Analisi

Modello dei dati

- Individuare
  - Oggetti e classi rilevanti per il sistema da sviluppare
    - Limitarsi esclusivamente a quelle classi che fanno parte del vocabolario del dominio del problema
  - Relazioni tra le classi
  - Per ogni classe
    - Responsabilità
    - Attributi
    - Operazioni fondamentali
      cioè servizi forniti all'esterno (interfaccia)
- Raggruppare le classi in sottosistemi (o *package*)

Analisi

Modello dei dati

- Attività non strettamente sequenziali, ma reiterate sino alla produzione di un modello coerente
- Documentazione
  - Diagrammi delle classi e dei sottosistemi
  - Per ogni classe, descrizione che ne specifica scopo, responsabilità, attributi, operazioni
  - Per ogni attributo e ogni operazione, descrizione testuale accurata

Analisi

Individuazione delle classi

- Due analisti non produrranno mai due modelli delle classi identici per lo stesso dominio applicativo
- La letteratura è ricca di approcci raccomandati per l'individuazione delle classi
  - Approccio basato sulle frasi nominali
  - Approccio guidato dai casi d'uso
  - Approccio CRC (*Class-Responsibility-Collaboration*)
- La cosa migliore è usare un approccio misto

Analisi
Individuazione delle classi

- Fonti principali
  - Documento dei requisiti
  - Altri documenti di tutti i tipi che descrivono il sistema
- Altre fonti
  - Altri sistemi che funzionano nello stesso dominio o in domini analoghi
  - Enciclopedie, nomenclature e documenti tecnici che descrivono il dominio
- Riutilizzare classi, gerarchie e strutture ottenute da precedenti analisi nello stesso dominio

Analisi

Individuazione delle classi

- Elencare i nomi (semplici o composti) che compaiono nei documenti raccolti, convertendoli al singolare
- Eliminare i nomi che sicuramente
  - non si riferiscono a classi
  - indicano attributi (dati di tipo primitivo)
  - indicano operazioni
- Scegliere un solo termine significativo se più parole indicano lo stesso concetto (sinonimi)

- Il nome della classe deve essere un nome familiare
  - all'utente o
  - all'esperto del dominio del problema
  - non allo sviluppatore!

Analisi

Individuazione delle classi

- Attenzione agli aggettivi e agli attributi, possono
  - Indicare oggetti diversi
  - Indicare usi diversi dello stesso oggetto
  - Essere irrilevanti

  Ad esempio:
  - “Studente bravo” potrebbe essere irrilevante
  - “Studente fuori corso” potrebbe essere una nuova classe
- Attenzione alle frasi passive, impersonali o con soggetti fuori dal sistema

  devono essere rese attive ed esplicite, perché potrebbero mascherare entità rilevanti per il sistema in esame

Analisi
Individuazione delle classi

- Individuare Attori con cui il sistema in esame deve interagire
  - Persone
    Docente, Studente, Esaminatore,
    Esaminando, …
  - Sistemi esterni
    ReteLocale, Internet, DBMS, …
  - Dispositivi
    attuatori, sensori, …

- Individuare Modelli e loro elementi specifici, cioè oggetti descrittivi e istanze specifiche
  - Insegnamento – “Ingegneria del Software T”
  - CorsoDiStudio – “Ingegneria Informatica”
  - Facoltà – “Ingegneria”

Analisi

Individuazione delle classi

- Individuare Cose tangibili, cioè oggetti reali appartenenti al dominio del problema
  - Banco, LavagnaLuminosa, Schermo, Computer, …
- Individuare Contenitori (fisici o logici) di altri oggetti
  - Facoltà, Dipartimento, Aula, SalaTerminali, …
  - ListaEsame, CommissioneDiLaurea, OrdineDegliStudi, …
- Individuare Eventi o Transazioni che il sistema deve gestire e memorizzare

  - possono avvenire in un certo istante (ad es., una vendita) o
  - possono durare un intervallo di tempo (ad es., un affitto)
  - Appello, EsameScritto, Registrazione, AppelloDiLaurea, …

Analisi

Individuazione delle classi

- Per determinare se includere una classe nel modello, porsi le seguenti domande:
  - il sistema deve interagire in qualche modo con gli oggetti della classe?
    - utilizzare informazioni (attributi) contenute negli oggetti della classe
    - utilizzare servizi (operazioni) offerti dagli oggetti della classe
  - quali sono le responsabilità della classe nel contesto del sistema?

Analisi

Individuazione delle classi

- Attributi e operazioni devono essere applicabili a tutti gli oggetti della classe
- Se esistono
  - attributi con un valore ben definito solo per alcuni oggetti della classe e/o
  - operazioni applicabili solo ad alcuni oggetti della classe

  siamo in presenza di ereditarietà
- Esempio: dopo una prima analisi, la classe Studente potrebbe contenere un attributo *booleano* inCorso, ma un'analisi più attenta potrebbe portare alla luce la gerarchia:

  Studente
  StudenteInCorso
  StudenteFuoriCorso

Analisi
Individuazione delle responsabilità

- Responsabilità di una classe – descrizione ad alto livello dello scopo per cui è stata definita la classe
- Responsabilità di mantenere e gestire un insieme di informazioni (attributi membro e relativi metodi di accesso)
  - è una responsabilità standard che tutte le classi hanno
- Responsabilità di fornire un insieme di servizi pubblici tra loro correlati (operazioni pubbliche)
  - una classe può avere più responsabilità di questo tipo ad esempio, una classe Rettangolo potrebbe avere due responsabilità:
    - fornire un modello matematico della geometria del rettangolo (area, perimetro, ecc.)
    - fornire i servizi per disegnare il rettangolo su una superficie grafica (disegna, colora, ecc.)

Analisi
Individuazione delle responsabilità

- Obiettivo – aiutare nell'identificazione di
  - Classi
  - Attributi
  - Operazioni
  - Relazioni tra le classi

- Principali sorgenti di informazione
  - Documento dei requisiti
  - Classi già identificate
- Annotare tutte le informazioni che gli oggetti devono mantenere e gestire
- Cercare verbi che rappresentano azioni fatte da un oggetto e raggruppare le azioni per tipologia di servizio

Analisi
Individuazione delle responsabilità

- Una volta identificate, le responsabilità vanno assegnate alle classi in base ai seguenti criteri:
  - Distribuire le responsabilità in modo bilanciato (molti oggetti che interagiscono tra loro)
  - Assegnare le responsabilità al livello più generale possibile salendo la gerarchia delle classi
  - Mantenere il comportamento collegato alle informazioni ad esso necessarie
- Se ci sono oggetti che dominano lo scenario, facendo quasi tutto, esiste un problema di distribuzione delle responsabilità

Analisi
Individuazione delle responsabilità

- Metodo di analisi CRC (*Class-Responsibility-Collaboration*) di Cunningham e Beck

Nome della classe	
Lista di responsabilità	**Lista di collaboratori**

- I collaboratori sono le altre classi con le quali la classe in esame deve interagire
- Si utilizzano schede di cartoncino di 10x15 cm

Analisi
Individuazione delle relazioni

- La maggior parte delle classi (degli oggetti) interagisce con altre classi (altri oggetti) in vari modi
- Quando si modella un sistema è necessario individuare:
  - non solo le entità coinvolte nel sistema
  - ma anche le relazione tra tali entità
- Una relazione (*relationship*) è una connessione tra entità

Analisi
Individuazione delle relazioni

- Nella modellazione *object-oriented* le relazioni più importanti sono:
  - Ereditarietà 
  - Associazione
    - Associazione generica 
    - Aggregazione 
    - Composizione 
  - Dipendenza 
    - Collaborazione (relazione usa)
    - Istanza – Classe «instanceOf»
    - Istanza di classe generica – Classe generica «bind»(…)

Analisi
Individuazione delle relazioni

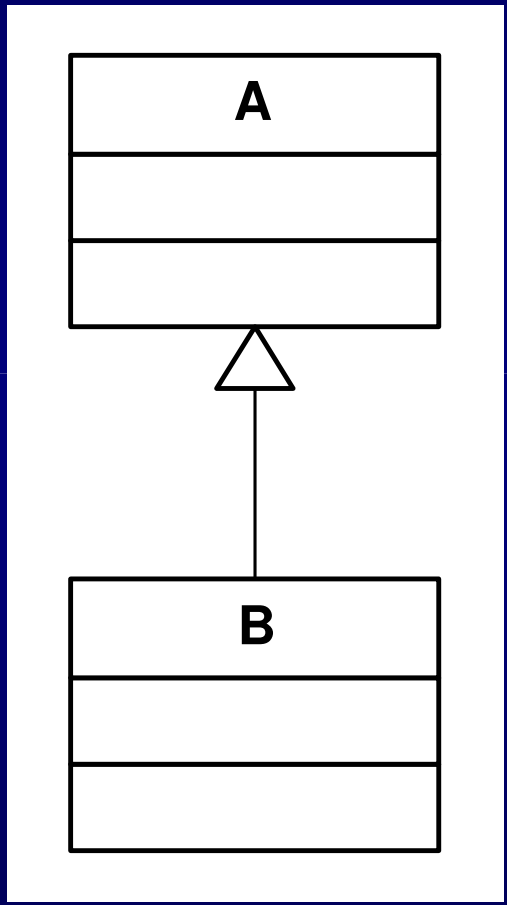
- In ogni tipo di relazione, esiste un cliente C che **dipende** da un fornitore di servizi F
- C ha bisogno di F
  per lo svolgimento di alcune funzionalità
  che C non è in grado di effettuare autonomamente
- Conseguenza
  per il corretto funzionamento di C
  è indispensabile il corretto funzionamento di F

Analisi
Individuazione delle relazioni

Tipo di relazione	Cliente	Fornitore
Ereditarietà	sottoclasse	superclasse
Associazione	contenitore	contenuto
Dipendenza	classe dipendente (che usa)	classe da cui si dipende (che viene usata)
	istanza	classe
	istanza di classe generica	classe generica

Analisi
Individuazione dell'ereditarietà

- La classe B (*specializzazione* o *sottoclasse*) è in relazione IsA con la classe A (*generalizzazione* o *superclasse*) e ne *eredita*
  - relazioni
  - attributi
  - operazioni



Analisi
Individuazione dell'ereditarietà

- L'ereditarietà deve rispecchiare una tassonomia effettivamente presente nel dominio del problema
  - Non usare l'ereditarietà dell'implementazione (siamo ancora in fase di analisi!)
  - Non usare l'ereditarietà solo per riunire caratteristiche comuni ad es., Studente e Dipartimento hanno entrambi un indirizzo, ma non per questo c'è ereditarietà!

