



# DIAGRAMMI DI SEQUENZA

## INGEGNERIA DEL SOFTWARE

Università degli Studi di Padova

Dipartimento di Matematica

Corso di Laurea in Informatica, A.A. 2017 – 2018

rcardin@math.unipd.it

# SOMMARIO

- o Introduzione
- o Partecipanti e messaggi
- o Concetti avanzati

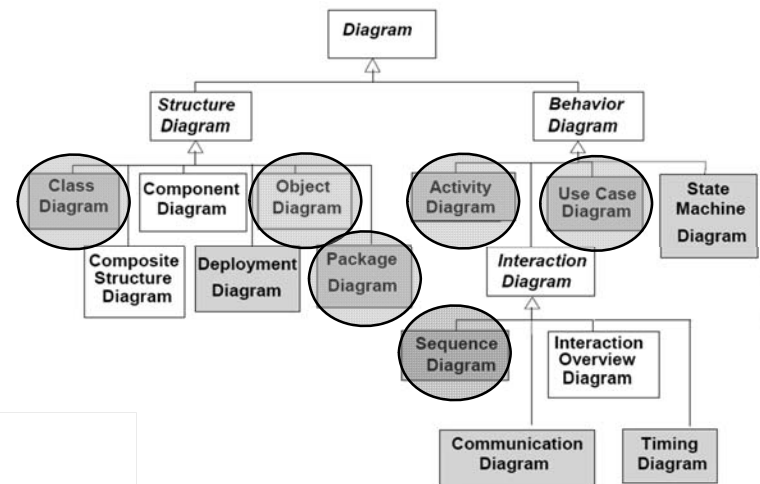


# SOMMARIO

- o Introduzione
- o Partecipanti e messaggi
- o Concetti avanzati

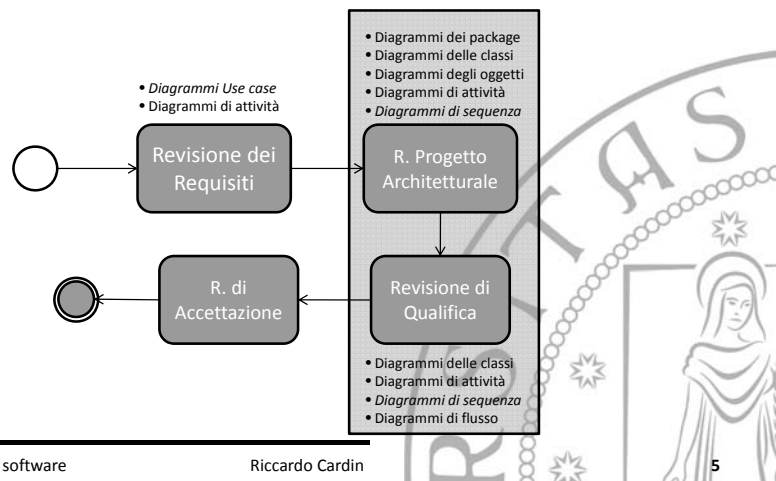


# DIAGRAMMI DI SEQUENZA



# DIAGRAMMI DI SEQUENZA

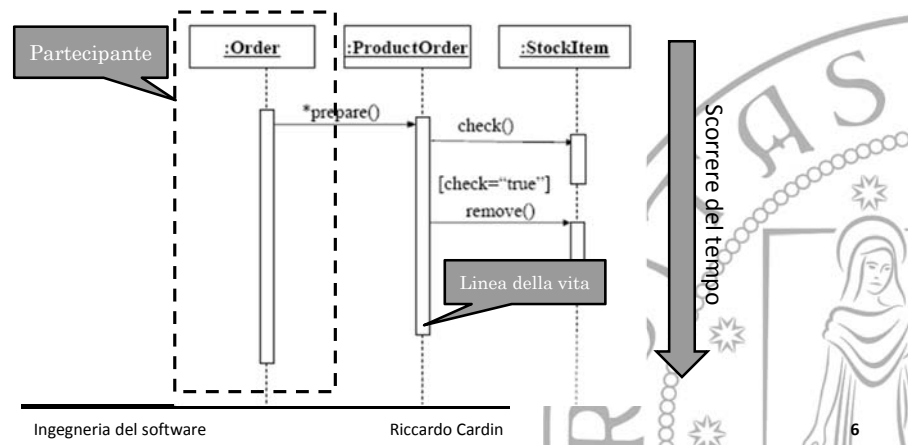
## o Specifica Tecnica, Definizione di Prodotto



# DIAGRAMMI DI SEQUENZA

## Definizione

Descrivono la collaborazione di un gruppo di oggetti che devono implementare collettivamente un comportamento



