

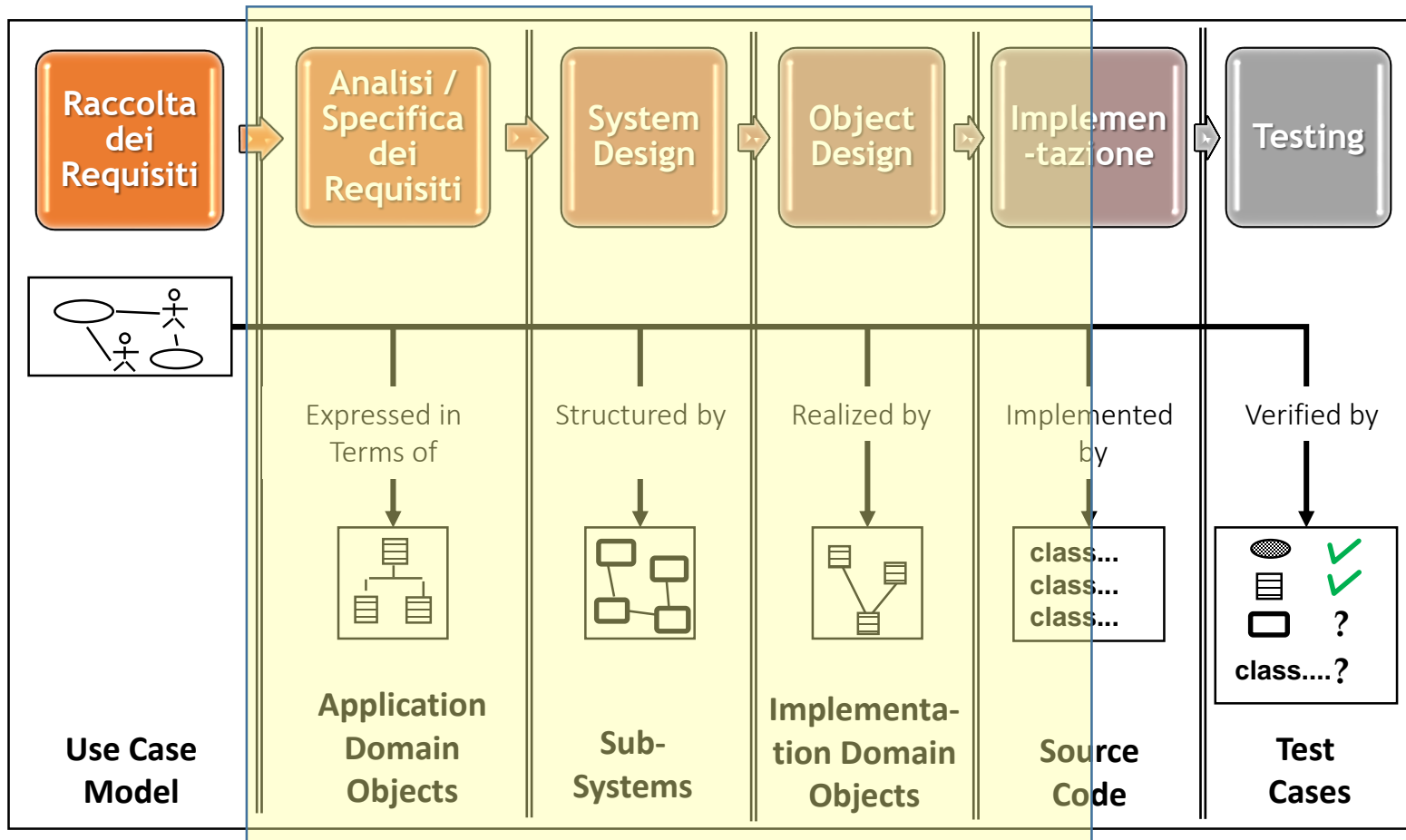


UNIVERSITÀ DEGLI STUDI DI NAPOLI  
**FEDERICO II**

# Ingegneria del Software – Gli Activity Diagrams

Prof. Sergio Di Martino

# Ciclo di Vita del Software



# Gli Activity Diagrams

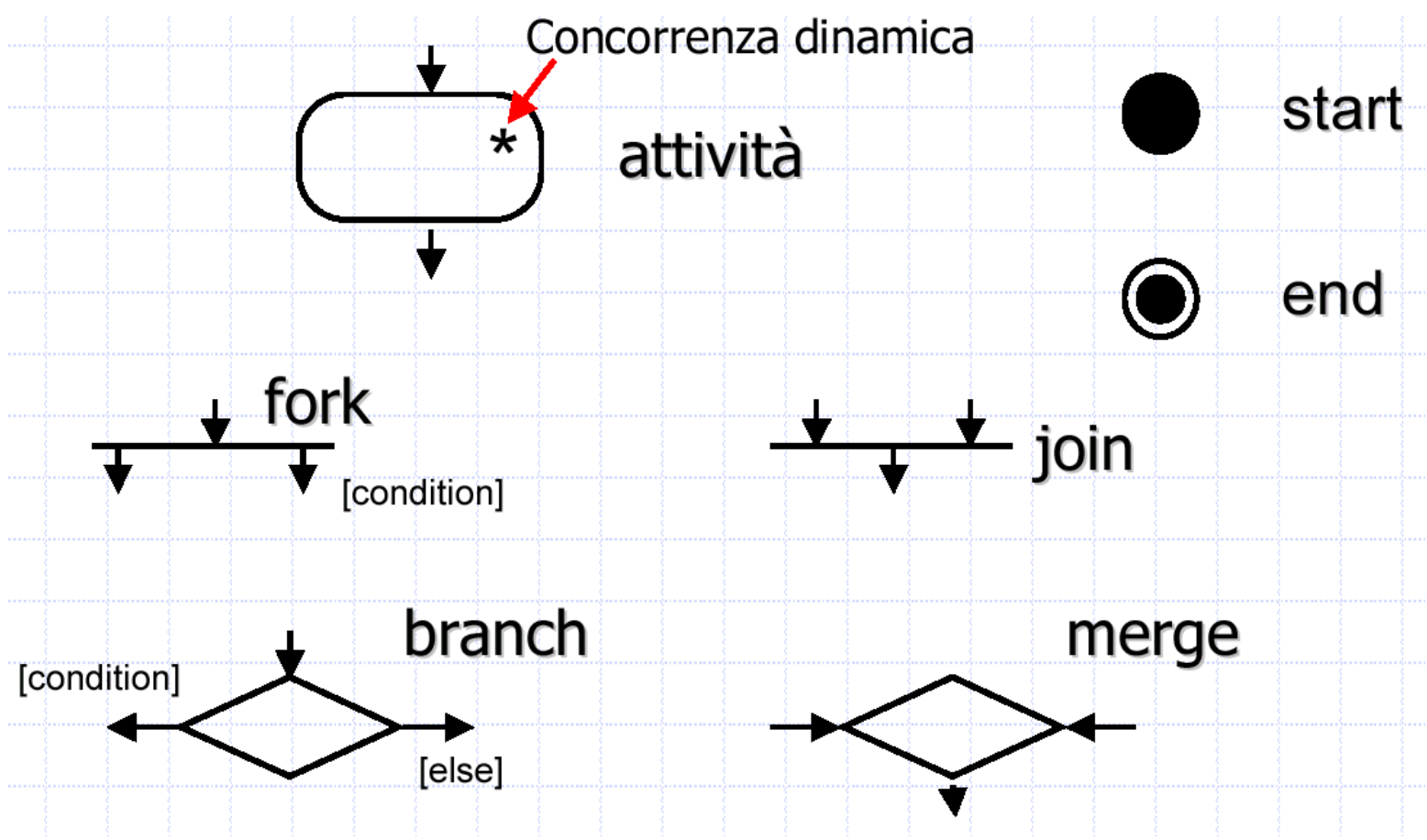
# Activity Diagram (1)

- Forniscono la sequenza di operazioni che definiscono un'attività più complessa
- Permettono di rappresentare processi paralleli e la loro sincronizzazione
- Un Activity Diagram può essere associato
  - Ad uno Use Case
  - A una classe
  - All'implementazione di un'operazione
- Retaggio della scomposizione “funzionale”

# Activity Diagram (2)

- Derivano da event diagrams, SDL e reti di Petri
- Servono a rappresentare sistemi di workflow, oppure la logica interna di un processo di qualunque livello
- Utili per modellare
  - comportamenti sequenziali
  - non determinismo
  - concorrenza
  - sistemi distribuiti
  - business workflow
  - operazioni