Analisi
Individuazione delle associazioni

- Un'associazione rappresenta una relazione strutturale tra due istanze di classi diverse o della stessa classe
- Un'associazione può
  - Rappresentare un contenimento logico (aggregazione)
    - Una lista d'esame contiene degli studenti
  - Rappresentare un contenimento fisico (composizione)
    - Un triangolo contiene tre vertici
  - Non rappresentare un reale contenimento
    - Una fattura si riferisce a un cliente
    - Un evento è legato a un dispositivo

Analisi
Individuazione delle associazioni

- Aggregazione
  Un oggetto x di classe X è associato a (contiene) un oggetto y di classe Y in modo non esclusivo
  x può condividere y con altri oggetti anche di tipo diverso (che a loro volta possono contenere y)
- Una lista d'esame contiene degli studenti
  - Uno studente può essere contemporaneamente in più liste d'esame
  - La cancellazione della lista d'esame non comporta l'eliminazione "*fisica*" degli studenti in lista

Analisi
Individuazione delle associazioni

- Composizione
  Un oggetto x di classe X è associato a (contiene) un oggetto y di classe Y in modo esclusivo
  y esiste solo in quanto contenuto in x

- Un triangolo contiene tre punti (i suoi vertici)
  - L'eliminazione del triangolo comporta l'eliminazione dei tre punti

- Se la distruzione del contenitore comporta la distruzione degli oggetti contenuti, si tratta di composizione, altrimenti si tratta di aggregazione

Analisi
Individuazione delle associazioni

Una volta determinata la presenza di un'associazione

- Tracciare la linea di connessione, eventualmente col simbolo di aggregazione o composizione dalla parte del contenitore
  - l'aggregazione è indicata con un rombo bianco dalla parte del contenitore
  - la composizione è indicata con un rombo nero dalla parte del contenitore
- Se possibile, dare un nome
  - all'associazione (e una direzione di lettura del nome)
  - ai ruoli delle classi coinvolte

scegliendo i nomi tra i termini del dominio del problema

Analisi
Individuazione delle associazioni

- Indicare le due molteplicità, cioè il numero di oggetti che possono partecipare all'associazione

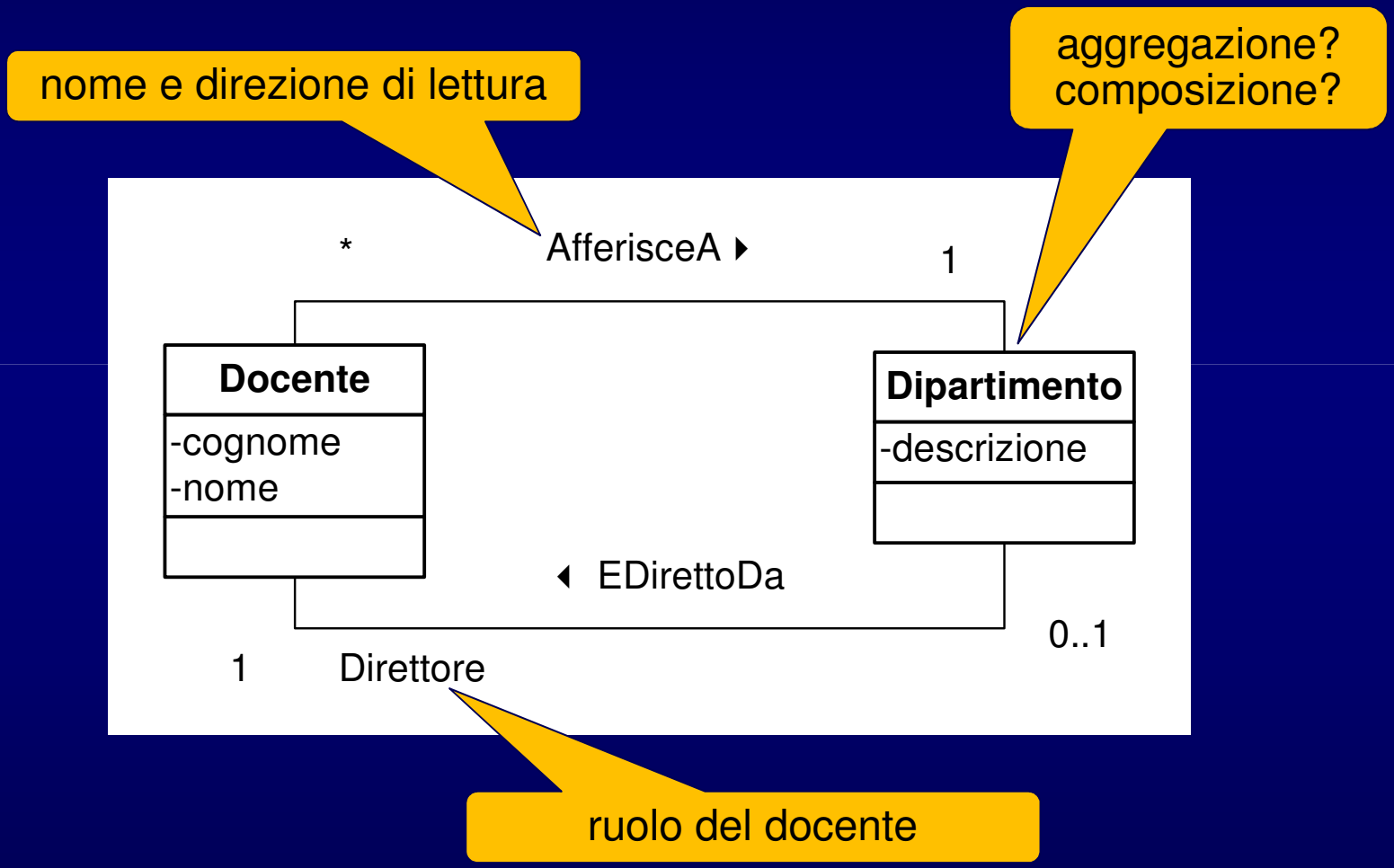
  Per ogni molteplicità, definire il valore, l'intervallo o l'insieme di valori e/o intervalli del
  - limite inferiore
    - la connessione è opzionale? il limite inferiore è 0
    - la connessione è obbligatoria? il limite inferiore è ≥ 1
  - limite superiore
    - la connessione è singola? il limite superiore è 1
    - la connessione è multipla? il limite superiore è > 1

Analisi
Individuazione delle associazioni

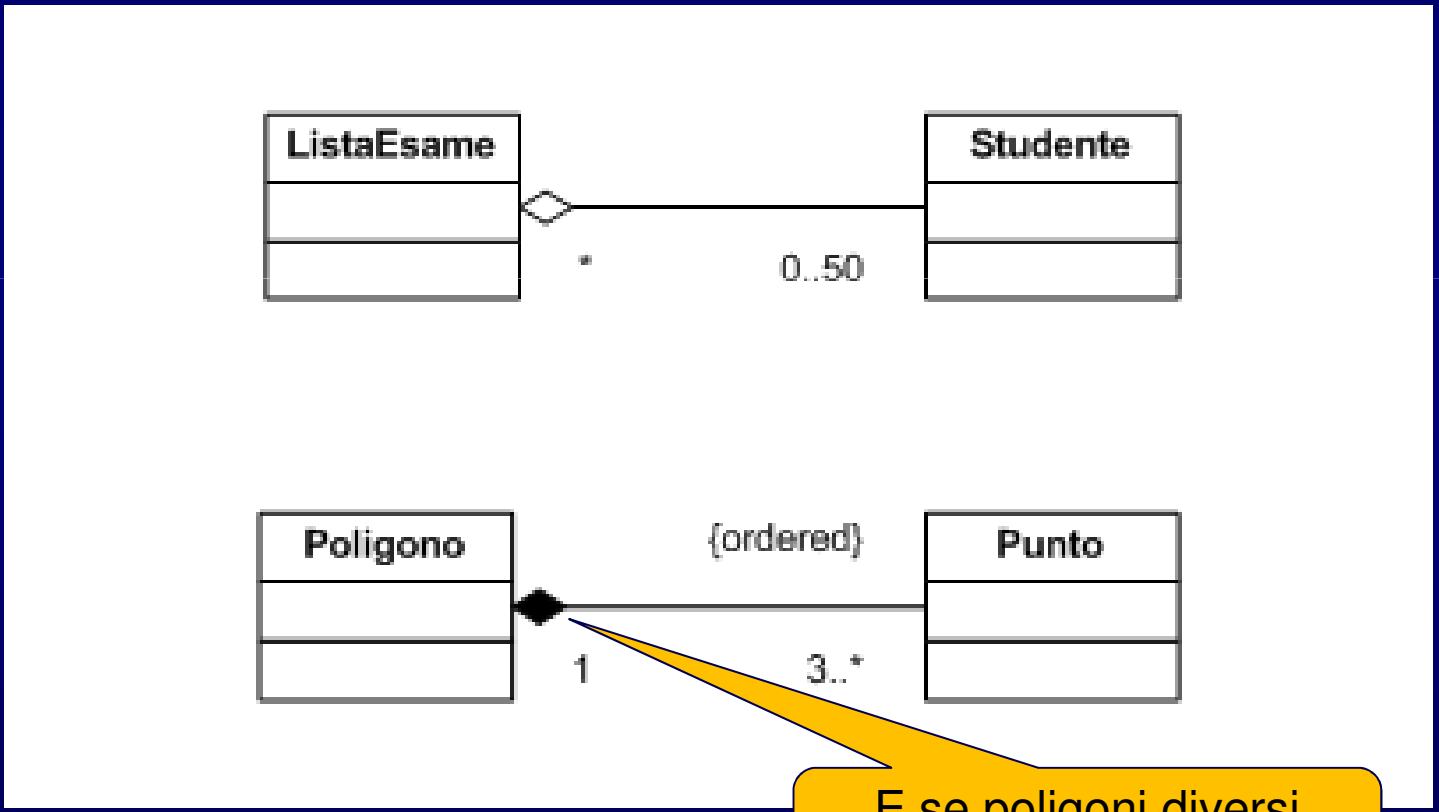
- Molteplicità

Simbolo	Significato	Esempi
n	valore singolo	1
*	da 0 a ∞	*
n..m	intervallo	0..1 1..*
n,m	insieme di valori e/o intervalli	1,3..5,7

