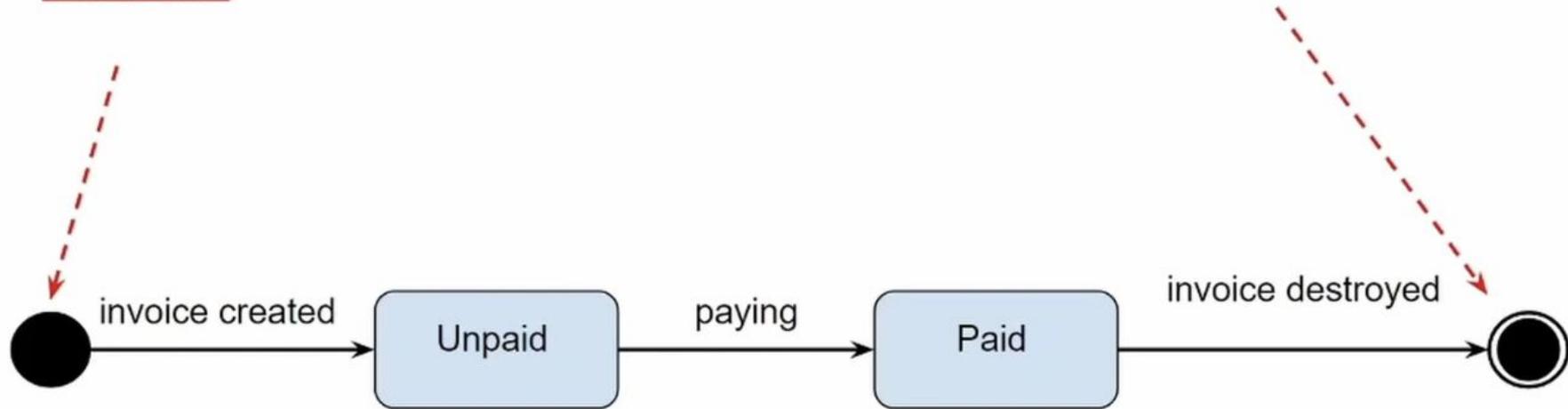
Deferrable Trigger

Event that does not trigger any state transition but remains in the event pool ready for processing when the object transitions to another state



Action performed on entry to state

Initial State



Final State

# Esercizio : modellare un Bancomat

- **Modellare un Bancomat**
- Disegnare un diagramma dei casi d'uso.
  - Il sistema comprende due attori: un **cliente**, che preleva denaro dal suo conto, in contanti o ricaricando su chip il suo bancomat, e un **addetto alla sicurezza**, che inserisce denaro nel bancomat.
  - I casi d'uso dovrebbero includere: *PrelevaContanti*, *CaricaDenaro*, *ControllaSaldo*, *RiempiBancomat*. Include anche i seguenti casi eccezionali: *DenaroEsaurito*, *TransizioneAnnullata* (cioé, il cliente ha selezionato il tasto Cancella senza completare la transazione) e *DenaroSuChipFuoriServizio*.
  - E' possibile usare l'ereditarietà tra casi d'uso
- Scrivere il flusso di eventi e specificare tutti i campi per il caso d'uso *ControllaSaldo*
  - Specificare le relazioni

# Diagramma degli stati

Il diagramma degli stati (e delle transizioni) **viene definito riferendosi a una singola classe.**

E intende descrivere le funzioni di un generico oggetto di quella classe.

Il diagramma rappresenta le sequenze di stati - le risposte e le azioni che un oggetto attraversa lungo la sua vita in risposta agli stimoli ottenuti.

Uno stato è una situazione in un cui un oggetto ha un oggetto di proprietà stabili.

Una transizione modella un cambiamento di stato.

## Esempio degli stati della vita di una lampadina

Descriviamo la vita di una lampadina



## Esempio di transizione



Quando premiamo un interruttore la lampadina cambia stato da ACCESO e SPENTO.

L'azione di “accendere” o “spegnere” è data dalla transizione.

La transizione è denotata dalla sequenze sintassi Evento[condizione] / Azione e ci porta da uno stato  $S_1$  a uno stato  $S_2$ .

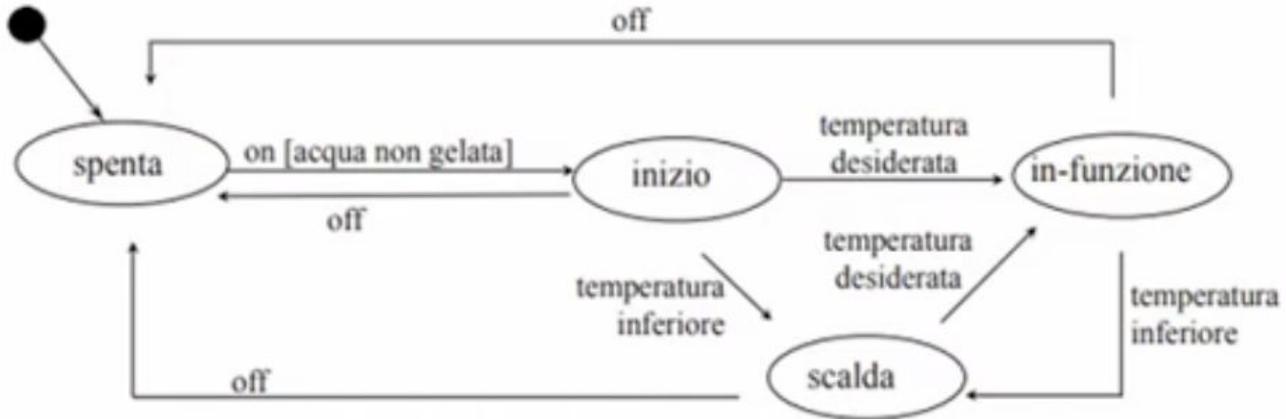
**Se l'oggetto si trova nello stato  $S_1$  e riceve l'evento E e la condizione C è verificata.**

**Allora attiva l'esecuzione dell'azione A e passa allo stato  $S_2$ .**

# Esempio funzionamento di una caldaia

In questo esempio ogni transizione è rappresentata solo da eventi e condizioni.

Partiamo dalla caldaia spenta, la possiamo accendere ecc...

