

# UML: Sequence diagram

Gianluca Pironato

# Sequence diagram in UML

- ▶ Mostrano come diversi componenti interagiscono tra loro in un ordine temporale, che a loro volta possono essere intesi come parti di un sistema oppure come classi, nel caso in cui si modelli direttamente del codice.
- ▶ Evidenziano in dettaglio la sequenza di messaggi/eventi/chiamate a metodi così da facilitare l'analisi del comportamento dinamico di un sistema.
- ▶ Vengono utilizzati per documentare processi e comprendere i requisiti di un programma, al di là dei linguaggi di programmazione scelti.
- ▶ Possono aiutare anche nell'individuazione di casi d'uso (es. mi accorgo che non gestisco alcune condizioni) o, al contrario, nella comprensione a fondo dei meccanismi dietro a un certo caso d'uso (es. prima di passare alla codifica).

# Elementi base

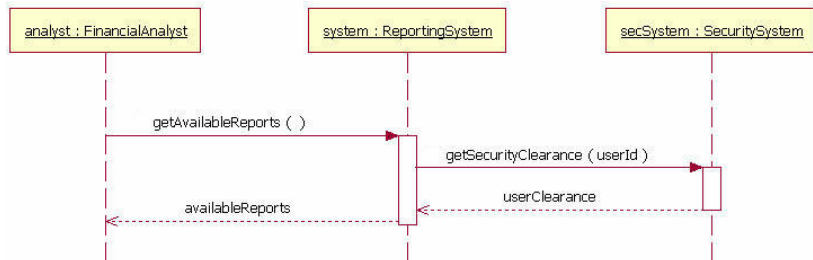
- ▶ **Attori e/o oggetti:**
  - ▶ gli attori vengono rappresentati con figure umane stilizzate e sono entità esterne che interagiscono con il sistema (come quelli dello use-case diagram);
  - ▶ gli oggetti vengono invece rappresentati con rettangoli e sono le componenti interne del sistema o le classi di codice.
  - ▶ se il diagramma deve rappresentare istanze specifiche di certe classi e non le classi in generale allora si pone nel rettangolo il nome dell'istanza e la si sottolinea.
- ▶ **Lifeline:** linea tratteggiata verticale che rappresenta l'esistenza di un attore o oggetto nel tempo; leggere il diagramma scendendo lungo la lifeline equivale a visualizzare il passare del tempo.

# Messaggi

- ▶ Esprimono le comunicazioni o le interazioni (più generale) tra attori e oggetti.
- ▶ Esistono due tipi principali di messaggi:
  - ▶ **messaggi di invio**: rappresentati da freccia con linea continua, indicano l'invio di una richiesta o l'esecuzione di un'azione (sincrona se con punta della freccia robusta, asincrona se con punta della freccia stilizzata);
  - ▶ **messaggi di ritorno**: rappresentati da freccia con linea tratteggiata, indicano le risposte o restituzioni di informazioni.
- ▶ I messaggi vengono disposti lungo le lifeline in ordine temporale, dall'alto verso il basso e da sinistra verso destra.
- ▶ Se un'azione essenziale da indicare nel diagramma prende luogo nel contesto di un solo attore, la freccia corrispondente risulterà essere un cappio sulla lifeline dell'attore.

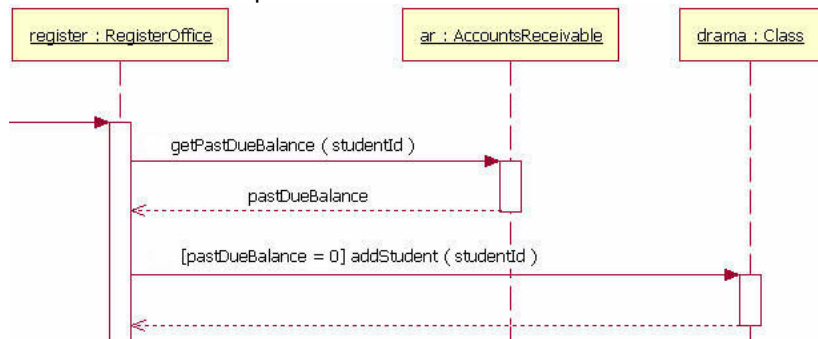
# Messaggi

**N.B.** un diagramma come questo ci permette, per esempio, di dire che `getAvailableReports()` è un metodo della classe `ReportingSystem`.



# Messaggi condizionali

Vedasi l'uso della guardia sul messaggio che viene inviato solo se rispettata una certa condizione.