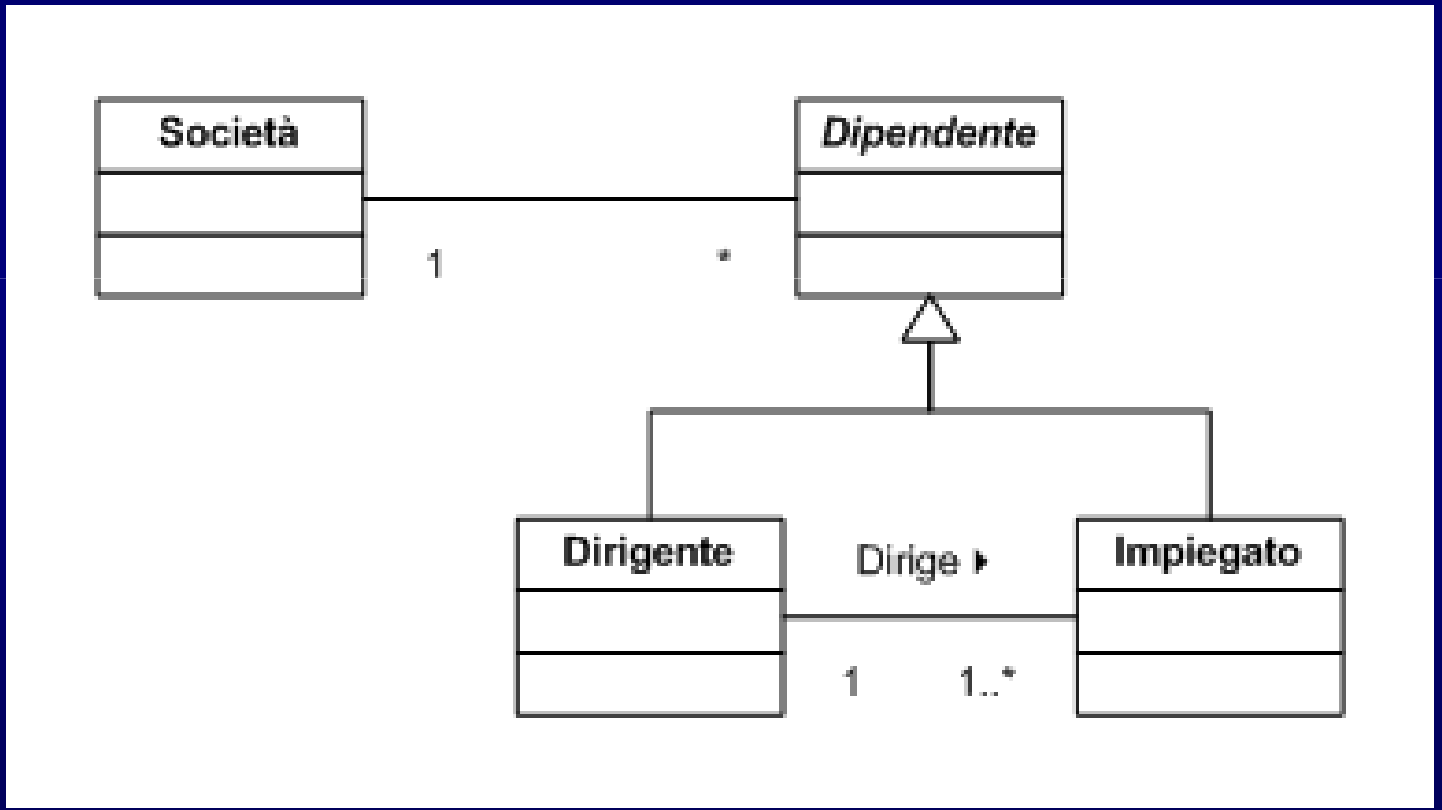
Analisi
Esempi di associazioni



Analisi
Esempi di associazioni



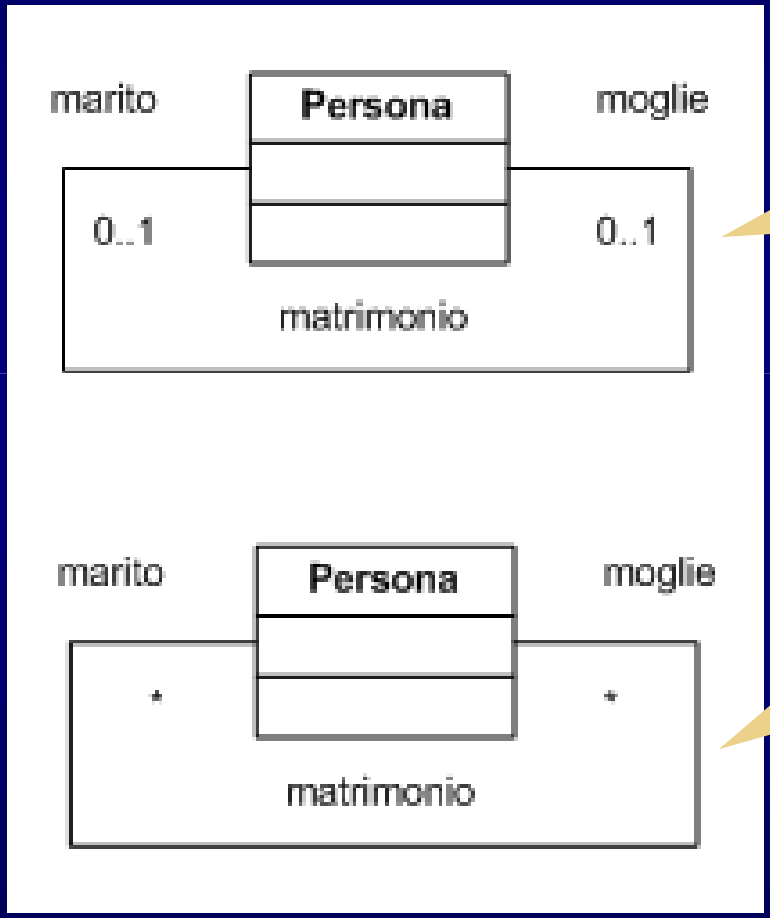
Analisi
Esempi di associazioni



Analisi
Individuazione delle associazioni

- Attenzione alle associazioni molti a molti
  possono nascondere una classe (**classe di associazione**) del tipo “evento da ricordare”
- Ad esempio,
  - la connessione “matrimonio” tra Persona e Persona può nascondere una classe Matrimonio, che lega due Persone
  - la connessione “possiede” tra Proprietario e Veicolo può nascondere una classe CompraVendita, che lega un Proprietario a un Veicolo
  - la connessione “èDirettoDa” tra Dipartimento e Docente può nascondere una classe Direzione che lega un Dipartimento e un Docente che svolge il ruolo di Direttore del dipartimento

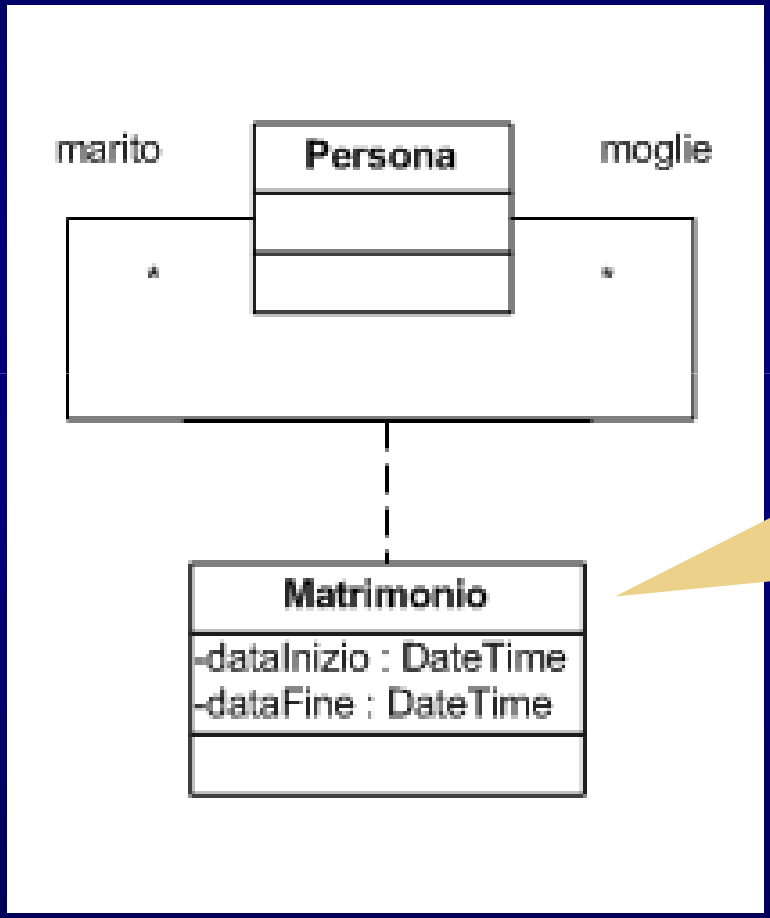
Analisi
1° esempio di associazione



Modella una situazione **corrente** in cui una Persona è associata al massimo ad un'altra Persona mediante matrimonio

Vorrebbe modellare una situazione **storica** in cui si tiene traccia di tutte le associazioni mediante matrimonio di una Persona a 0+ altre Persone

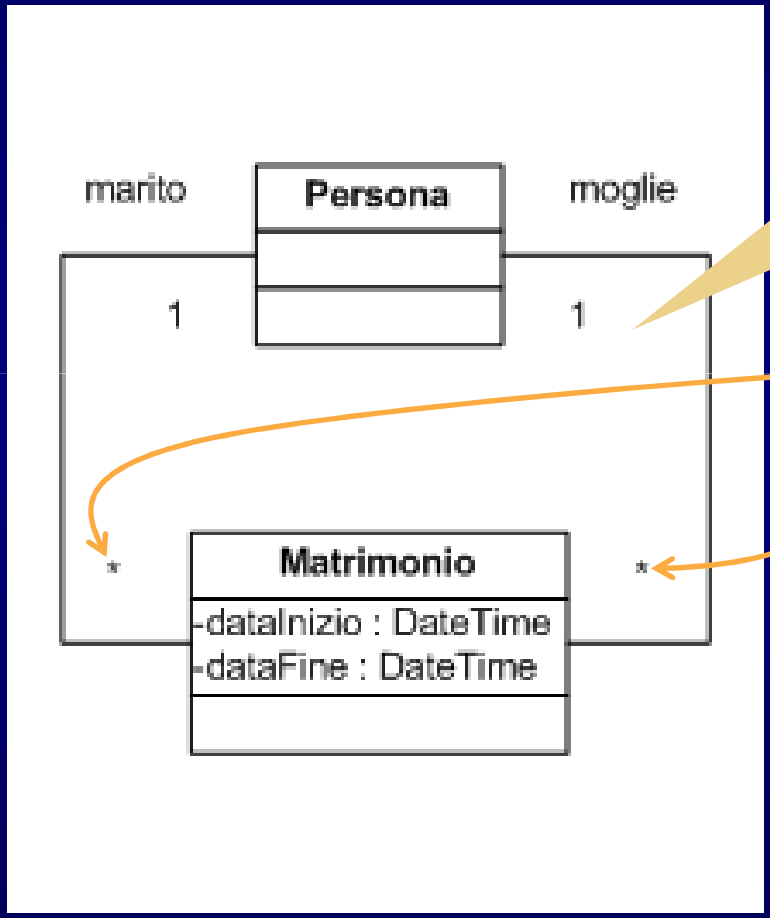
Analisi
1° esempio di associazione



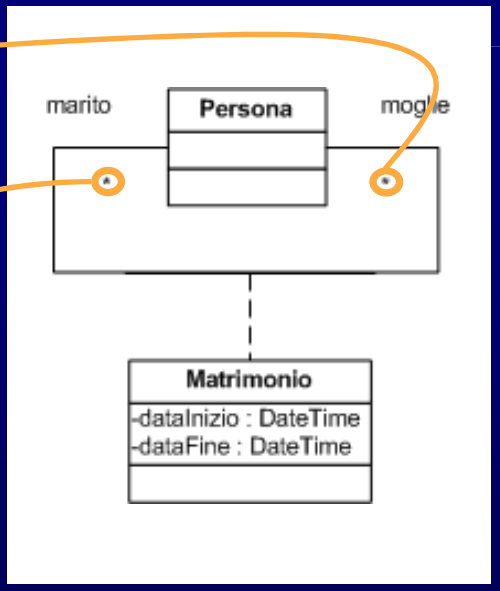
Modella una situazione **storica** in cui si tiene traccia di tutte le associazioni mediante matrimonio di una Persona a 0+ altre Persone

NB: vincoli sulle date

Analisi
1° esempio di associazione

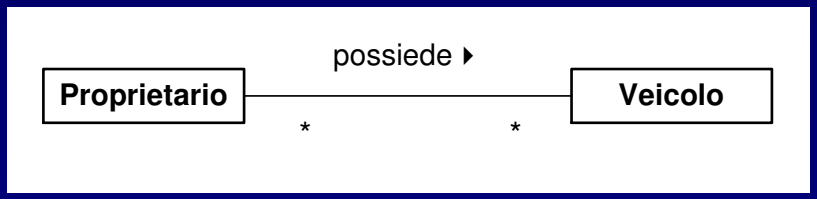


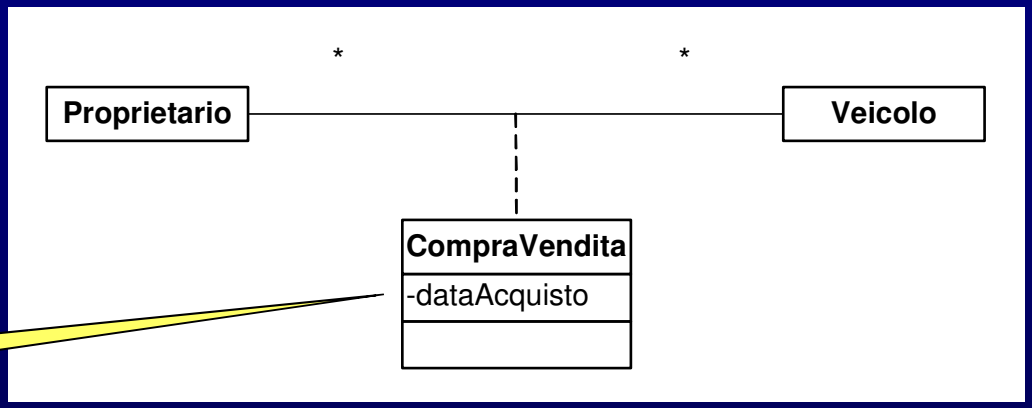
Modello equivalente al precedente in cui la classe associazione è stata concretizzata in una classe normale
(da usare solo in **progettazione**)



Analisi
2° esempio di associazione

- Un proprietario può possedere molti veicoli
- Un veicolo può essere di molti proprietari
  - in tempi successivi
  - in comproprietà