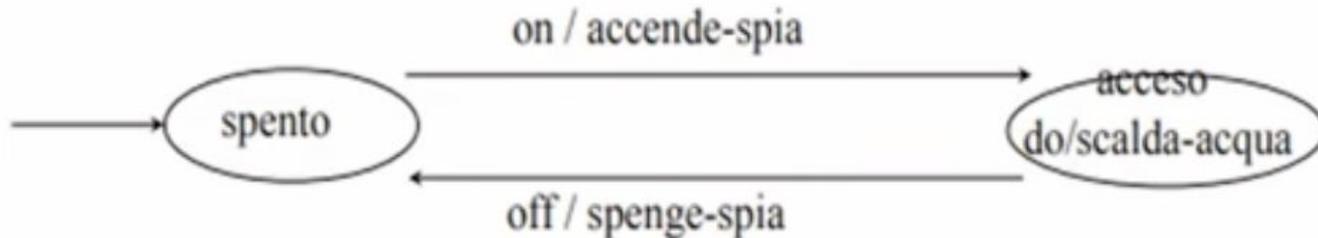


## Do/attività

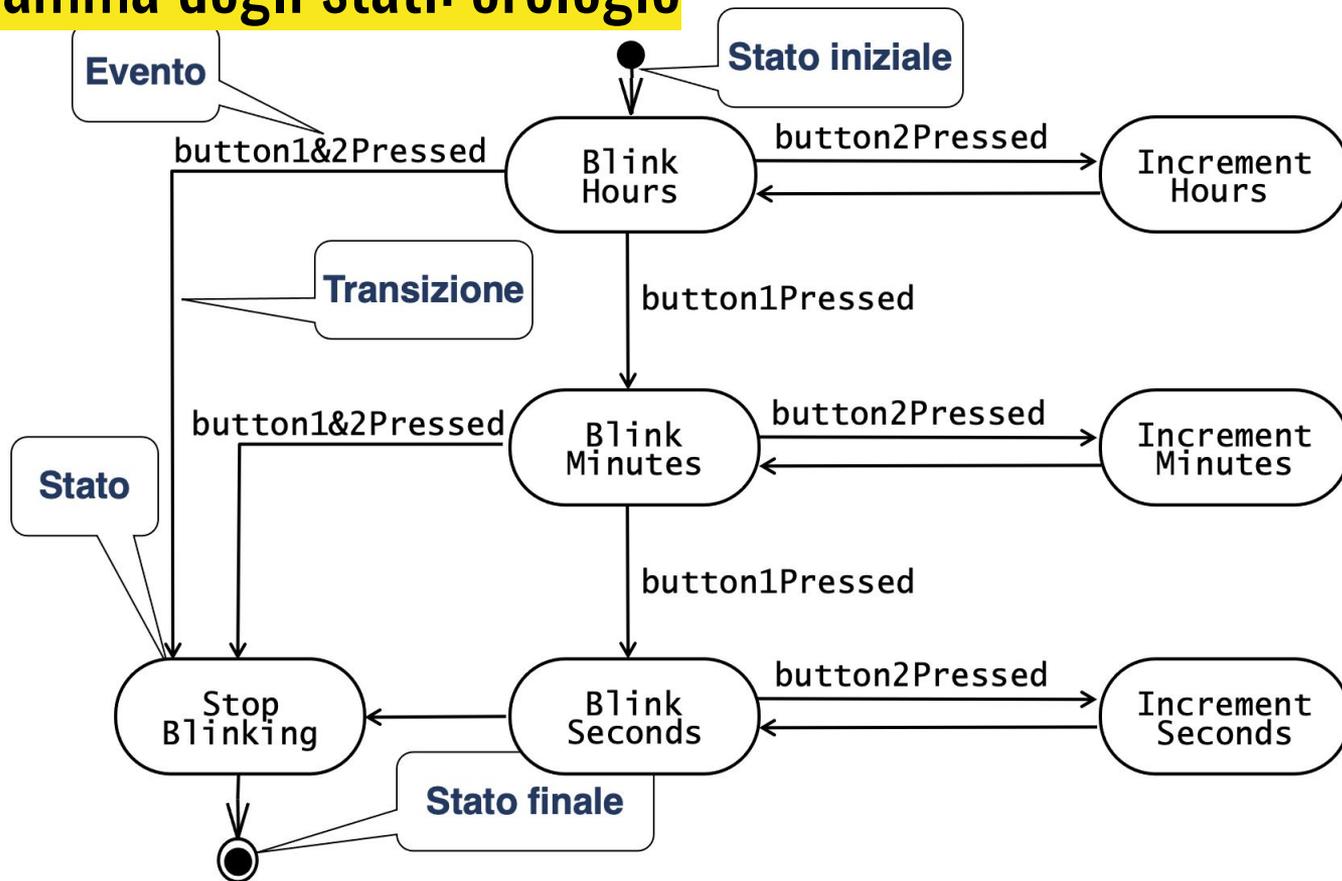
Alcune volte desideriamo affrontare dei processi che l'oggetto esegue senza cambiare stato.

Questi processi si chiamano attività e vengono mostrate tramite l'annotazione do/attività.

Esempio: lo scaldabagno, quando arriva l'evento "on" -> accendere spia quando sei nello stato "acceso" do-> scaldava acqua ma rimane sempre in questo stato.



# Diagramma degli stati: orologio



Rappresenta il comportamento di un singolo oggetto con un comportamento dinamico interessante

# Diagrammi UML

- **Diagrammi dei casi d'uso**
    - Descrivono il comportamento funzionale del sistema come sono visti dagli utenti
  - **Diagrammi delle classi**
    - Descrivono la struttura statica del sistema: oggetti, attributi, associazioni
  - **Diagrammi delle sequenze**
    - Descrivono il comportamento dinamico tra gli oggetti del sistema
  - **Diagrammi degli stati**
    - Descrivono il comportamento dinamico di un singolo oggetto
  - **Diagrammi delle attività**
    - Descrivono il comportamento dinamico di un sistema, in particolare il flusso di lavoro
-

