## **UML:** Diagrammi di interazione

- → Sequence
- → Commmunication
- → modellazione dinamica (interazione + behaviour)
- → descrizione della collaborazione in un gruppo di oggetti
  - → scambio di messaggi:
    - → sincroni: mittente invia e attende risposta
    - → asincroni: mittente invia e basta (processi concorrenti)
      - → differente notazione UML
  - → per mandare un messaggio deve esistere una relazione tra i due comunicanti
    - → messaggi tipo chiamate di operazioni

## **Sequence Diagram**

- → notazione: partecipanti, linee di vita, messaggi a-/sincroni, messaggio trovato (da fuori, non specificato), barre di attivazione (esecuzione o passaggio controllo a altra operazione), ritorni (opzionali)
- → tempo discendente verticalmente
- → interpretabilità come codice facilitata:
  - → partecipanti come classi, messaggi come metodi (parametri come esistenza di barra di attivazione)
- → messaggi: return\_val = message ( params ) : return\_type (magenta opzionali)
  - → valori di ritorno opzionali nel diagramma se appesantiscono troppo
  - → anche self: chiamata di operazione appartenente alla stessa classe
- → **creazione** partecipanti
  - → messaggio (freccia) verso rettangolo partecipante nuovo: creazione sul posto
- → distruzione partecipanti:
  - → fine linea di vita temporale: X a simboleggiare terminazione
- → alternative di esecuzione?
  - → frame di esecuzione (rischio dimensionale) oppure diagrammi separati
- → **frame** di interazione:
  - → rappresentazione di cicli e condizioni
  - → notazione: "cornice" per isolare parte di programma
    - $\rightarrow$  operatore: a cosa serve il frame

```
es.: loop (for), opt (if-then), alt (if-then-else), ...
```

- → guardia: diversa per diversi tipi di operatori
  - es.: su cosa si loopa, condizione da verificare, ...
- → possibilità di annidare frames (tipo loop e poi condizioni interne, ...)
- → utilizzo a livello di:
  - → requisiti (SSD): rappresentazione alternativa di uno use case
  - → design (SD): più usati; partecipanti ≡ classi OO

## **Communication Diagram**

- → abbastanza ridondante, comunque enfasi a comunicazione tra oggetti
- → rappresenta sequenze di chiamate tramite numerazione