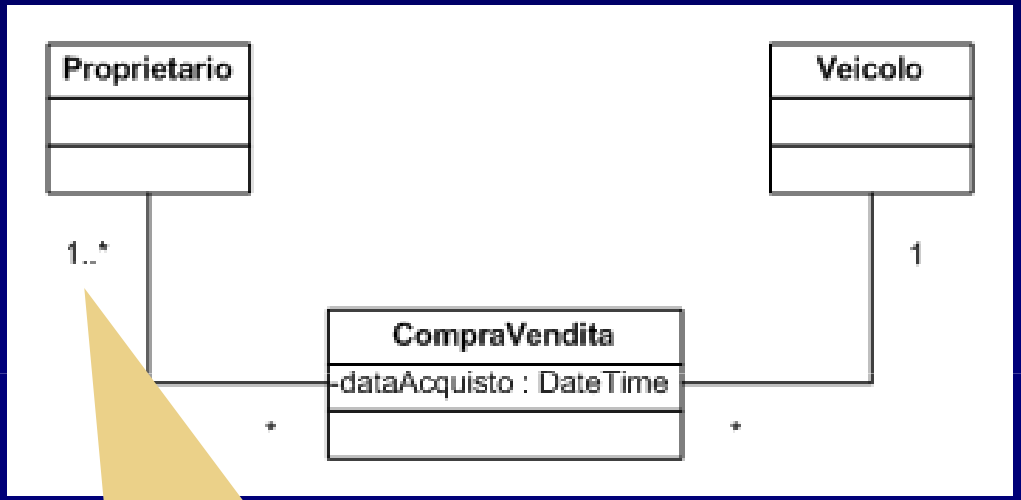




Legame 1 a 1

Analisi
2° esempio di associazione

- Un proprietario può partecipare a più compravendite
- Una compravendita è relativa a un solo veicolo
- Un veicolo può essere contem-poraneamente di più proprietari



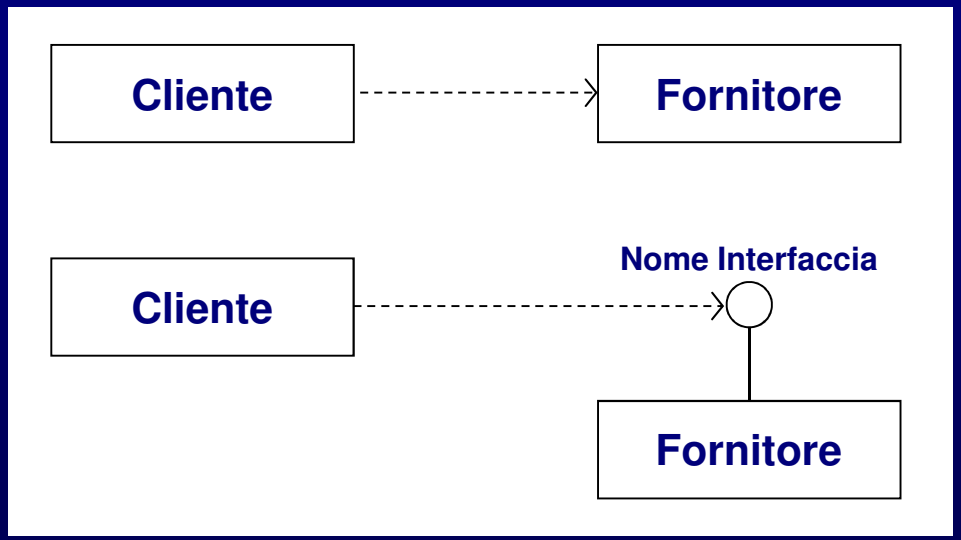
A causa della molteplicità 1..* NON può essere utilizzata una classe associazione (legame sempre 1 a 1)

Analisi
Individuazione delle collaborazioni

- Una classe A è in relazione USA con una classe B (A USA B) quando A ha bisogno della collaborazione di B per lo svolgimento di alcune funzionalità che A non è in grado di effettuare autonomamente
  - Un'operazione della classe A ha bisogno come argomento di un'istanza della classe B
    - void fun1(B b) { … usa b … }
  - Un'operazione della classe A restituisce un valore di tipo B
    - B fun2(…) { B b; … return b; }
  - Un'operazione della classe A utilizza un'istanza della classe B (ma non esiste un'associazione tra le due classi)
    - void fun3(…) { B b = new B(…); … usa b … }

Analisi
Individuazione delle collaborazioni

- La relazione non è simmetrica
  A dipende da B, ma B non dipende da A
- Evitare situazioni in cui una classe, tramite una catena di relazioni USA, alla fine dipende da se stessa



Analisi
Individuazione degli attributi

- Ogni attributo modella una proprietà atomica di una classe
  - Un valore singolo
    - Una descrizione
    - Un importo
    - …
  - Un gruppo di valori strettamente collegati tra loro
    - Un indirizzo
    - Una data
    - …
- Proprietà non atomiche di una classe devono essere modellate come associazioni
- Run-time, in un certo istante, ogni oggetto avrà un valore specifico per ogni attributo della classe di appartenenza: informazione di stato

Analisi
Individuazione degli attributi

- Il nome dell'attributo
  - Deve essere un nome familiare
    - all'utente o
    - all'esperto del dominio del problema
    - non allo sviluppatore!
  - Non deve essere il nome di un valore ("qualifica" sì, "ricercatore" no)
  - Deve iniziare con una lettera minuscola
- Esempi
  - cognome
  - dataDiNascita
  - annoDiImmatricolazione

Analisi
Individuazione degli attributi

- Esprimere tutti i vincoli applicabili all'attributo
  - Tipo (semplice, struttura, enumerativo)
    - cognome: string
    - sesso: {maschio, femmina}
  - Valori ammessi
  - Valore di *default*
    - sesso: {maschio, femmina} = maschio
  - Vincoli di creazione o di accesso
  - Vincoli dovuti ai valori di altri attributi

Analisi
Individuazione degli attributi

- Esprimere tutti i vincoli applicabili all'attributo
  - Opzionalità
    - votoFinale [0..1]: Votazione
  - Unità di misura, precisione
  - Visibilità (opzionale in fase di analisi)
    - Pubblica +
    - Protetta #
    - Privata –

  Attenzione:
  gli attributi devono essere sempre privati!

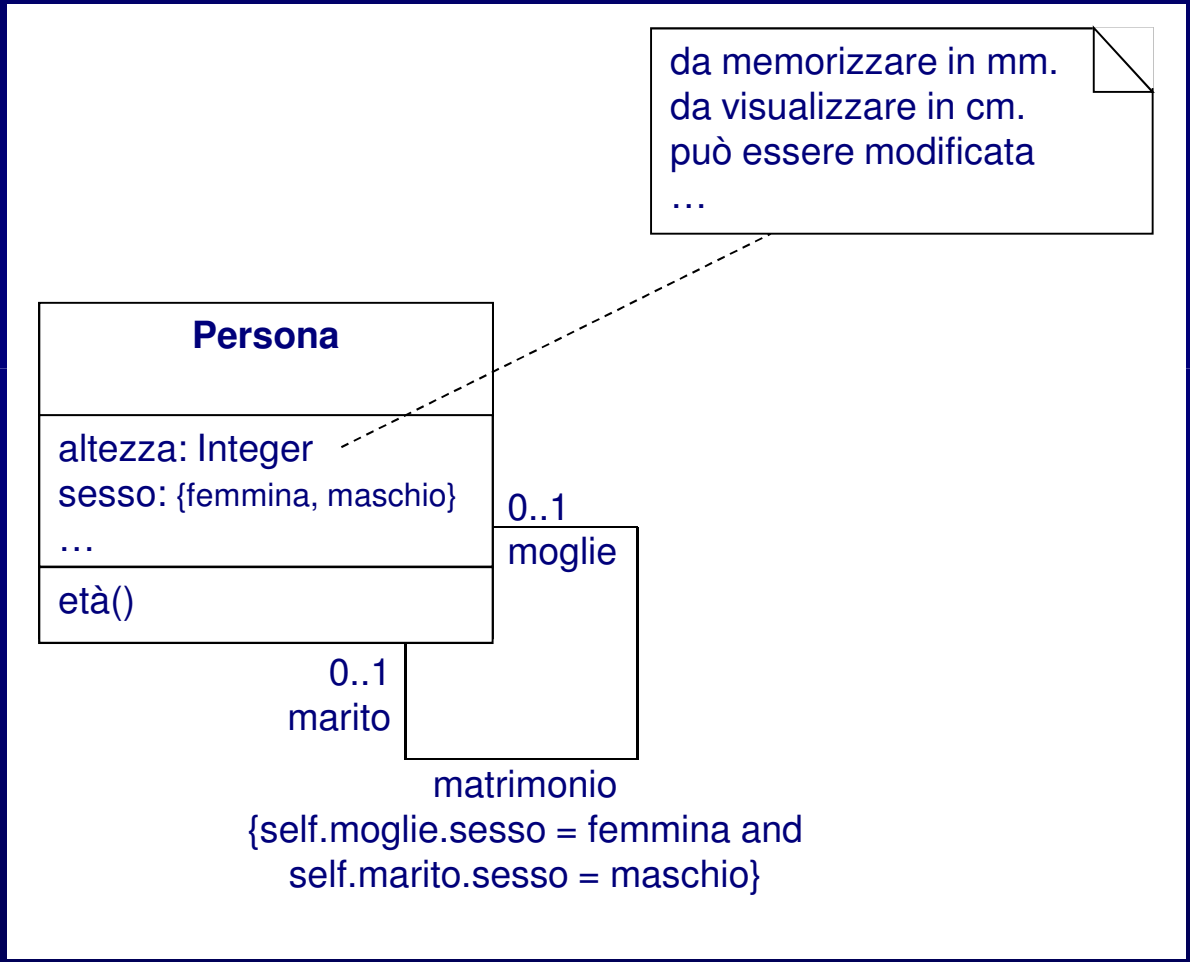
Analisi
Individuazione degli attributi

- Esprimere tutti i vincoli applicabili all'attributo
  - Appartenenza alla classe (e non all'istanza)
    Attributi (e associazioni) possono essere di classe, cioè essere unici nella classe e quindi non fare parte della struttura dati delle istanze
    - -numIstanze: int = 0

Analisi
Individuazione degli attributi

- I vincoli possono essere scritti
  - Utilizzando direttamente UML
    - Tipo
    - Opzionalità
    - Visibilità
    - …
  - Utilizzando *Object Constraint Language* (OCL) descritto in “The Unified Modeling Language Reference Manual”
  - Come testo in formato libero in un commento UML

Analisi
Individuazione degli attributi



Analisi
Individuazione degli attributi

- Attenzione
  nel caso di attributi con valore booleano
  “vero o falso”, “sì o no”,
  il nome dell’attributo potrebbe essere
  uno dei valori di un’enumerazione