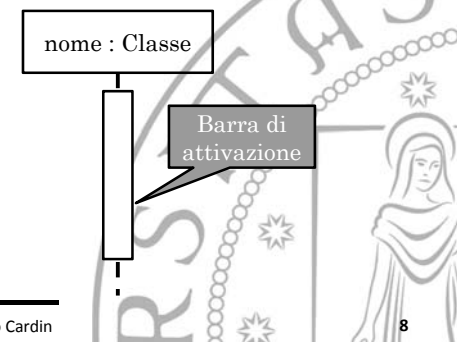
# SOMMARIO

- o Introduzione
- o Partecipanti e messaggi
- o Concetti avanzati

# PARTECIPANTI

## o Entità che detengono il flusso del caso d'uso

- UML 1.x → Istanze di classi (oggetti)
- UML 2.x → Concetto più lato
  - o Eliminata la sottolineatura
- Barra di attivazione
  - o Indica in quale momento un partecipante è attivo
  - o Opzionale, ma molto utile

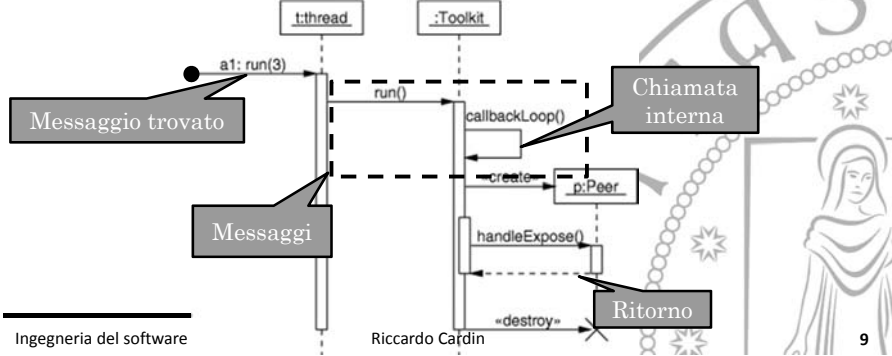


```
attribute = signal_name (args) : return_type
args = <name> : <class>
```

## MESSAGGI (SEGNALI)

### o Dati e operazioni scambiati tra i partecipanti

- Chiamata a metodi degli oggetti
- Messaggio trovato
  - o Primo messaggio che scaturisce dall'esterno



## MESSAGGI (SEGNALI)

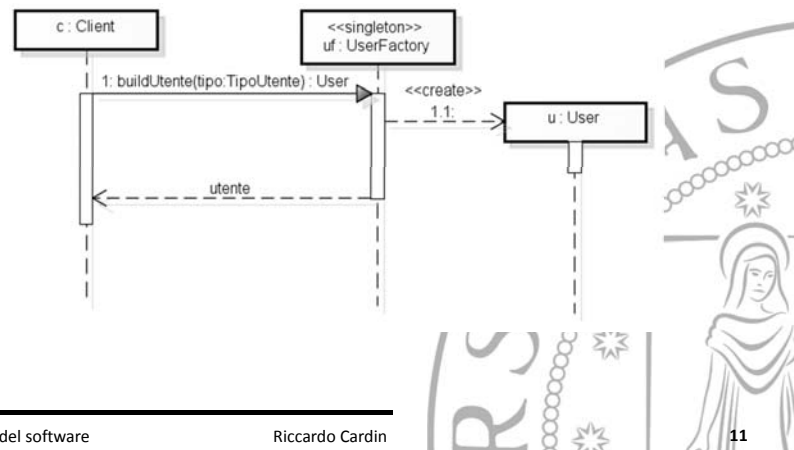
### o Esempio principale

#### Esempio

È richiesto lo sviluppo di un'applicazione che permetta la gestione di un semplice blog. In particolare devono essere disponibili almeno tutte le funzionalità base di un blog: deve essere possibile per un utente inserire un nuovo post e successivamente per gli altri utenti deve essere possibile commentarlo. Queste due operazioni devono essere disponibili unicamente agli utenti registrati all'interno del sistema. La registrazione avviene scegliendo una username e una password. La username deve essere univoca all'interno del sistema.