# Box di attivazione

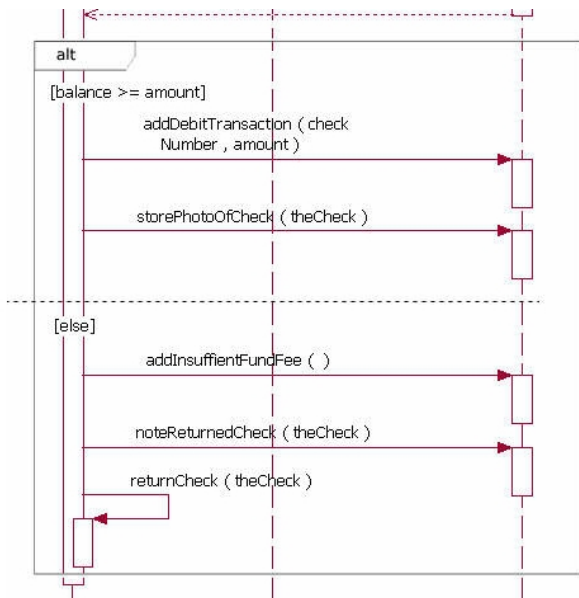
- ▶ Gli **activation boxes** evidenziano quando e per quanto tempo un oggetto è attivo, ovvero sta eseguendo un processo (invia e/o riceve).
- ▶ Vengono disegnati lungo le lifeline e mostrano visivamente l'intervallo in cui un oggetto elabora un messaggio.
- ▶ Questo aiuta a capire il tempo di esecuzione e l'ordine delle operazioni.

# Frame alt

- ▶ In situazioni in cui il flusso di messaggi varia a seconda di condizioni, si utilizzano i **frame alternativi**.
- ▶ Un frame alternativo delimita due scenari mutuamente esclusivi, ad esempio:
  - ▶ se la carta è valida, allora l'atm chiede il pin;
  - ▶ se la carta non è valida, allora l'atm espelle la carta.
- ▶ L'uso di frame alternativi permette di visualizzare le decisioni e di ramificare i flussi di esecuzione nel diagramma.

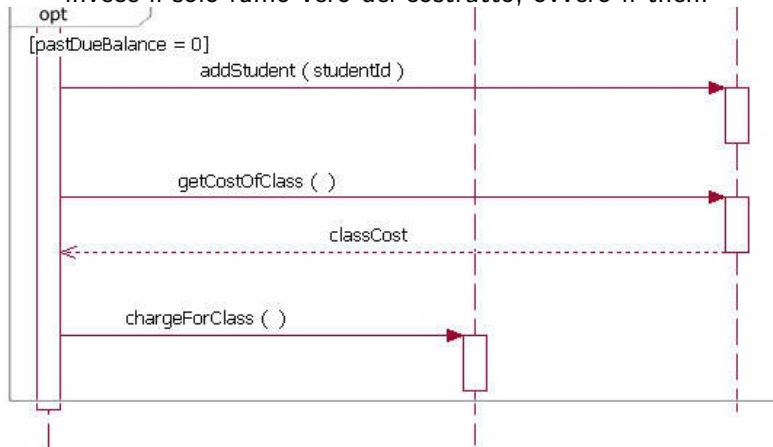


## Frame alt



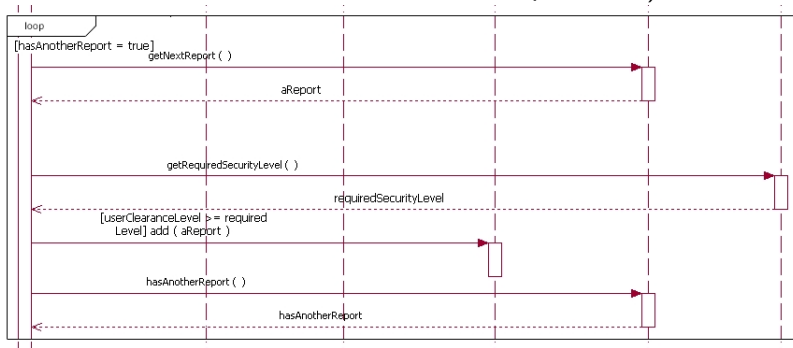
## Frame opt

Se il frame alt rappresenta il costrutto if-then-else, il frame opt è invece il solo ramo vero del costrutto, ovvero if-then.



# Frame loop

Infine, anche il costrutto di iterazione è disponibile (nella documentazione UML si trovano più frame).



# Ulteriori dettagli

- ▶ I diagrammi di sequenza possono rappresentare interazioni molto complesse:
  - ▶ le etichette dei messaggi possono rappresentare chiamate a metodi o funzioni specifiche;
  - ▶ possono essere utilizzati per modellare interazioni asincrone o sincrone;
  - ▶ possono essere aggiunti note e commenti per chiarire il funzionamento di particolari scambi.
- ▶ Un diagramma chiaro permette di comprendere il flusso di informazioni e identificare eventuali colli di bottiglia o errori.

# Conclusioni

- ▶ I diagrammi di sequenza in uml sono strumenti potenti per visualizzare aspetti comportamentali del sistema.
- ▶ Si possono utilizzare per fare documentazione di processi, per analizzare i requisiti di un sistema e comunque in diverse fasi del ciclo di vita del software.
- ▶ Comprendere la notazione è essenziale per progettare e comunicare soluzioni software efficaci.

- ▶ Diagrammi di sequenza, IBM <https://www.ibm.com/docs/en/rsm/7.5.0?topic=uml-sequence-diagrams>
- ▶ Dettaglio notazioni diagrammi di sequenza, IBM <https://developer.ibm.com/articles/the-sequence-diagram/>