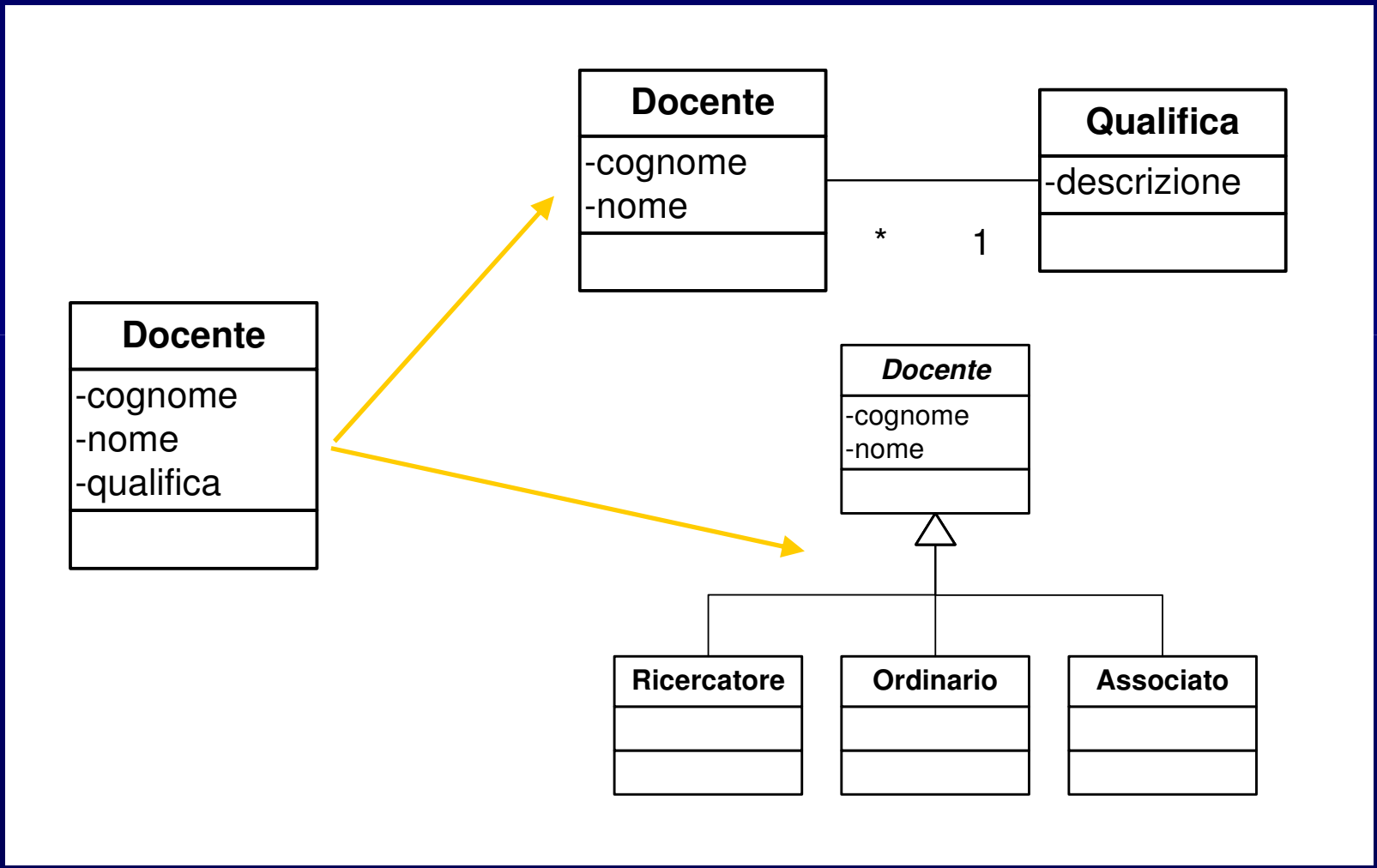
  Ad esempio:
  - tassabile (sì o no) potrebbe diventare
    tipoTassazione { tassabile, esente, ridotto, … }

Analisi
Individuazione degli attributi

- Attenzione
  attributi
  - con valore “non applicabile” o
  - con valore opzionale o
  - a valori multipli (enumerazioni)

  possono nascondere
  - ereditarietà o
  - una nuova classe

Analisi
Individuazione degli attributi



Analisi
Individuazione degli attributi

- Attenzione
  nel caso di attributi calcolabili (ad esempio, età), specificare
  - sempre l'operazione di calcolo
  - mai l'attributo
- Se memorizzare oppure no un attributo calcolabile è una decisione progettuale, un compromesso tra
  - tempo di calcolo
  - spazio di memoria

Analisi
Individuazione degli attributi

- Attenzione
  non definire attributi che indicano il tipo (o categoria) di un oggetto
- Deve essere sempre possibile ottenere il tipo di un oggetto mediante un'operazione, ad esempio
  - `nomeClasse()`
  - `GetType()` – in .NET
    - `GetType().ToString()`
      per avere il nome della classe

Analisi
Individuazione degli attributi

- Applicare l'ereditarietà:
  - Posizionare attributi e associazioni più generali più in alto possibile nella gerarchia
  - Posizionare attributi e associazioni specializzati più in basso

Analisi
Individuazione delle operazioni

- Per completare la fase di analisi,
  è necessario descrivere in modo dettagliato gli aspetti più volatili del sistema
  - Le operazioni (o servizi) che ogni classe deve offrire pubblicamente
  - La sequenza temporale di tali operazioni

Analisi
Individuazione delle operazioni

- Il nome dell'operazione
  - Deve appartenere al vocabolario standard del dominio del problema
  - Potrebbe essere un verbo
    - all'imperativo (scrivi, esegui, ...) o
    - in terza persona (scrive, esegue, ...)

    in modo consistente in tutto il sistema
  - Dovrebbe iniziare con una lettera maiuscola (convenzione .NET)

Analisi
Individuazione delle operazioni

- Notazione UML

  visibilità nomeOperazione(lista-parametri): tipoRestituito

  +GetCognome(): string

  +SetCognome(nuovoValore: string)

  +GetNumIstanze(): int

  +*Visualizza*(disp: OutputDevice)

Analisi
Individuazione delle operazioni

- Operazioni standard
  - Operazioni che tutti gli oggetti hanno per il semplice fatto di esistere e di avere degli attributi e delle relazioni con altri oggetti
  - Sono implicite e, di norma, non compaiono nel diagramma delle classi di analisi
- Altre operazioni
  - Devono essere determinate
    - servizi offerti agli altri oggetti
  - Compaiono nel diagramma delle classi di analisi

Analisi
Individuazione delle operazioni

Operazioni standard	
Costruttore	Inizializza un nuovo oggetto (`.ctor`) Invocato automaticamente dalla **`new`**
Distruttore	Effettua tutte le azioni necessarie per rilasciare le risorse possedute dall'oggetto Invocato automaticamente da GC o **`delete`**
Accessor	**`Get`** - restituisce il valore di un attributo atomico o il riferimento a un oggetto (nel caso di associazioni)
	`Set` - modifica il valore di un attributo atomico oppure collega/scollega un oggetto a un altro oggetto (nel caso di associazioni)

Analisi
Individuazione delle operazioni

Classi contenitori (di analisi)

- Operazioni standard
  - Aggiungi, rimuovi, conta, itera, …
- Altre operazioni – riguardano l'insieme degli oggetti, non il singolo oggetto
  - Calcoli da effettuare sugli oggetti contenuti
    - CalcolaSulleParti(), Totalizza()
  - Selezioni da fare sugli oggetti contenuti
    - TrovaPartiSpecifiche()
  - Operazioni del tipo
    - EseguiUnAzioneSuTutteLeParti()

Analisi
Individuazione delle operazioni

- Distribuire in modo bilanciato le operazioni nel sistema
- Mettere ogni operazione "*vicino*" ai dati ad essa necessari
- Applicare l'ereditarietà
  - Posizionare le operazioni più generali più in alto possibile nella gerarchia
  - Posizionare le operazioni specializzate più in basso

Analisi
Individuazione delle operazioni

- Descrivere tutti i vincoli applicabili all'operazione
  - Parametri formali, loro tipo e significato
  - Pre-condizioni e post-condizioni
  - Invarianti di classe
  - Eccezioni sollevate
  - Eventi che attivano l'operazione
  - Applicabilità dell'operazione
  - …

Analisi
Individuazione delle operazioni

- PRE-CONDIZIONE
  Espressione logica riguardante le
  aspettative sullo stato del sistema
  prima che venga eseguita un'operazione
- Ad esempio, per l'operazione
  `CalcolaRadiceQuadrata(valore)`,
  la pre-condizione potrebbe essere: "**`valore ≥ 0`**"
- Esplicita in modo chiaro che
  è responsabilità della procedura chiamante
  controllare la correttezza degli argomenti passati

Analisi
Individuazione delle operazioni

- POST-CONDIZIONE
  Espressione logica riguardante le aspettative sullo stato del sistema dopo l'esecuzione di un'operazione
- Ad esempio, per l'operazione `CalcolaRadiceQuadrata(valore)`, la post-condizione potrebbe essere: "`valore == risultato * risultato`"

Analisi
Individuazione delle operazioni

- INVARIANTE di classe
  Vincolo di classe (espressione logica)
  che deve essere sempre verificato
  - sia all'inizio
  - sia alla fine

  di tutte le operazioni pubbliche della classe
- Può non essere verificato solo durante l'esecuzione dell'operazione

Analisi
Individuazione delle operazioni

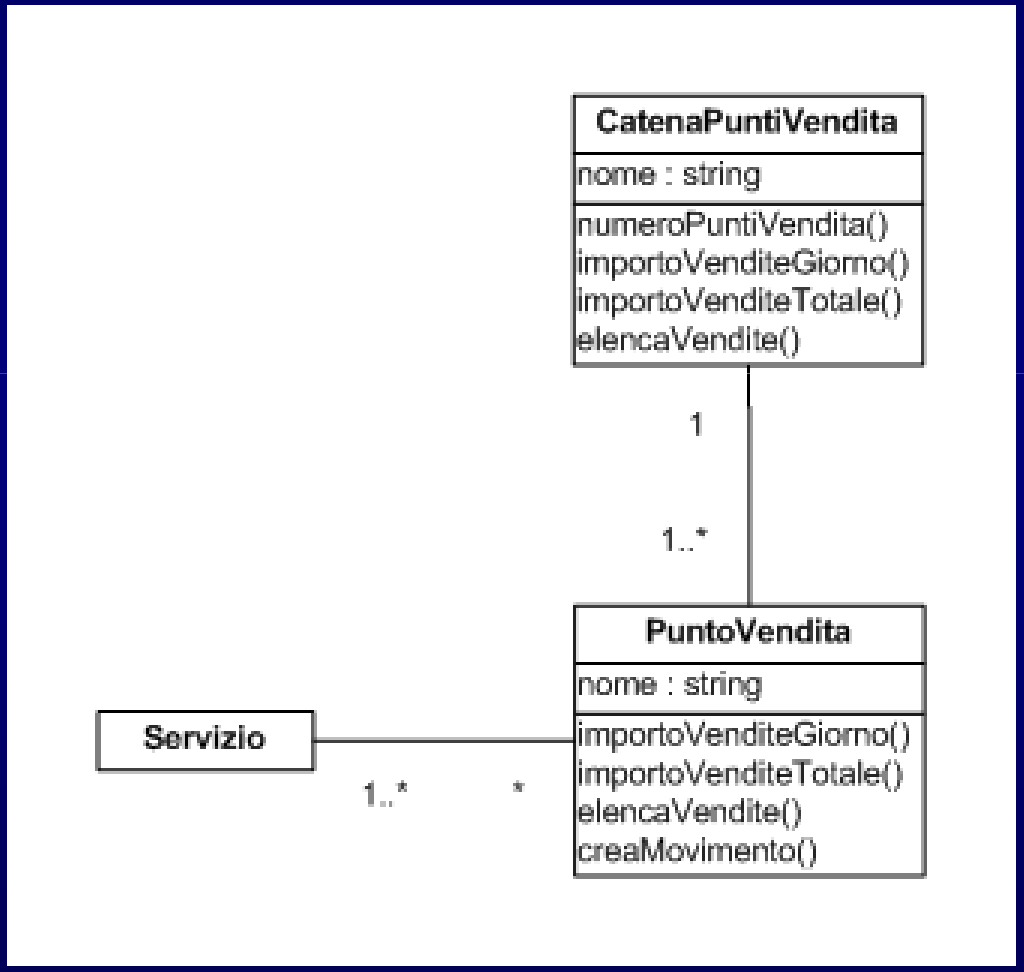
- In caso di ridefinizione di un'operazione in una sotto-classe
  - Le pre-condizioni devono essere identiche o meno stringenti
  - Le post-condizioni devono essere identiche o più stringenti
  - Gli invarianti di classe devono essere identici o più stringenti