# Diagramma delle sequenze

---

# Diagrammi delle sequenze

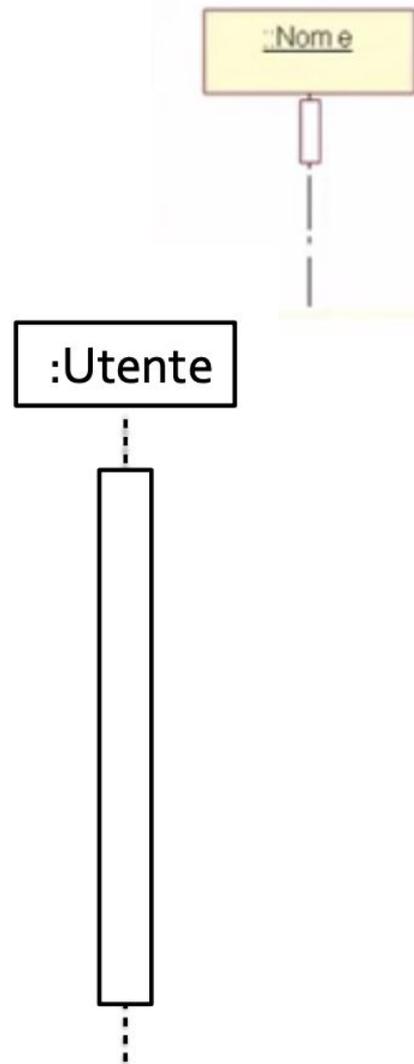
Nel momento in cui definiamo lo stato è necessario instaurare una comunicazione tra i vari oggetti: il diagramma delle sequenze mette in relazione gli oggetti tra loro in base al trascorrere del tempo.

# Diagrammi di sequenza

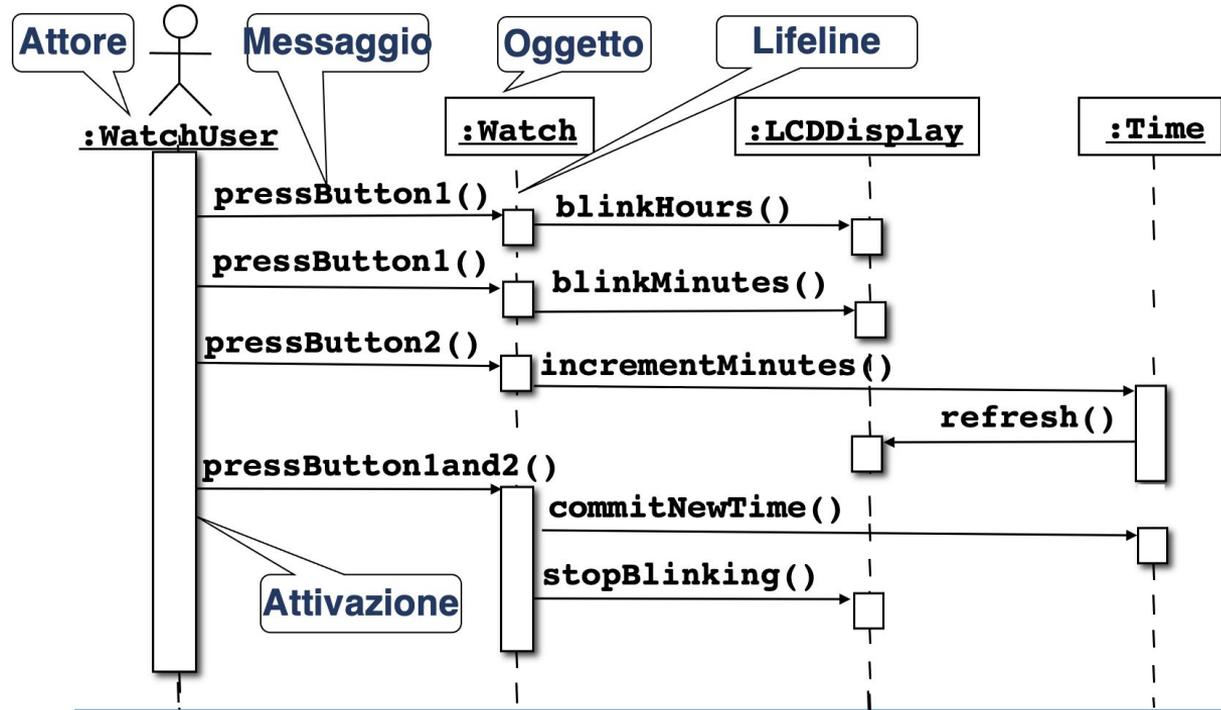
- Si usano:
  - per descrivere le **interazioni**: scambio di **messaggi** e **dati** tra oggetti
    - per esempio un attore e il sistema per la realizzazione di un caso d'uso
    - oppure, in fase di progettazione, i messaggi scambiati tra sottosistemi
  - organizzati in sequenza temporale

## Elementi di un diagramma di sequenza

- Oggetti partecipanti alle interazioni sono rappresentati con linee di vita formate da:
  - un rettangolo, che indica ruolo (nell'interazione) e/o tipo dell'oggetto (uno dei due obbligatorio, entrambi solo se utile)
  - una linea verticale chiamata linea di vita dell'oggetto
    - questa linea è **tratteggiata** quando l'oggetto è **inattivo**,
    - **continua e doppia** quando l'oggetto è **attivo**. Oggetti sempre attivi (es attori) hanno l'intera linea di vita continua e doppia.



# Diagramma delle sequenze



I diagrammi delle sequenze rappresentano il comportamento di un Sistema come messaggi (*interazioni*) tra oggetti differenti

# Diagramma delle sequenze

- Usati
  - Durante l'analisi dei requisiti per rifinire le descrizioni dei casi d'uso e per trovare oggetti aggiuntivi (oggetti partecipanti)
  - Durante la progettazione del sistema per rifinire le interfacce dei sottosistemi
- I diagrammi delle sequenze rappresentano orizzontalmente gli oggetti partecipanti nell'interazione e verticalmente il tempo
- Esempio
  - Un orologio con due pulsanti (2Bwatch)