# Elementi Grafici

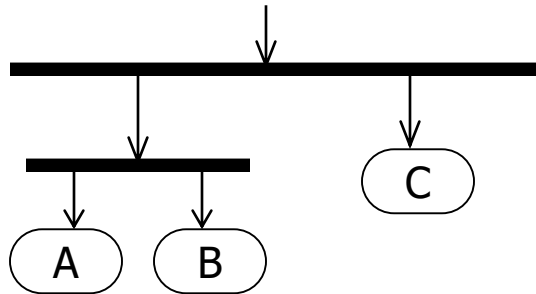


*Le attività possono essere gerarchiche*

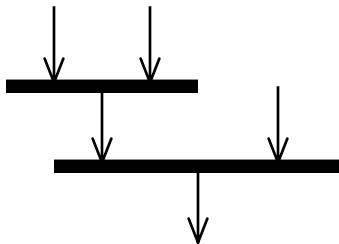
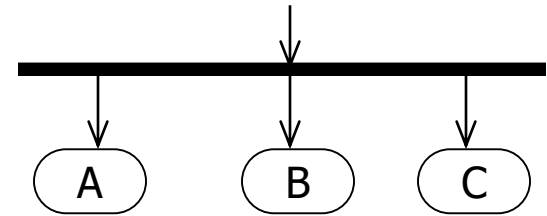
# Activity Diagram: Elementi

- **Activity:** una esecuzione non atomica entro uno state machine
  - Una activity è composta da action, elaborazioni atomiche comportanti un cambiamento di stato del sistema o il ritorno di un valore
- **Transition:** flusso di controllo tra due action successive
- **Guard expression:** espressione booleana (condition) che deve essere verificata per attivare una transition
- **Branch:** specifica percorsi alternativi in base a espressioni booleane; un branch ha una unica transition in ingresso e due o più transition in uscita
- **Synchronization bar:** usata per sincronizzare flussi concorrenti
  - fork: per splittare un flusso su più transition verso action state concorrenti
  - join: per unificare più transition da più action state concorrenti in una sola
    - il numero di fork e di join dovrebbero essere bilanciati

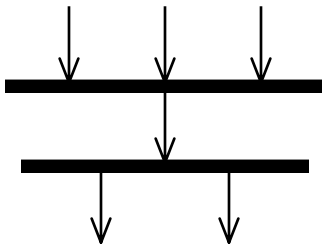
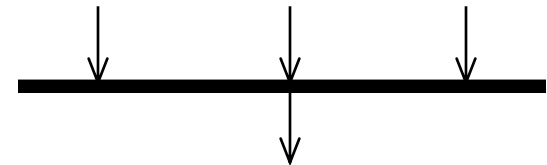
# Combinazione di Fork e Join