Analisi
Individuazione delle operazioni

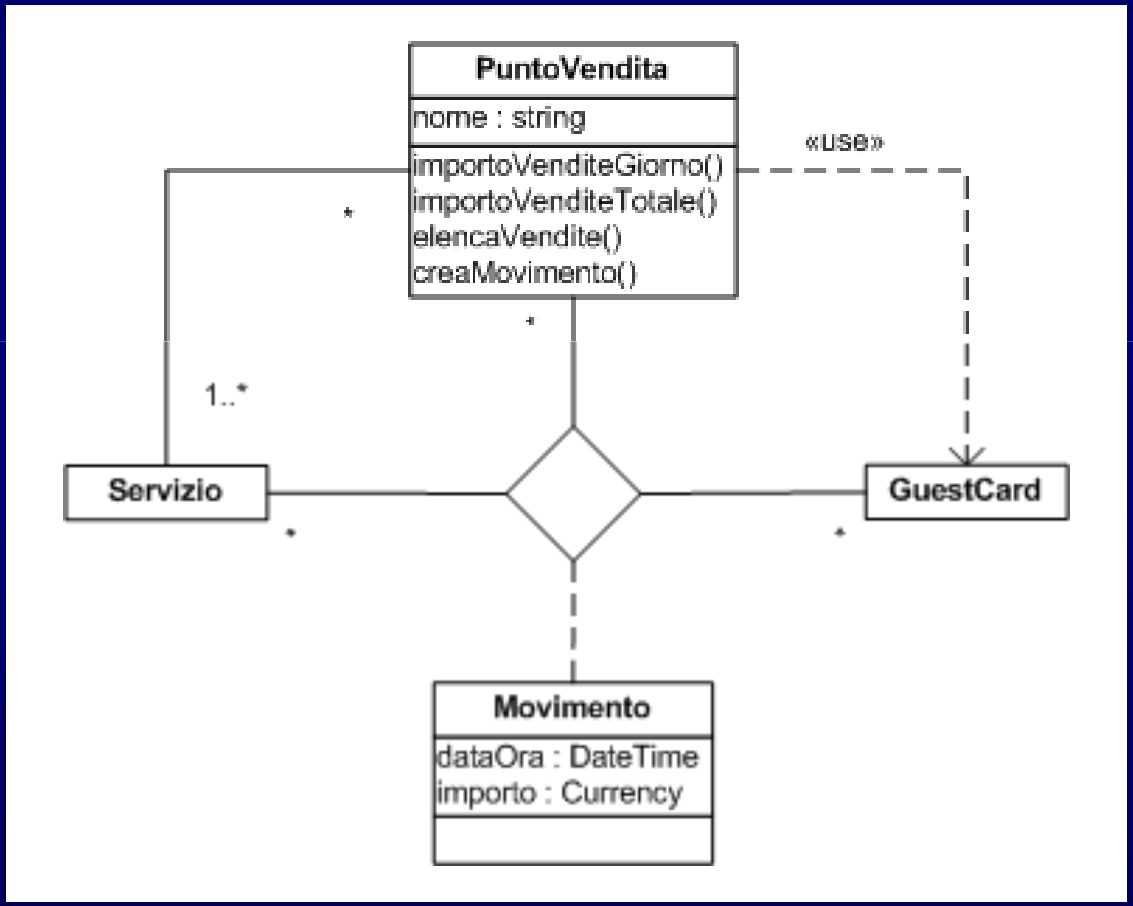
- ECCEZIONE

  Si verifica quando un'operazione
  - viene invocata nel rispetto delle sue pre-condizioni
  - ma non è in grado di terminare la propria esecuzione nel rispetto delle post-condizioni

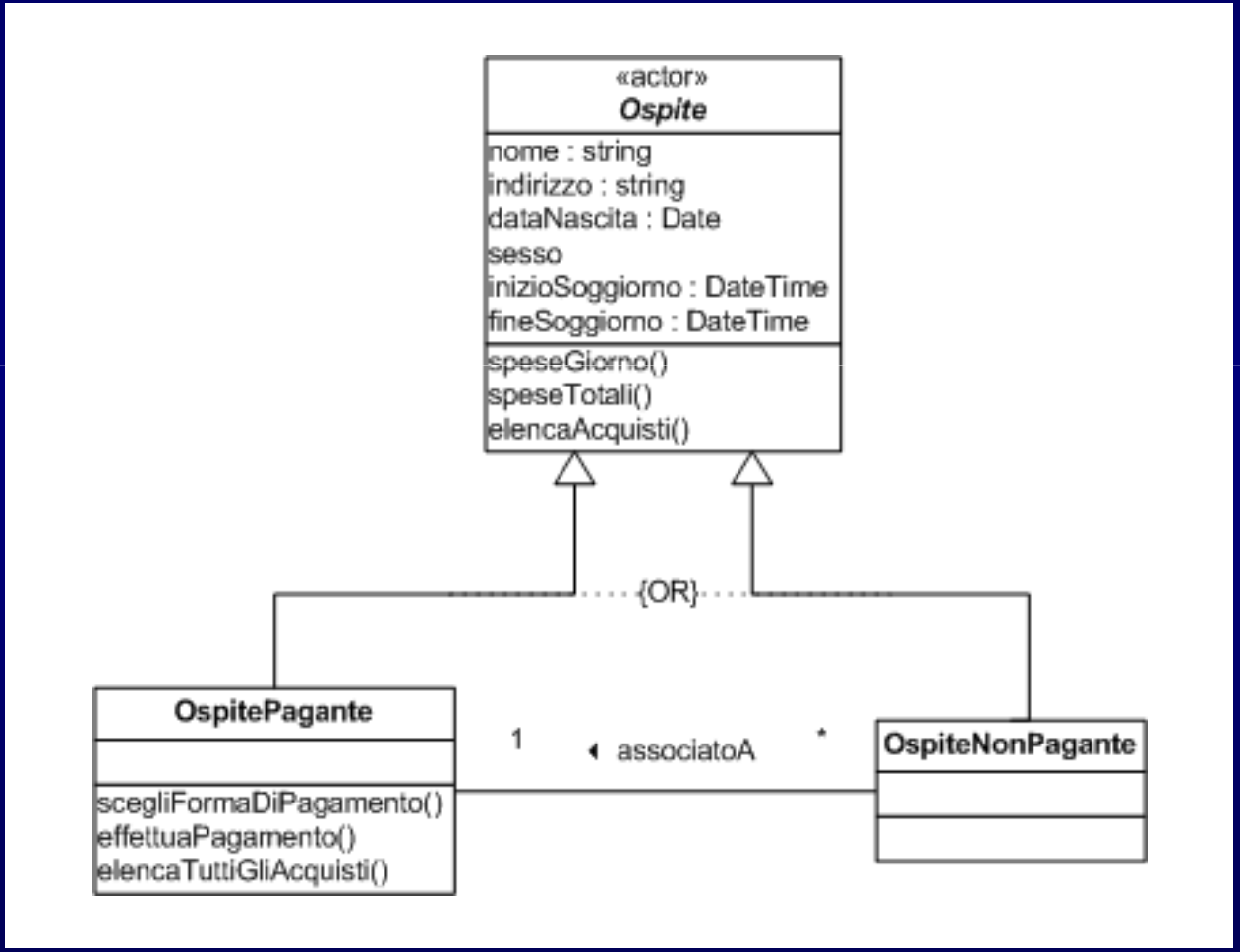
Esempio: Villaggio Turistico
Modello dei dati



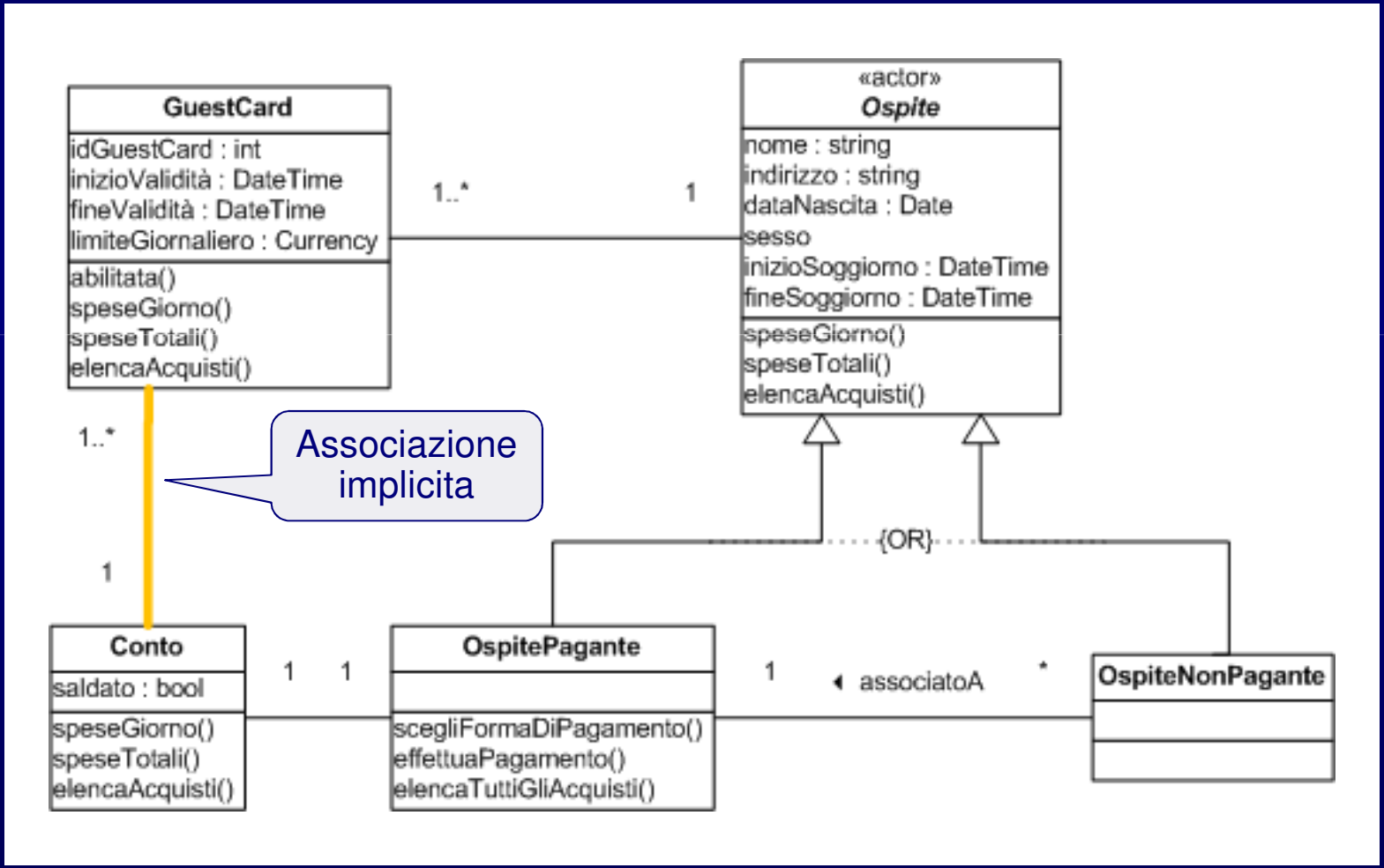
Esempio: Villaggio Turistico
Modello dei dati