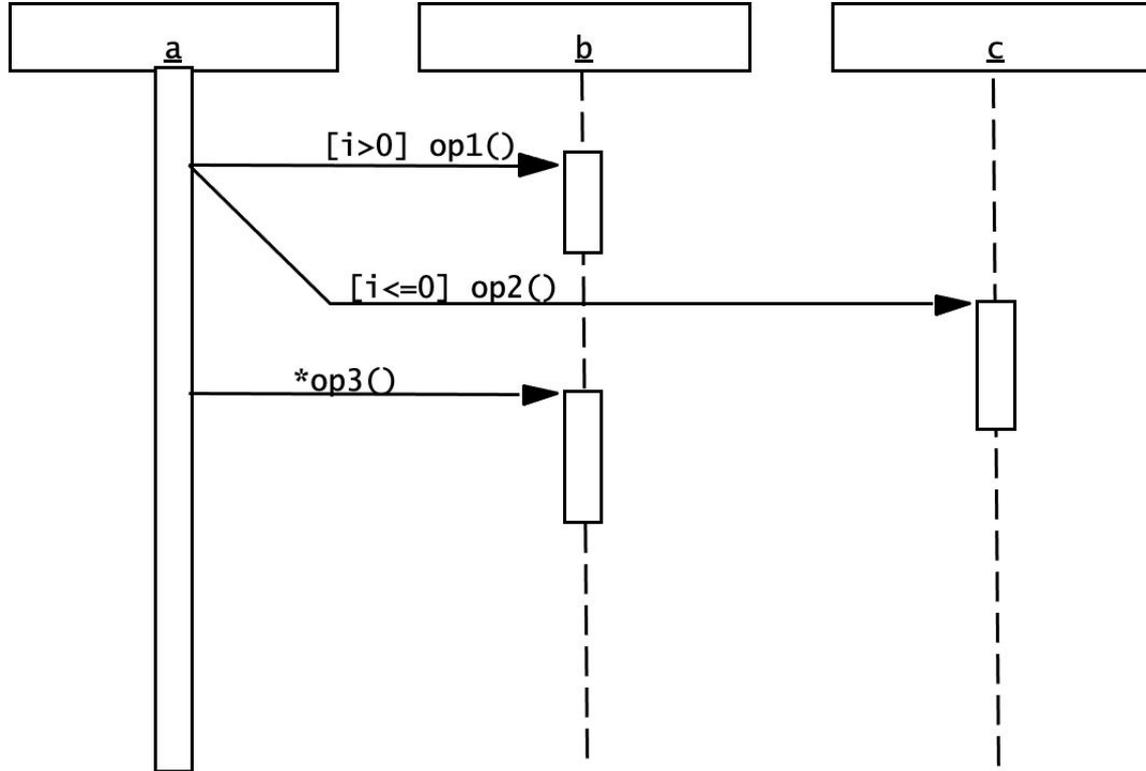
# Notazioni

- Le **colonne** rappresentano gli oggetti che partecipano nell'interazione
- Le **frecce** rappresentano i messaggi
  - Le etichette rappresentano i nomi dei metodi che possono contenere argomenti
- I **rettangoli** verticali rappresentano le attivazioni (esecuzione dei metodi)
- L'attore che inizia l'interazione è rappresentato nella prima colonna a sinistra
- I messaggi provenienti dall'attore rappresentano le interazioni descritte nei diagrammi dei casi d'uso
  - Se altri attori comunicano con il sistema durante il caso d'uso, questi attori sono rappresentati sul lato destro e possono ricevere messaggi

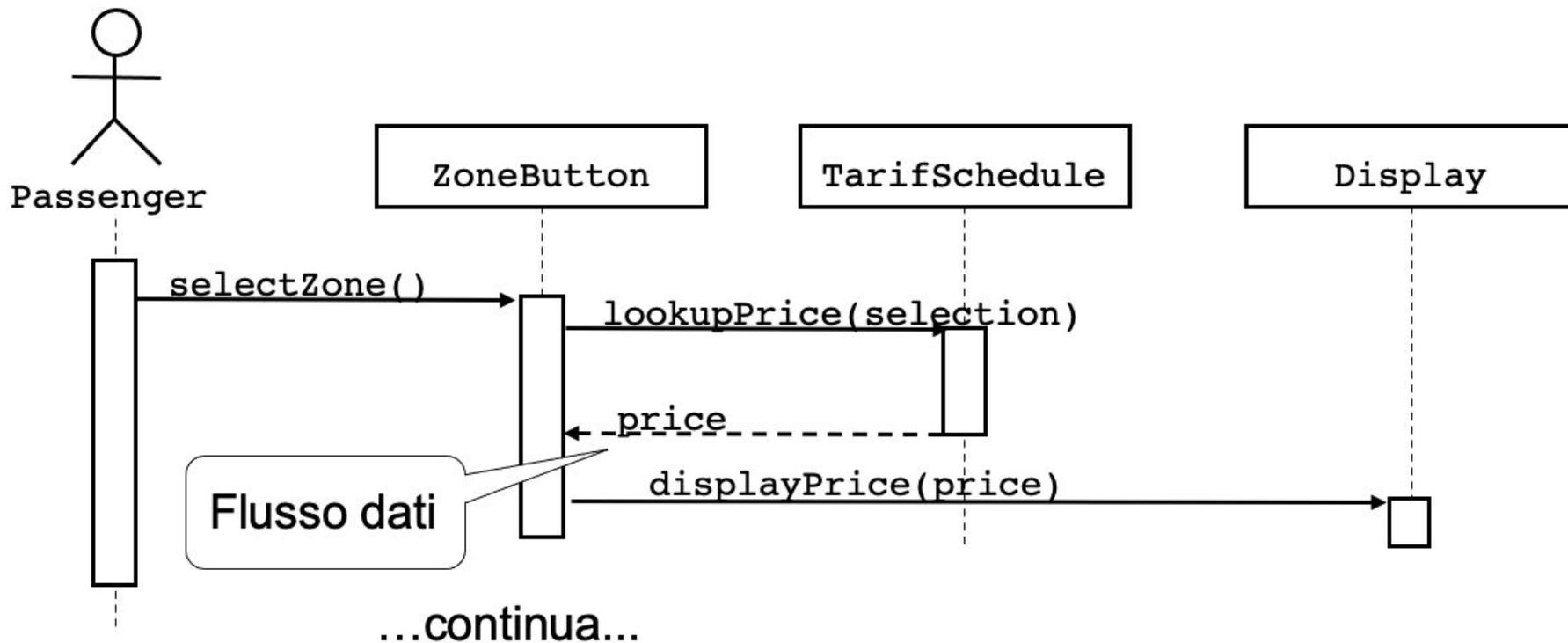
# Condizioni ed iterazioni

- I diagrammi delle sequenze possono essere usati per descrivere una sequenza astratta (tutte le possibili interazioni) o sequenze concrete (una possibile interazione)
- Sono disponibili notazioni per esprimere condizioni o iterazioni (quando si descrivono tutte le possibili interazioni)
  - Una **condizione** su un messaggio è rappresentata da una espressione tra parentesi quadre prima del nome del messaggio. Se la condizione è vera il messaggio è inviato
  - Una invocazione **ripetitiva** di un messaggio è denotata da un '\*' prima del nome del messaggio