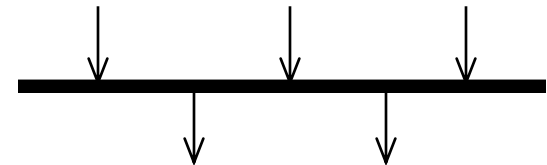
*equivalente a*



*equivalente a*

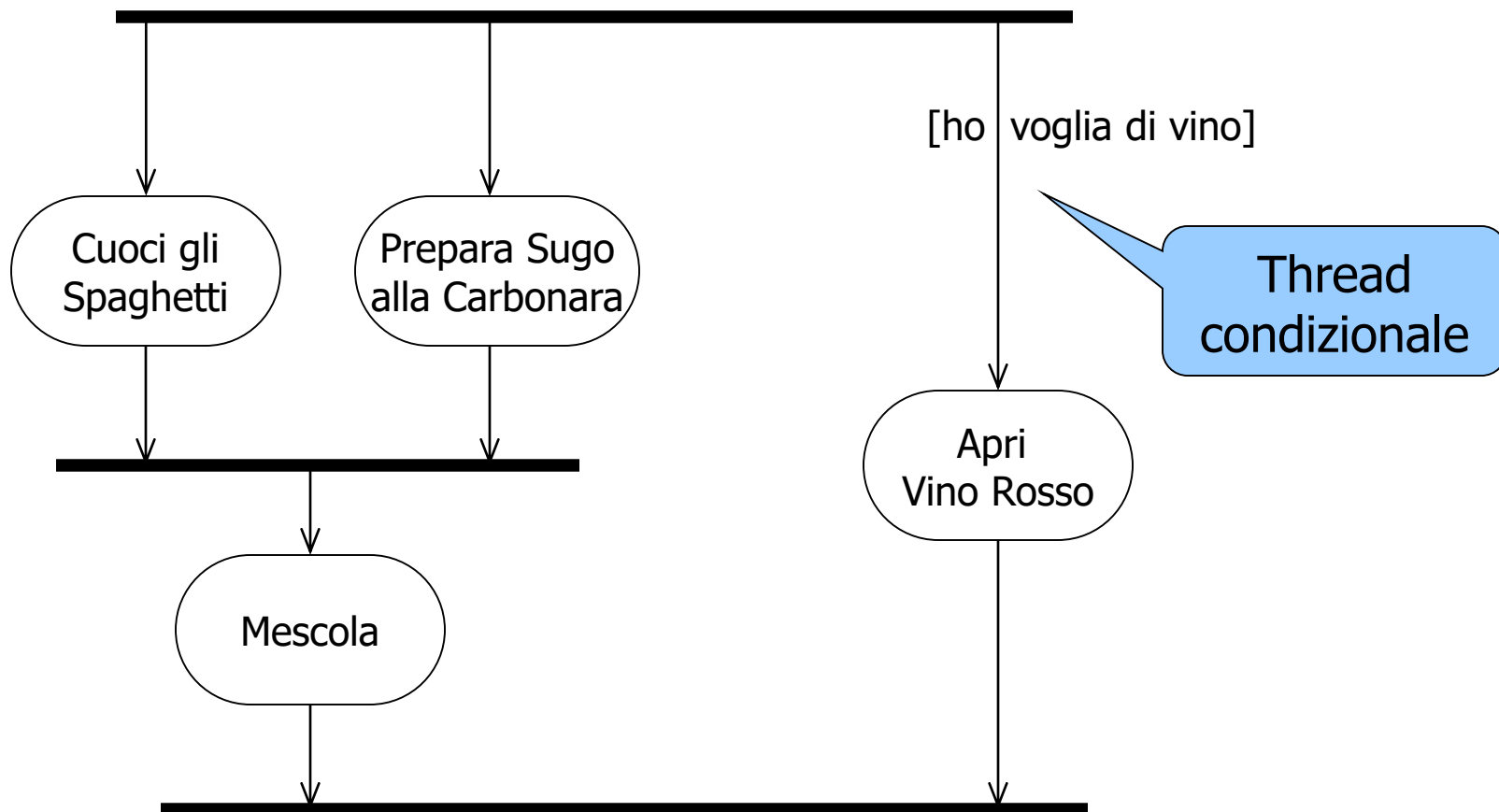


*equivalente a*



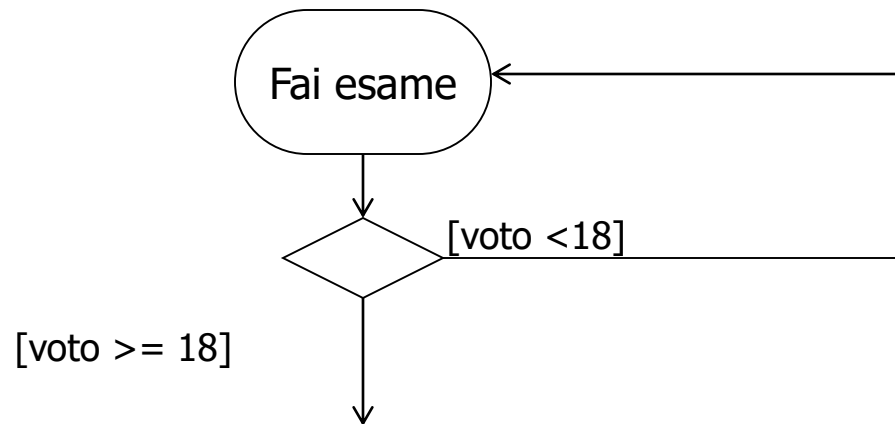


# Thread condizionale

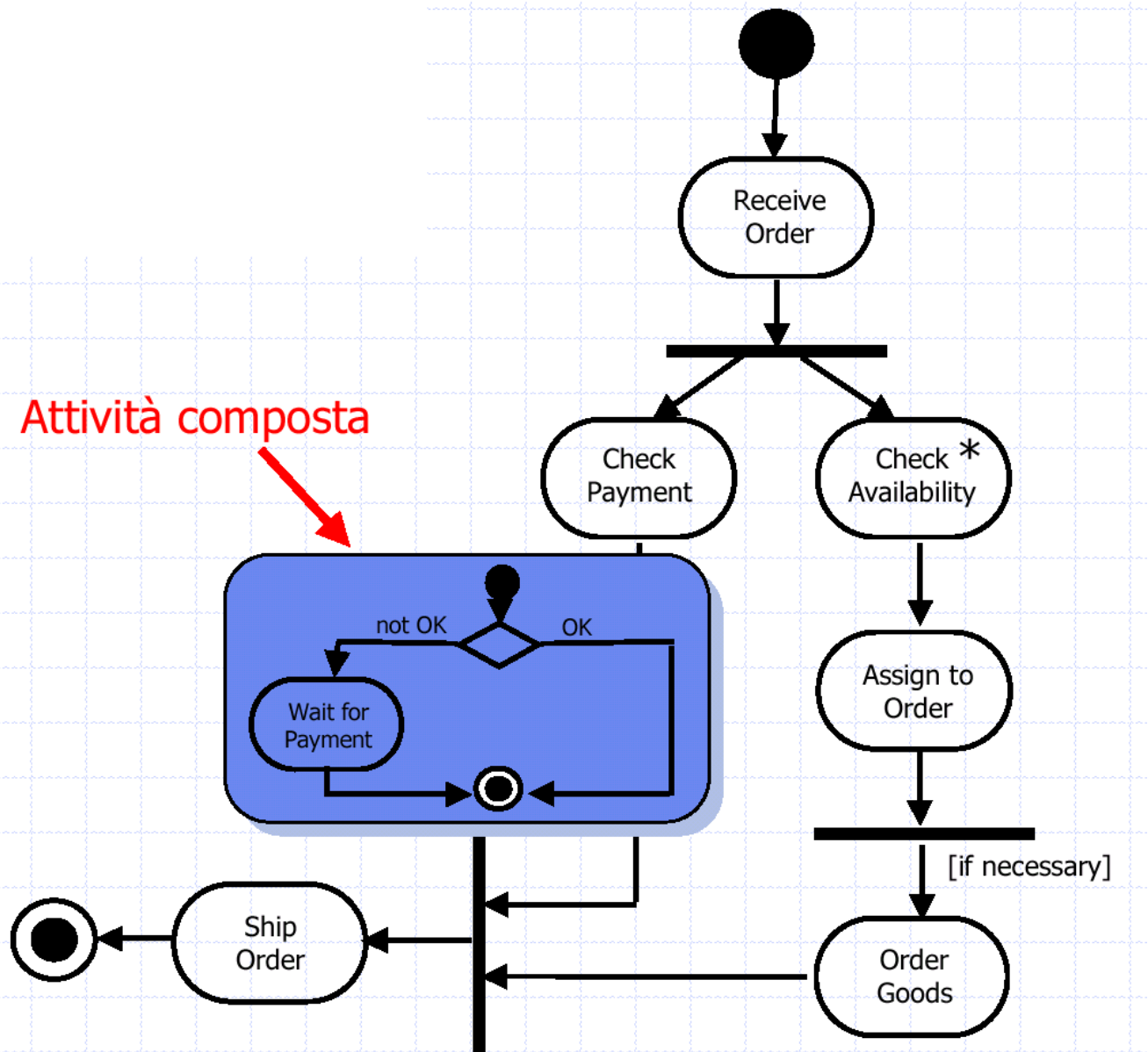


# Cicli

- Esempio: fare l'esame di Ingegneria del Software fino a quando il voto non è sufficiente



# Esempio: Gestione Ordini



# Swimlanes

- Costrutto grafico rappresentante un insieme partizionato di action/activity;
- Identificano le responsabilità relative alle diverse operazioni
  - Parti di un oggetto
  - Oggetti diversi
- In un Business Model identificano le unità organizzative
- Per ogni oggetto responsabile di action/activity nel diagramma è definito un swimlane, identificato da un nome univoco nel diagramma
  - le action/activity state sono divise in gruppi e ciascun gruppo è assegnato allo swimlane dell'oggetto responsabile per esse
  - l'ordine con cui gli swimlane si succedono non ha alcuna importanza
  - le transition possono attraversare swimlane per raggiungere uno state in uno swimlane non adiacente a quello di start della transition

# Esempio: Swimlanes