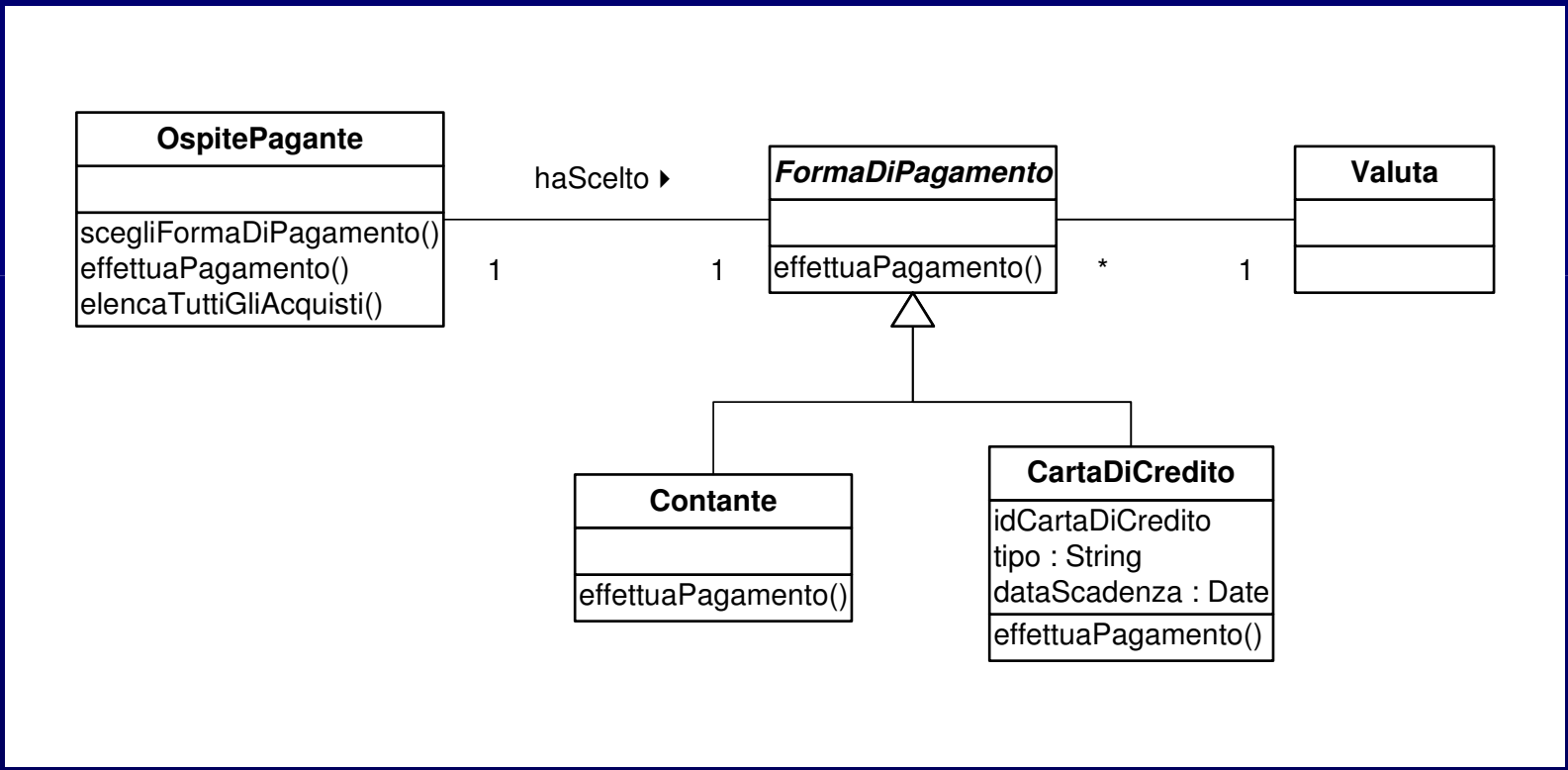
Esempio: Villaggio Turistico
Modello dei dati



Esempio: Villaggio Turistico
Modello dei dati



Esempio: Villaggio Turistico
Modello dei dati



Analisi
Modello dinamico

- Descrive il comportamento del sistema durante il suo funzionamento
- Diagrammi dei casi d'uso - evidenziano
  - le risposte del sistema a sollecitazioni esterne (richieste degli utenti o eventi)
- Diagrammi di sequenza - evidenziano
  - lo scambio di messaggi (le interazioni) tra gli oggetti
  - l'ordine in cui i messaggi vengono scambiati tra gli oggetti (sequenza di invocazioni delle operazioni)
- Diagrammi di stato - evidenziano
  - i possibili stati che un oggetto può assumere
  - gli eventi interni o esterni che attivano transizioni da uno stato all'altro

Analisi
Modello dinamico

- Non spingere la definizione del modello dinamico sino ai minimi dettagli
- Utilizzare i diagrammi dinamici solo per descrivere il funzionamento del sistema
  - in risposta a sollecitazioni esterne
  - in fasi particolarmente significative
  - nei casi più critici

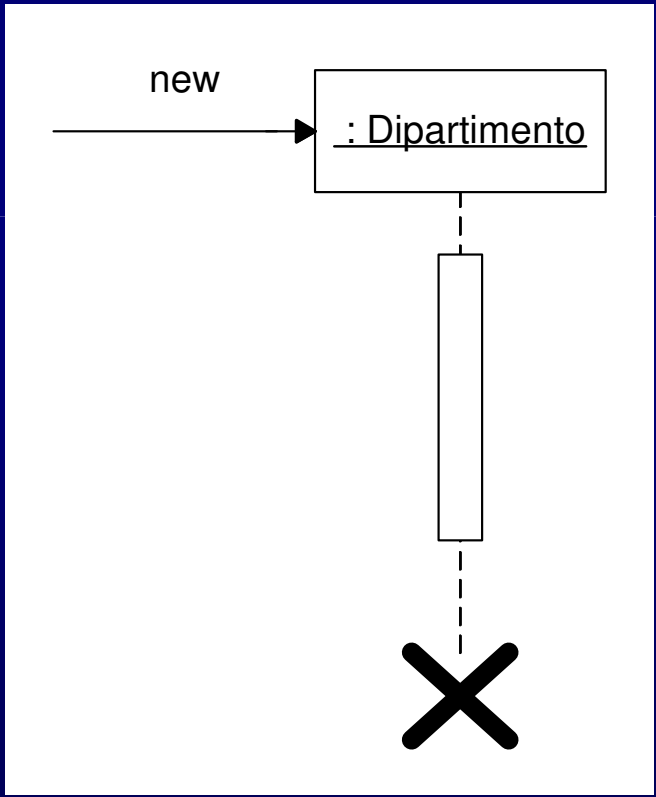
Analisi
Diagrammi di sequenza

- Obiettivo: porre in evidenza
  - lo scambio di messaggi (le interazioni) tra gli oggetti
  - l'ordine in cui i messaggi vengono scambiati tra gli oggetti (sequenza di invocazioni delle operazioni)
- L'invio di un messaggio (o richiesta di un servizio)
  - da un oggetto mittente (cliente)
  - a un oggetto destinatario (fornitore)

  corrisponde all'invocazione di un'operazione della classe del destinatario o di una sua superclasse
- Ogni messaggio comprende:
  - il nome dell'operazione invocata
  - gli eventuali parametri attuali
  - un eventuale valore restituito

Analisi
Diagrammi di sequenza

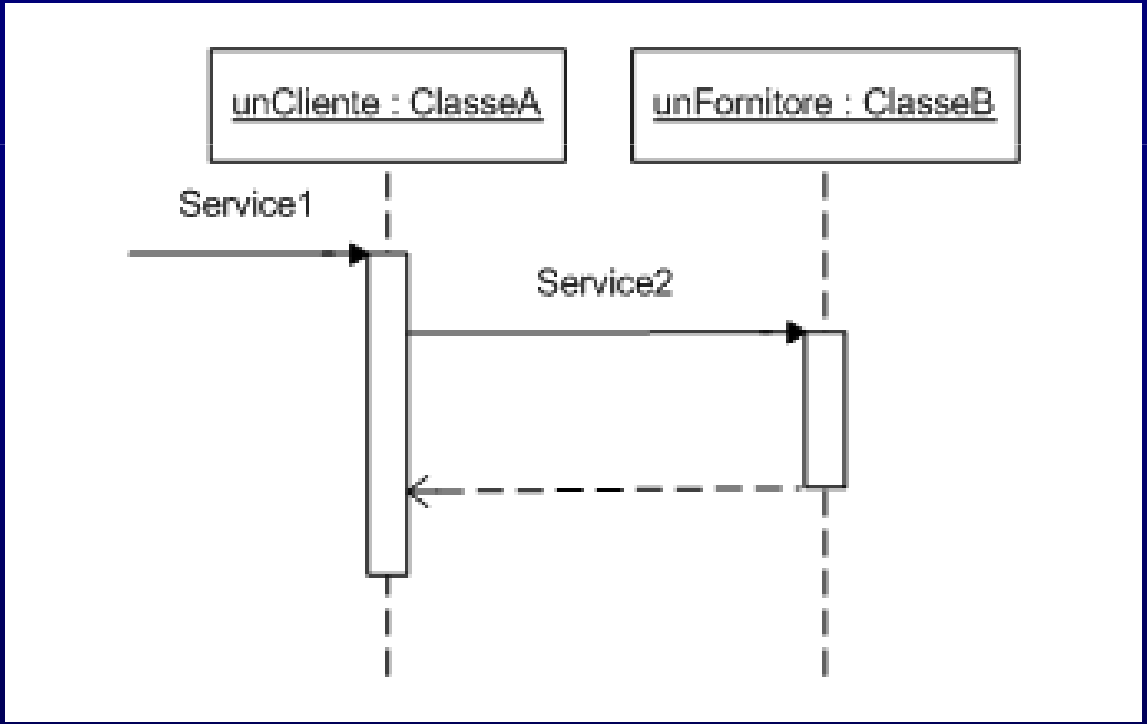
- Gli oggetti sono indicati da rettangoli con dentro i nomi sottolineati dell'oggetto e/o della relativa classe



- Le linee verticali tratteggiate (*lifeline*) indicano il tempo di vita degli oggetti
- Il flusso temporale scorre dall'alto in basso
- Gli ispessimenti delle linee verticali denotano un'elaborazione in corso
- È possibile indicare esplicitamente quando un oggetto viene creato e quando viene distrutto

Analisi
Diagrammi di sequenza

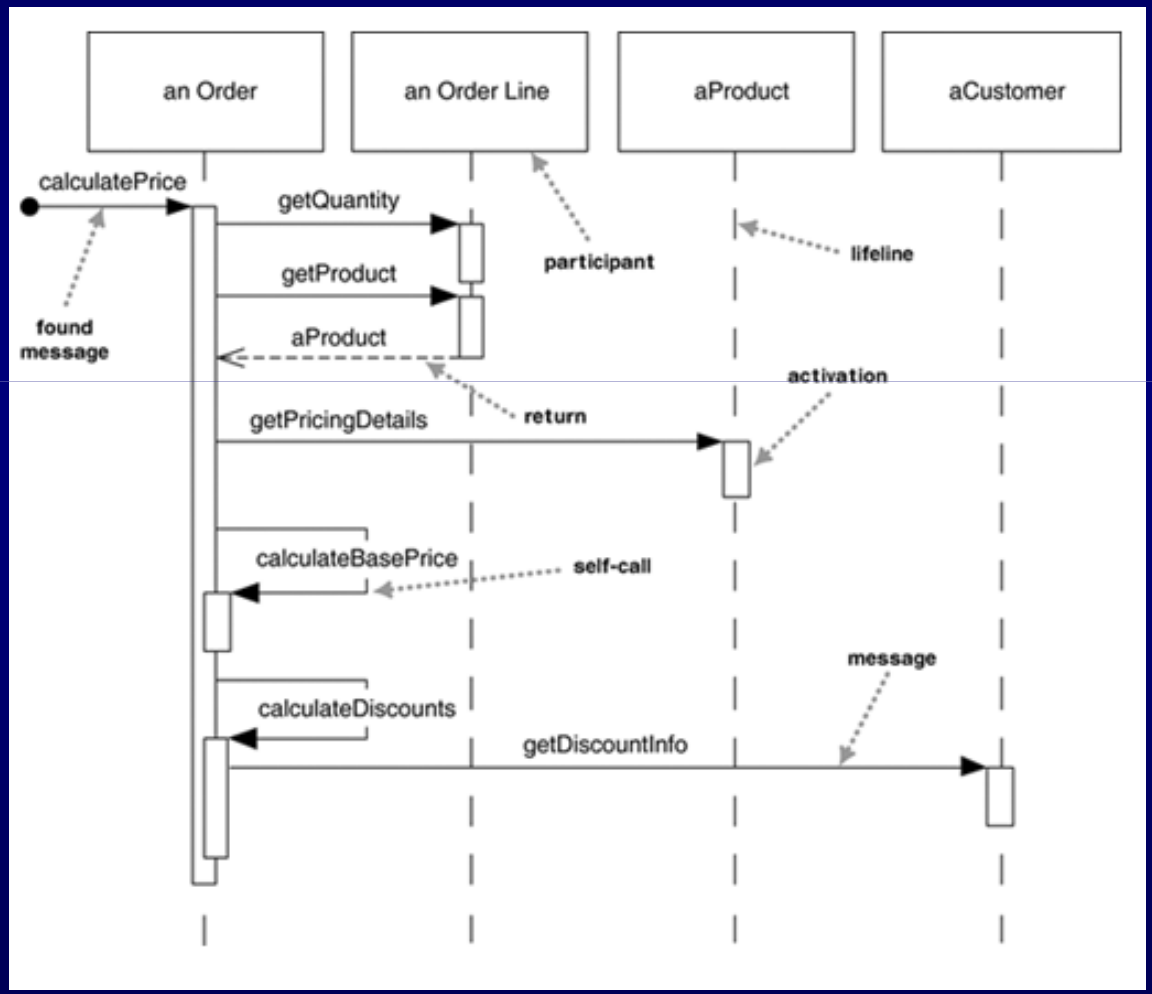
- Tra gli oggetti vi possono essere invii di messaggi, denotati da frecce, e ritorni (opzionali)