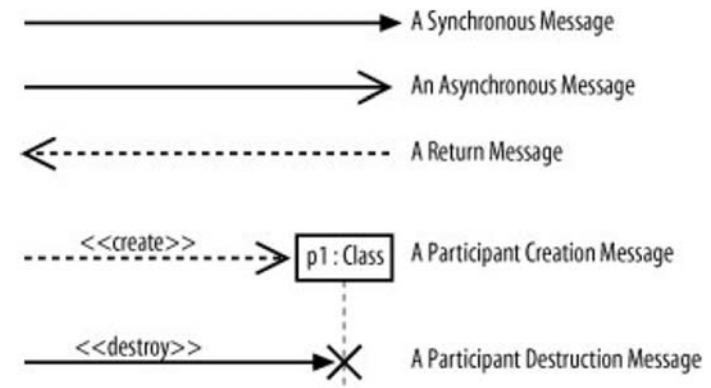
## MESSAGGI (SEGNALI)

### o Esempio 1



## MESSAGGI (SEGNALI)

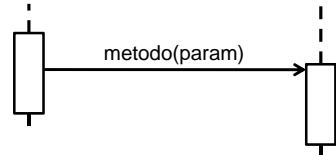
### o Tipologie



# MESSAGGI (SEGNALI)

## o Passaggio di dati

- Nessuna tecnica di modellazione standard!!!
- Metodo classico



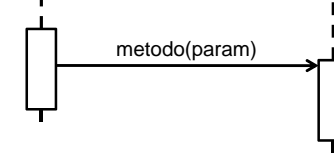
- Girini dei dati (*data tadpoles*)



# MESSAGGI (SEGNALI)

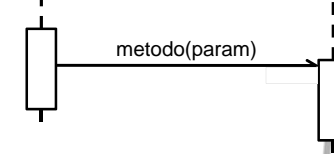
## o Messaggi sincroni

- Il chiamante rimane in attesa della risposta



## o Messaggi asincroni

- Il chiamante non rimane in attesa della risposta

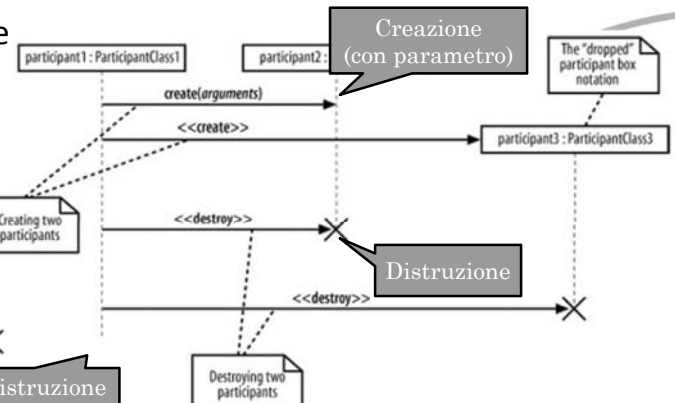


# MESSAGGI (SEGNALI)

## o Creazione partecipanti

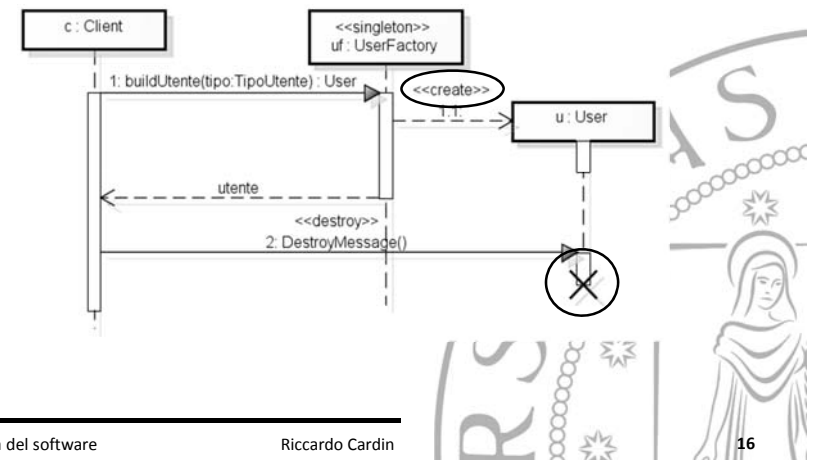
- Comodo l'utilizzo della parola "new"

## o Distruzione



# MESSAGGI (SEGNALI)

## o Esempio 2



## SOMMARIO

- Introduzione
- Partecipanti e messaggi
- Concetti avanzati

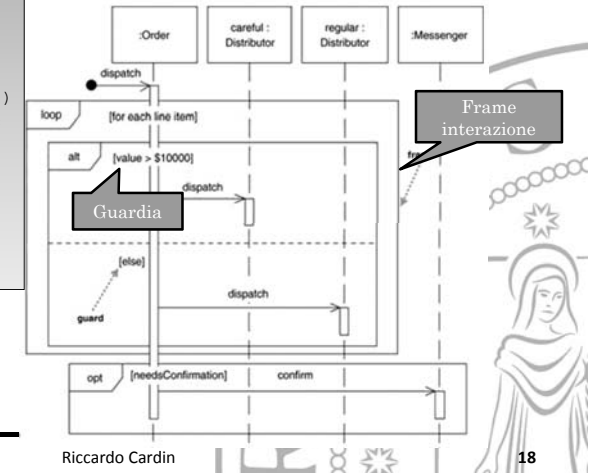


## CICLI E CONDIZIONI

- Frame di interazione (UML 2)