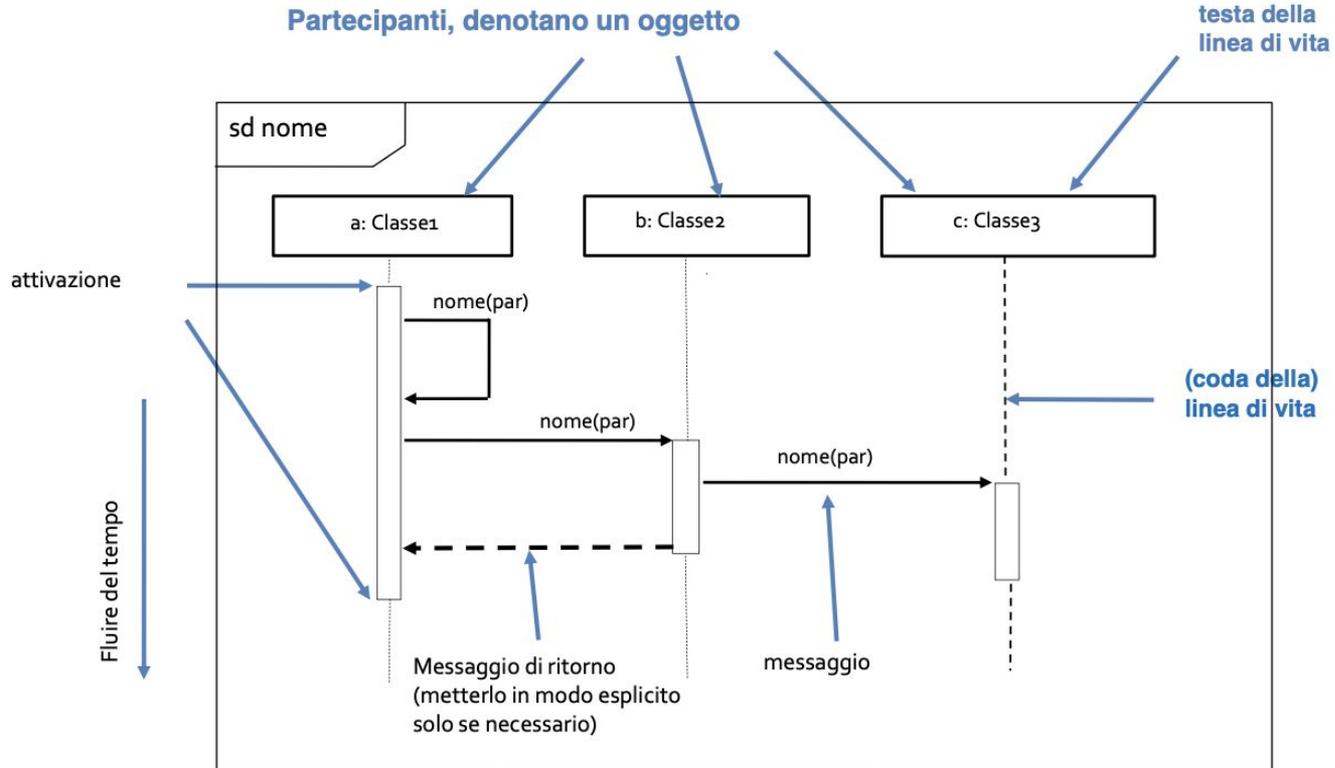
# Esempi di condizioni ed iterazioni



# Diagrammi delle sequenze: flusso di dati



# Diagrammi di sequenza



Messaggi scambiati, l'ordine cronologico è dall'alto in basso

# I messaggi: rappresentano invocazione di operazione o segnali

- Possono essere

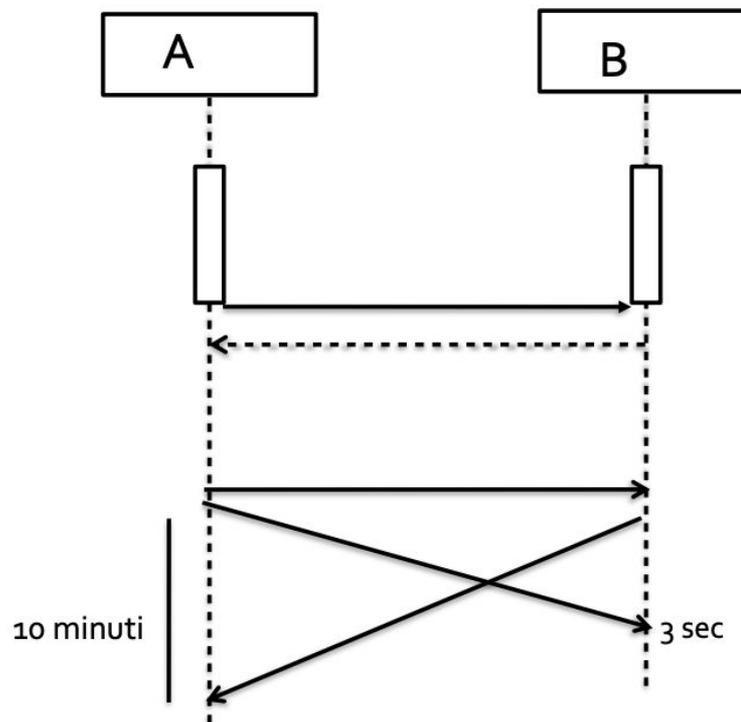
- sincroni

(es comunicazione diretta)

- di return (opzionali)

- asincroni

(es invio email)