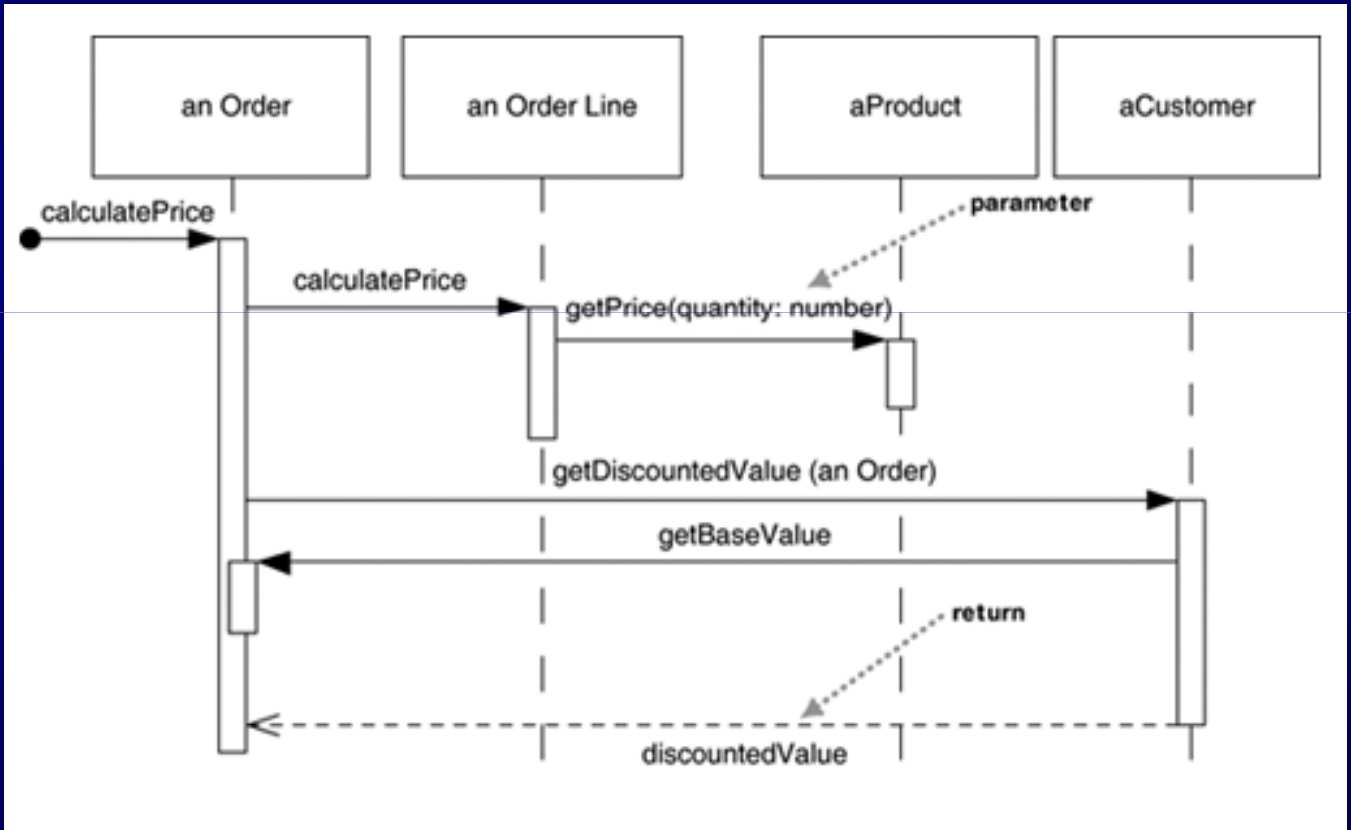
Analisi
Diagrammi di sequenza

- Un oggetto (cliente) può inviare un messaggio (chiedere un servizio) ad un altro oggetto (fornitore)
  - solo durante l'esecuzione di un proprio metodo
  - solo se conosce il fornitore
- Il fornitore è un oggetto globalmente accessibile (una classe o un oggetto globale)
- Il fornitore è contenuto nel cliente (per valore o per riferimento)
- Il fornitore è stato passato come parametro attuale al metodo del cliente
- Il fornitore è il risultato dell'invio di un precedente messaggio dentro il metodo del cliente – due possibilità:
  - Il fornitore esisteva già (e viene recuperato)
  - Il fornitore viene creato, utilizzato (e distrutto)

Analisi
Diagrammi di sequenza



Analisi
Diagrammi di sequenza



Analisi
Diagrammi di stato

- Lo stato di un oggetto è dato dal valore dei suoi attributi e delle sue associazioni
- In molti domini applicativi, esistono oggetti che, a seconda del proprio stato, rispondono in maniera diversa ai messaggi ricevuti
  - Dispositivi (spento, in attesa, operativo, guasto, ecc.)
  - Transazioni complesse (in definizione, in esecuzione, completata, fallita, ecc.)
  - …
- In questi casi, è opportuno disegnare un diagramma di stato per l'oggetto, mostrando i possibili stati e gli eventi che attivano transizioni da uno stato all'altro

Analisi
Diagrammi di stato

- STATO – Situazione, che si verifica durante la vita di un oggetto, nella quale l'oggetto
  - soddisfa certe condizioni
  - esegue certe attività
  - aspetta il verificarsi di certi eventi
- EVENTO – Può essere:
  - esterno – generato nell'ambito del mondo reale (mouse, tastiera, arrivo di un ordine, passaggio di qualifica, …)
  - interno – generato da un oggetto del sistema

Analisi
Diagrammi di stato

- **Call Event**
  Occurs when an element receives a call for an operation
- **Signal Event**
  Occurs when an element receives an explicit signal from another element
- **Change Event**
  Occurs when a designated condition (usually described as a Boolean operation) becomes true
- **Time Event**
  Occurs after a designated period of time or at a specific time or date

Analisi
Diagrammi di stato

- **TRANSIZIONE**

  Relazione tra due stati
  S1 (stato di partenza) e S2 (stato di arrivo)
  indicante che

  - un oggetto inizialmente nello stato S1
  - in seguito al verificarsi di un particolare evento
  - e se sono soddisfatte certe condizioni,
  - effettuerà certe azioni ed
  - entrerà nello stato S2

Analisi
Diagrammi di stato

- AZIONE

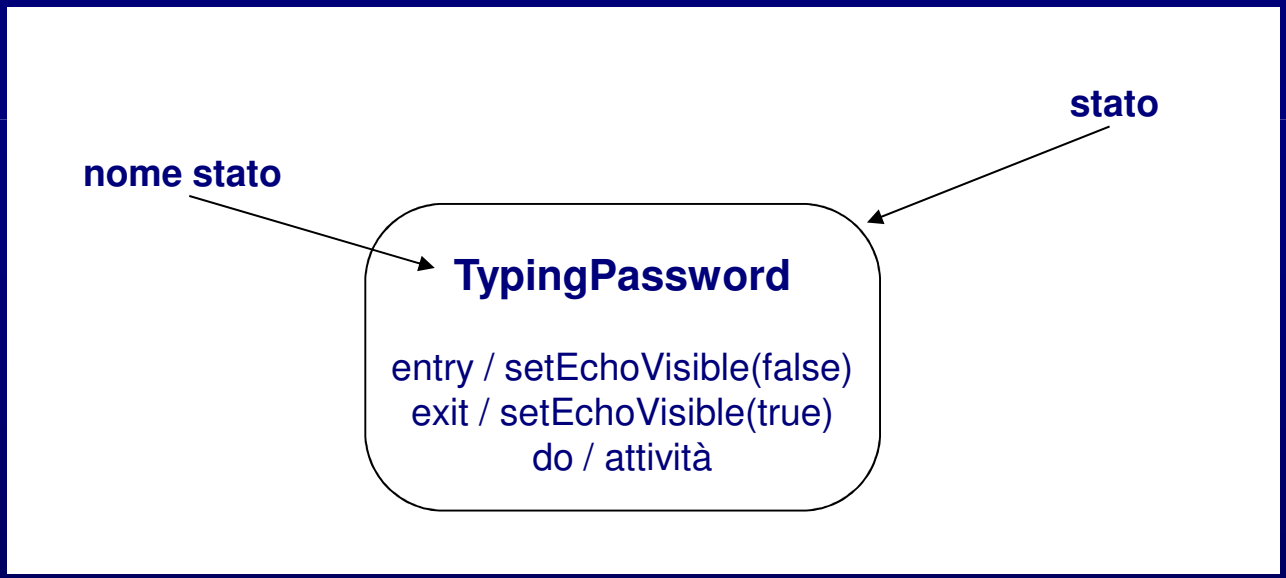
  Computazione atomica (non interrompibile)
  - associata a una transizione

- ATTIVITÀ

  Computazione non atomica (interrompibile)
  - associata a uno stato
  - insieme di azioni

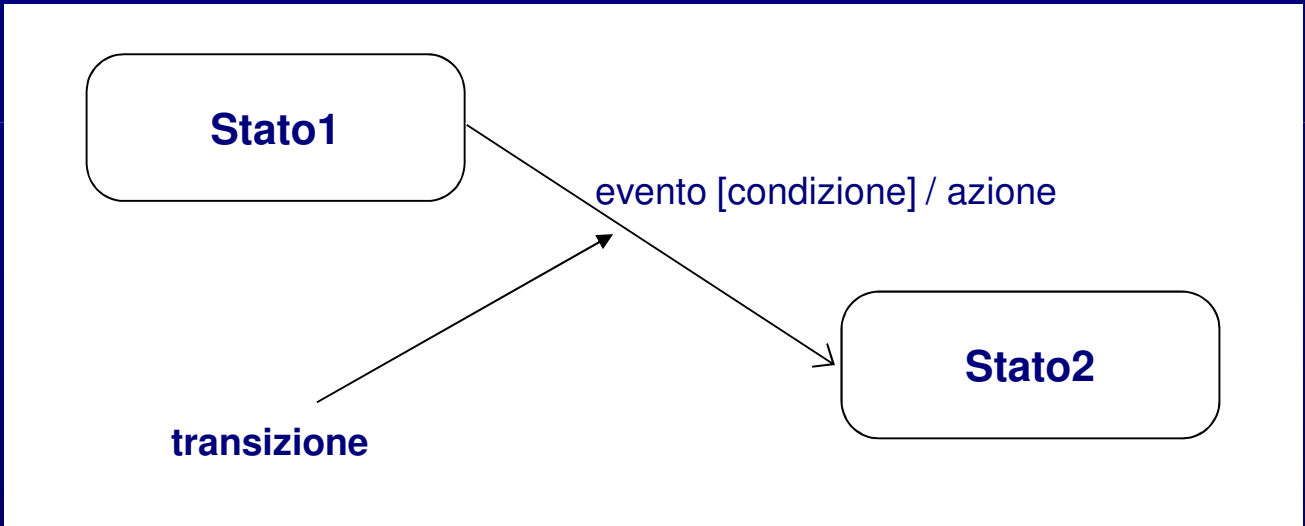
Analisi
Diagrammi di stato

- Notazione UML



Analisi
Diagrammi di stato

- Notazione UML



Analisi
Identificazione degli stati

- Esaminare i possibili valori degli attributi di un oggetto
- Determinare se il sistema prevede
  - comportamenti diversi
  - per valori diversi degli attributi
- Descrivere mediante un diagramma di stato
  - tutti i possibili stati che un oggetto può assumere durante la sua vita e
  - come cambia lo stato dell'oggetto in conseguenza degli eventi

Analisi
Diagrammi di stato