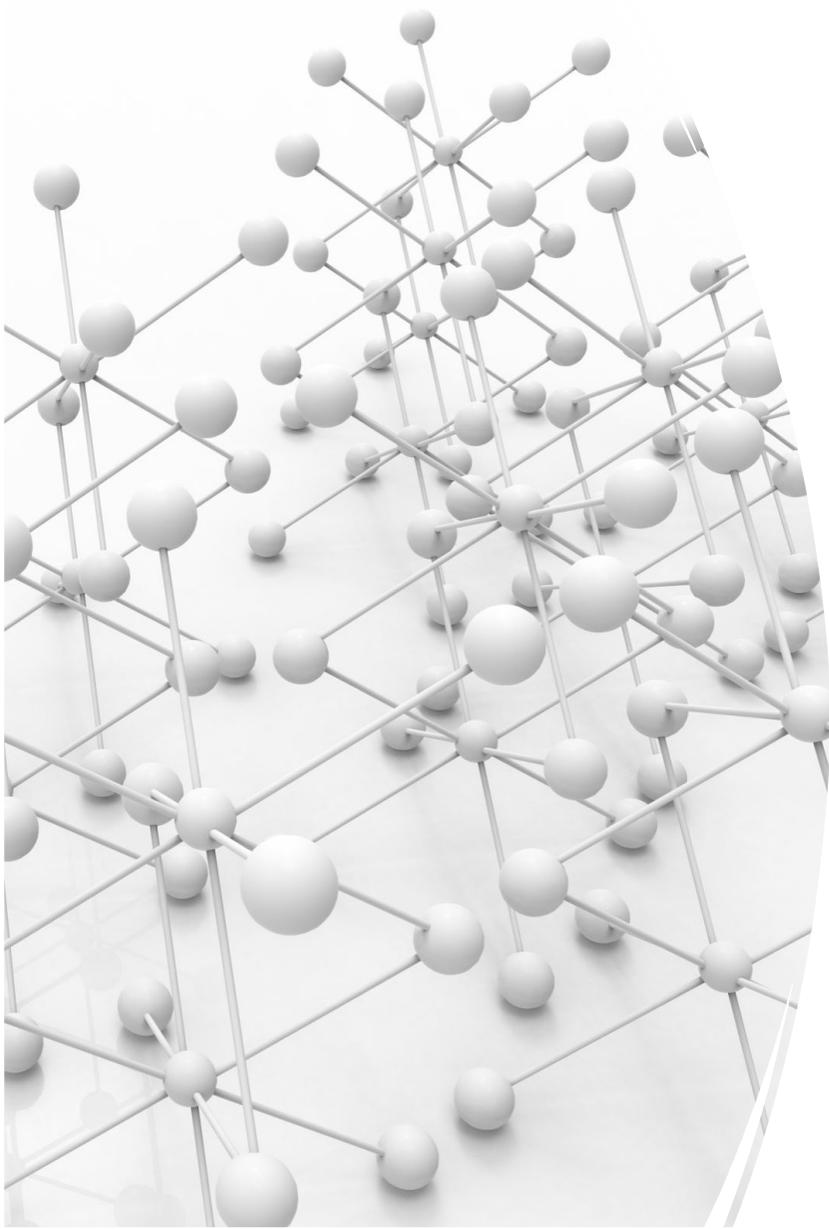


MODELLAZIONE CON UML: DIAGRAMMI DELLE CLASSI

Prof. Mariacarla Staffa



DIAGRAMMI DELLE CLASSI

- Classi ed oggetti
 - I diagrammi delle classi descrivono la struttura del sistema in termini di classi ed oggetti
 - Le **classi** sono astrazioni che specificano gli attributi ed il comportamento di un insieme di oggetti
 - Una classe è una collezione di oggetti che condividono un insieme di attributi che contraddistinguono gli oggetti come membri della collezione
 - Gli **oggetti** sono entità che incapsulano lo stato ed il comportamento
 - Ogni oggetto ha un'identità mediante la quale ci si riferisce ad esso individualmente e che lo distingue dagli altri oggetti
 - In UML, classi ed oggetti sono rappresentati da riquadri composti da tre compartimenti
 - Parte alta: nome della classe o dell'oggetto
 - Centro: attributi
 - Parte bassa: operazioni
 - Le parti relative agli attributi e alle operazioni possono essere omesse per chiarezza