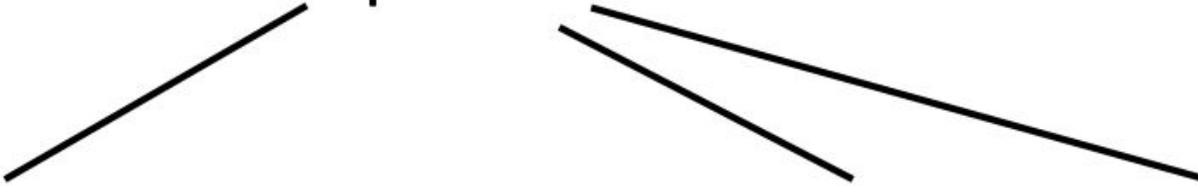
- eventualmente con esplicito consumo di tempo

# I messaggi possono essere

Simple	Rappresenta il trasferimento del controllo da un oggetto a un altro	
Sincrono	Se un oggetto invia un messaggio sincrono, allora si attende che gli venga restituita una risposta al messaggio stesso prima di poter continuare con altre operazioni	
Asincrono	Diversamente dai messaggi sincroni, se un oggetto invia un messaggio asincrono, non attende che gli venga inviata alcuna risposta prima di continuare con altre operazioni	
Ricorsivo	Se il ricevente è anche il mittente	