Pseudocodice

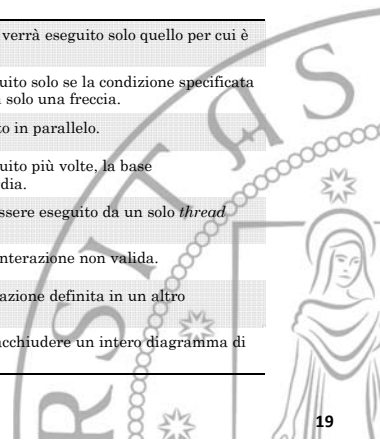
```
procedura spedizione
foreach (elementoLinea)
  if (prodotto.valore > 10K)
    raccomandata.spedizione
  else
    normale.spedizione
  end if
end for
if (necessitaConferma)
  messenger.conferma
end procedura
```



## CICLI E CONDIZIONI

- Frame di interazione

Operatore	Significato
alt	Frammenti multipli in alternativa; verrà eseguito solo quello per cui è verificata la condizione.
opt	Opzionale; il frammento viene eseguito solo se la condizione specificata è verificata. Equivalente a alt con solo una freccia.
par	Parallelo; ogni frammento è eseguito in parallelo.
loop	Ciclo; il frammento può essere eseguito più volte, la base dell'iterazione è indicata dalla guardia.
region	Regione critica; il frammento può essere eseguito da un solo thread alla volta.
neg	Negativo; il frammento mostra un'interazione non valida.
ref	Riferimento; si riferisce ad un'interazione definita in un altro diagramma
sd	Sequence diagram; utilizzato per racchiudere un intero diagramma di sequenza.



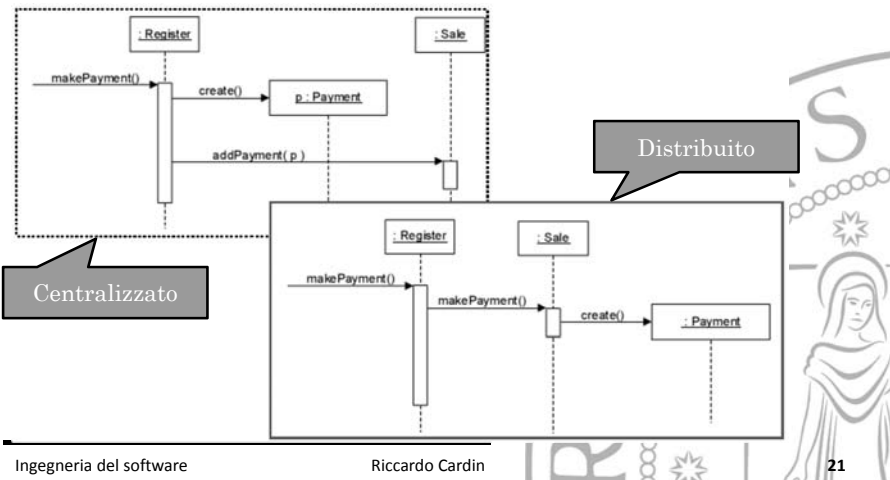
## MODELLAZIONE

- Ottimi per modellare le collaborazioni fra oggetti
  - Non la logica di controllo
- Inadeguatezza a modellare cicli e condizioni ...
  - Meglio i diagrammi di attività
    - ... o pseudocodice ...
- Controllo centralizzato VS Distribuito
  - Centralizzato
    - Unico partecipante che governa l'elaborazione
  - Distribuito
    - Suddivisione dei compiti dei partecipanti



## MODELLAZIONE

### Controllo centralizzato VS Distribuito



Ingegneria del software

Riccardo Cardin

21

## RIFERIMENTI

- o OMG Homepage – [www.omg.org](http://www.omg.org)
- o UML Homepage – [www.uml.org](http://www.uml.org)
- o UML Distilled, Martin Fowler, 2004, Pearson (Addison Wesley)
- o Learning UML 2.0, Kim Hamilton, Russell Miles, O'Reilly, 2006

Ingegneria del software

Riccardo Cardin

22

## GITHUB REPOSITORY



<https://github.com/rcardin/swe>

Ingegneria del software

Riccardo Cardin

23