CLASSI E OGGETTI

- Un **oggetto** è un'entità caratterizzata da
 - Un'identità
 - Uno stato
 - Un comportamento
- Una **Classe** descrive
 - Uninsieme di oggetti con caratteristiche simili
 - Cioè oggetti che hanno lo stesso tipo



CONVENZIONI UML PER CLASSI ED OGGETTI

- I nomi degli oggetti sono sottolineati per indicare che sono istanze
- I nomi delle classi iniziano con lettere maiuscole
- Agli oggetti, nei diagrammi degli oggetti, possono essere assegnati dei nomi (seguiti dalla loro classe) per semplificarne il riferimento
 - In questo caso, i nomi iniziano con lettere minuscole

ATTORE VS CLASSE VS OGGETTO



Attore

Un'entità da modellare esterna al sistema, interagisce con il sistema ('passenger')



Classe

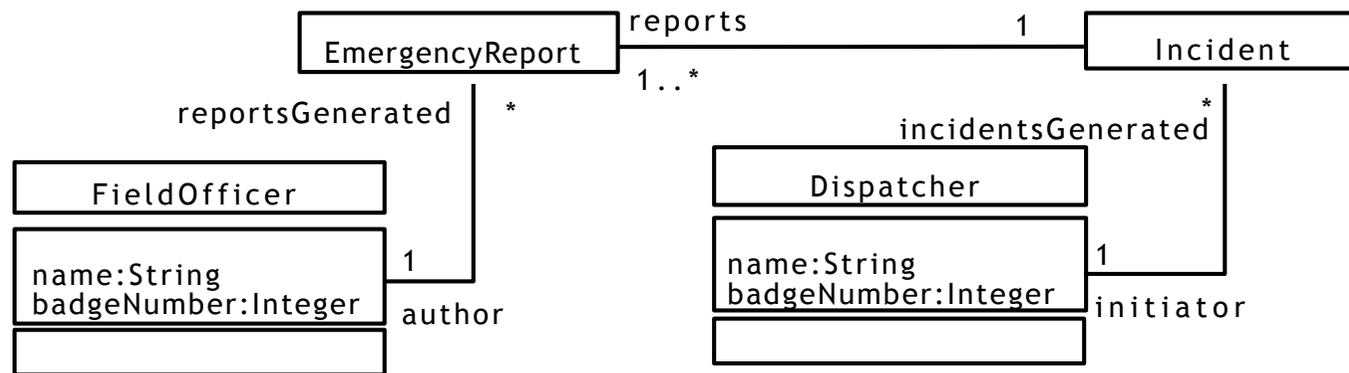
Un'astrazione che modella un'entità nel dominio applicativo o del dominio della soluzione ('User', 'Ticket distributor', 'Server')