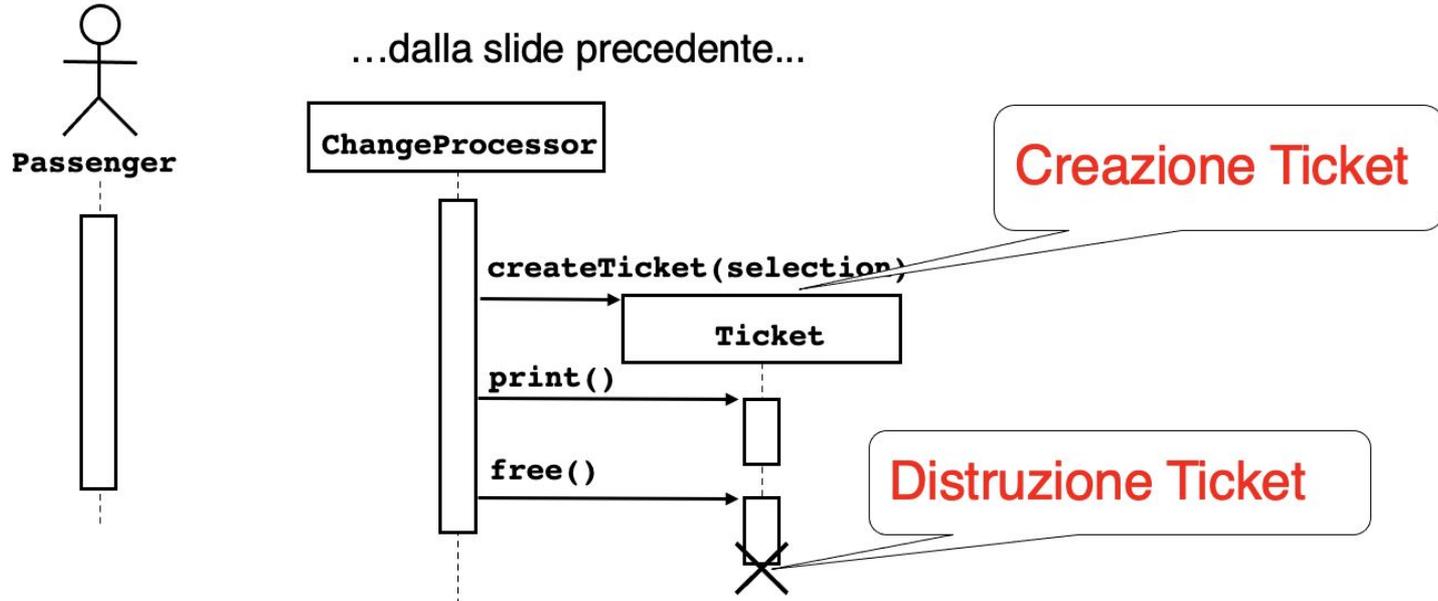
## Sintassi dei messaggi

opzionali



attributo = nomeMessaggio(arg1, arg2, ...) : valore di ritorno

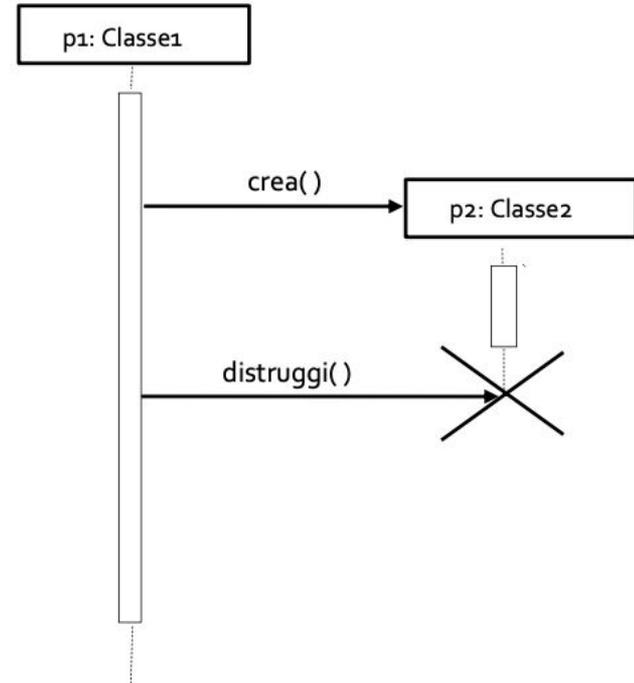
# Creazione e distruzione



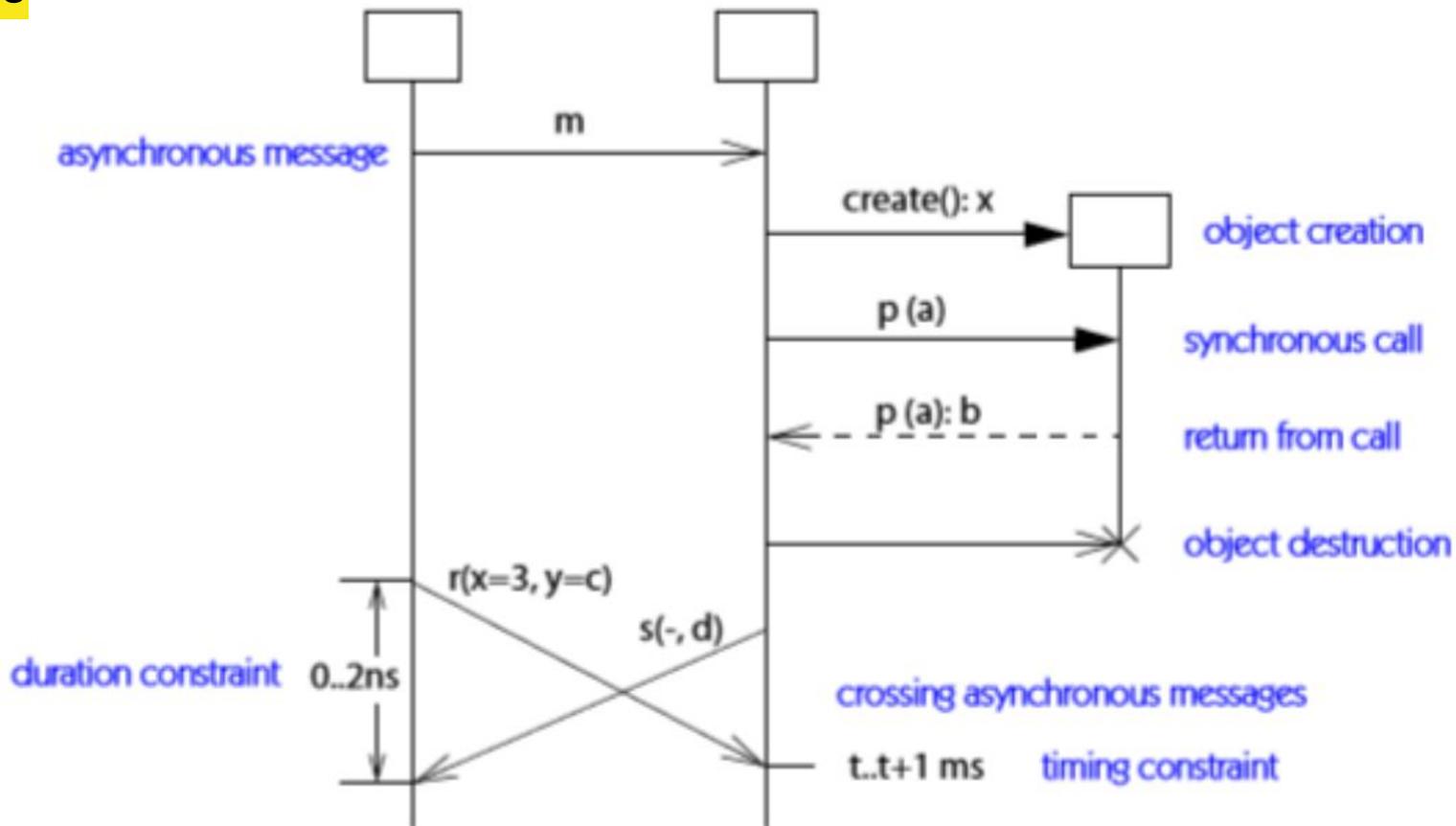
- La creazione è denotata da un messaggio freccia che punta all'oggetto
- La distruzione è denotata da una x alla fine dell'attivazione

# Creare e distruggere partecipanti

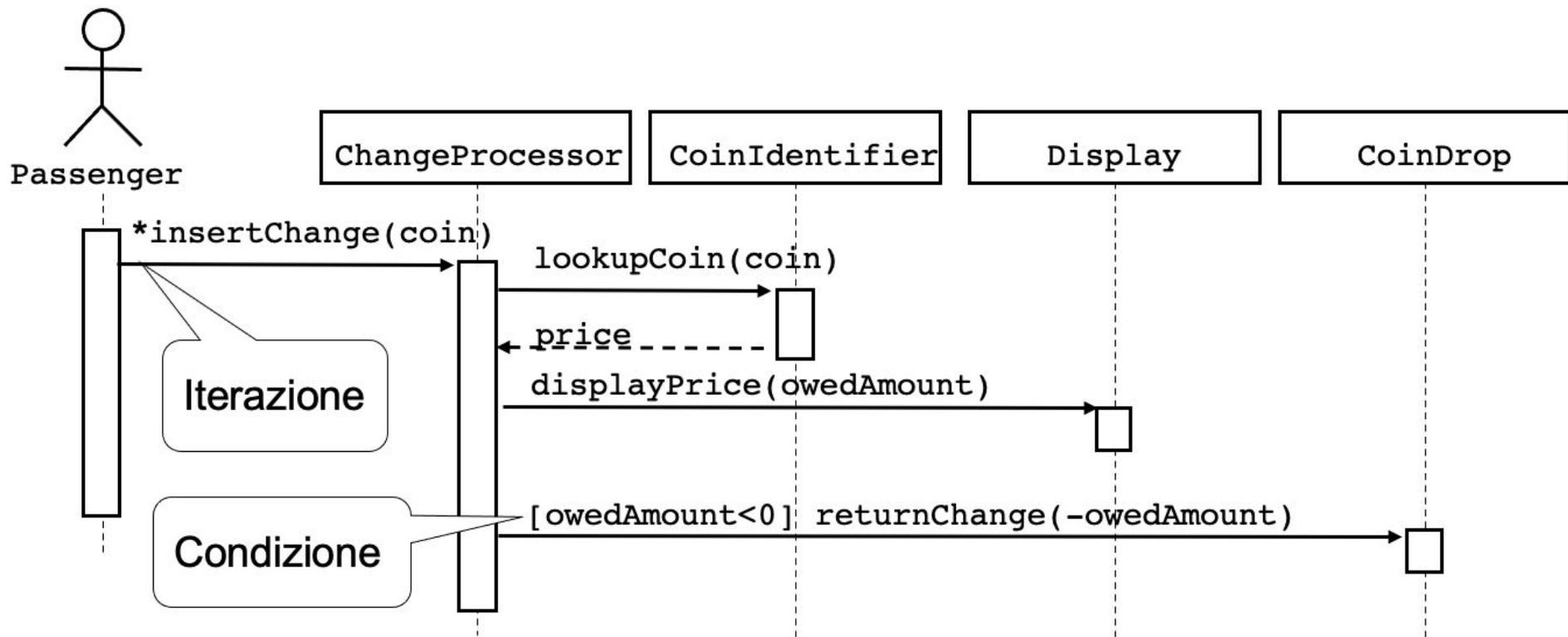
- Alcuni partecipanti possono essere
  - aggiunti dinamicamente all'interazione
  - cancellati