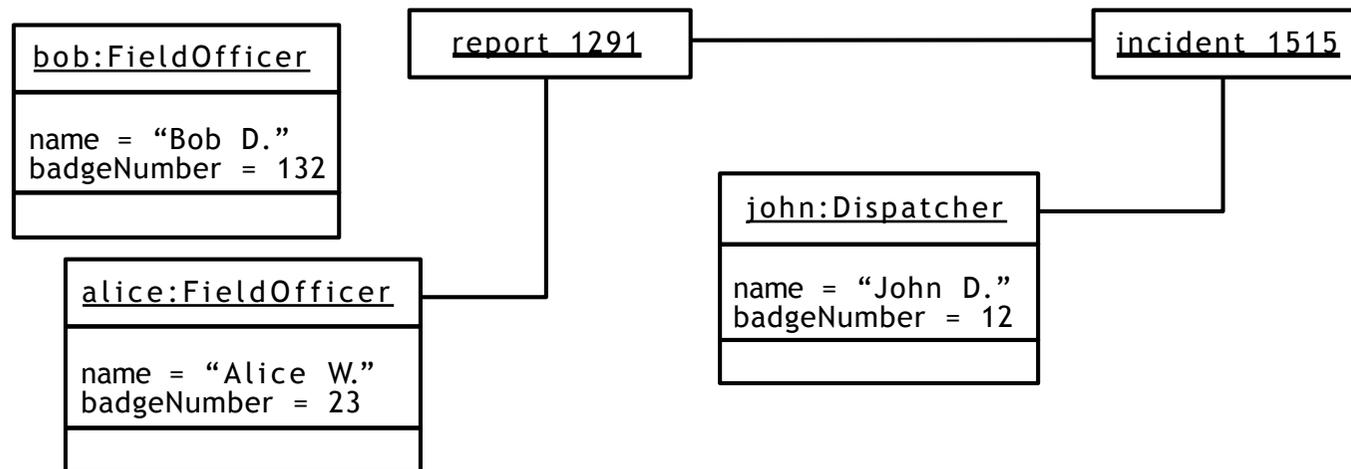
Oggetto

Una specifica istanza di una classe ('Joe, il passeggero che sta comprando un biglietto dal distributore di biglietti')

ESEMPIO DI DIAGRAMMA DELLE CLASSI: CLASSI PARTECIPANTI NEL CASO D'USO *ReportEmergency*



ESEMPIO DI DIAGRAMMA DEGLI OGGETTI: OGGETTI PARTECIPANTI NELLO SCENARIO `warehouseOnFire`



ASSOCIAZIONI E COLLEGAMENTI (LINK)

- Un collegamento o link rappresenta una connessione tra due oggetti
- Le associazioni sono relazioni tra classi e rappresentano gruppi di link
- Nell'esempio *FRIEND*, ogni oggetto *FieldOfficer* ha anche una lista di *EmergencyReport* che sono stati scritti dal *FieldOfficer*
- Nell'esempio del diagramma delle classi, la linea tra la classe *FieldOfficer* e la classe *EmergencyReport* è un'associazione
- Nell'esempio del diagramma degli oggetti, la linea tra l'oggetto *alice:FieldOfficer* e l'oggetto *report_1291:EmergencyReport* è un link
 - Rappresenta uno stato del sistema per cui *alice:FieldOfficer* ha generato *report_1291:EmergencyReport*



ASSOCIAZIONI SIMMETRICHE E ASIMMETRICHE

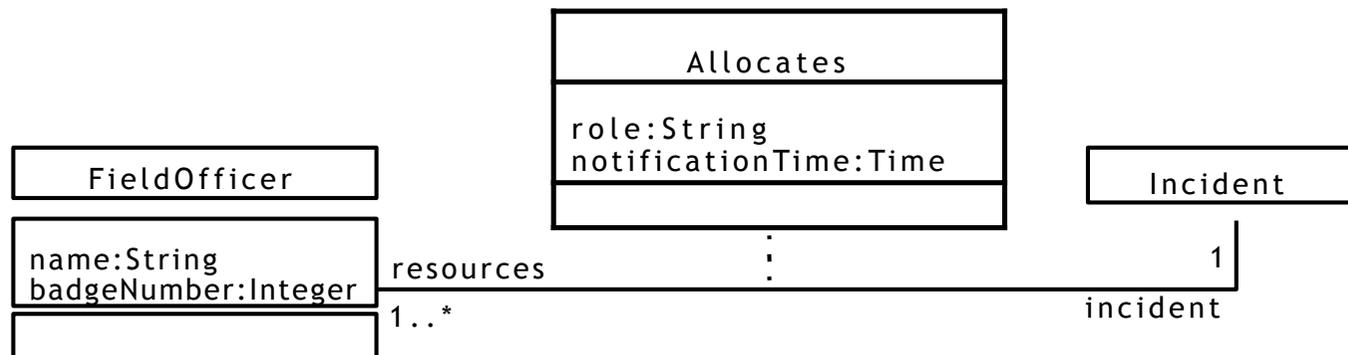
- Le associazioni possono essere bidirezionali (simmetriche) o unidirezionali (asimmetriche)
 - **Nei diagrammi precedenti sono tutte simmetriche**
- Una associazione asimmetrica è, ad esempio, quella tra le classi *Poligono* e *Punto*
 - **La freccia di navigazione indica che il sistema supporta solo il verso da poligono a punto**
 - Dato un poligono specifico, è possibile individuare tutti i punti che costituiscono il poligono.
 - Dato un punto specifico, non è possibile individuare il poligono di cui il punto fa parte.
 - Per convenzione, le associazioni senza frecce sono simmetriche