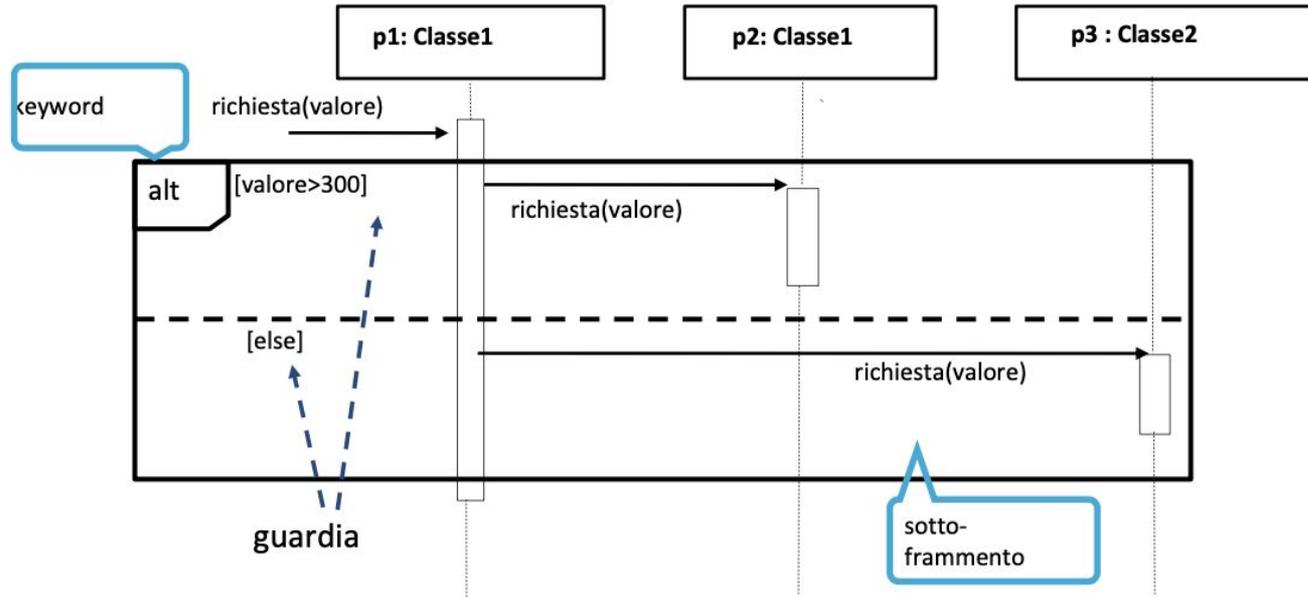
# Esempio



# Iterazioni e condizioni

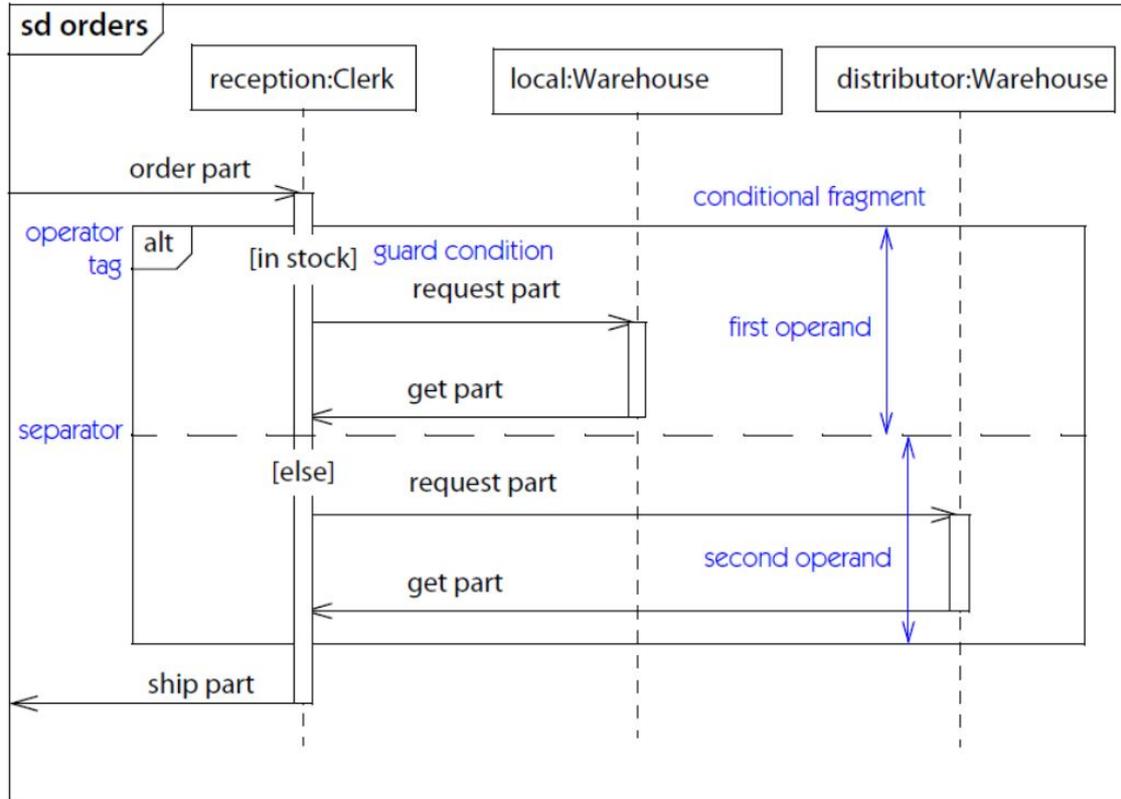


# Frame condizionale



- senza guardia = [true]
- più guardie vere: scelta non-deterministica
- tutte le guardie false: il frame viene saltato

# Frame condizionale : un altro esempio



# Proprietà diagrammi delle sequenze

- Rappresentano il **comportamento in termini di interazioni**
- Utile per identificare o trovare oggetti mancanti
- Comporta tempo per la preparazione, ma vale l'investimento
- Complementari al diagramma delle classi (che rappresenta la struttura)

# Esempio: distributore self service