CLASSE DI ASSOCIAZIONE

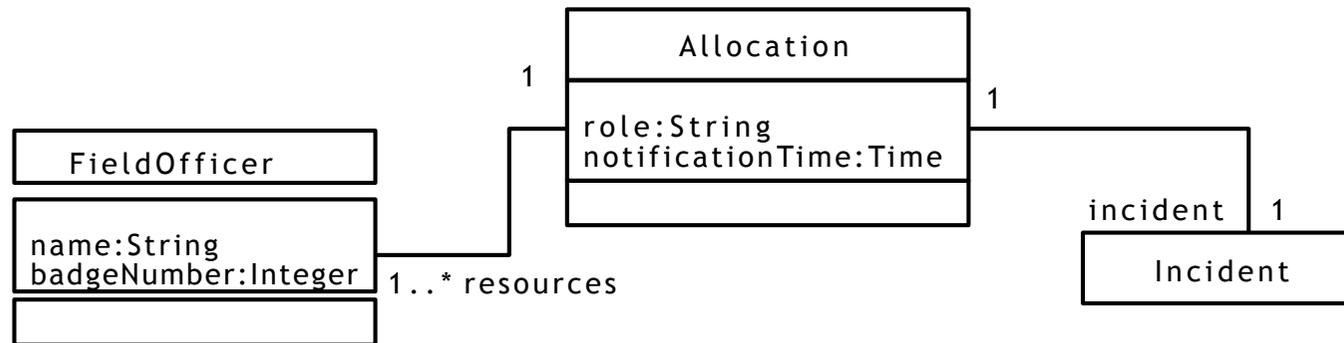
- Le associazioni sono simili alle classi poiché possono avere attributi ed operazioni
- Una tale associazione è chiamata classe di associazione
 - Disegnata con un simbolo di classe che contiene attributi e operazioni ed è connessa al simbolo di associazione con una linea tratteggiata
 - Ad esempio, l'allocazione dei *FieldOfficer* ad un Incidente è modellata come una classe di associazione con attributi ruolo (*role*) e ora di notifica (*notificationTime*)

ESEMPIO DI CLASSE DI ASSOCIAZIONE



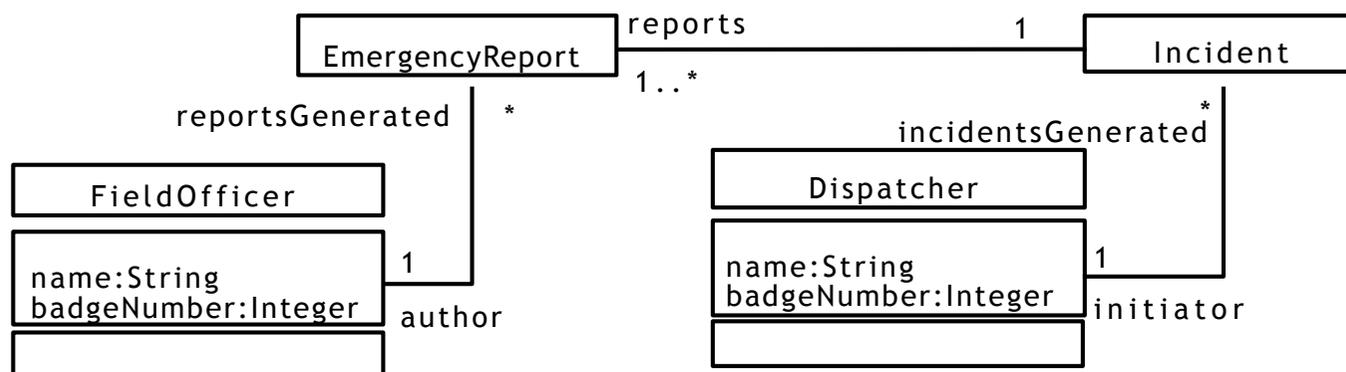
MODELLO ALTERNATIVO PER ALLOCATION

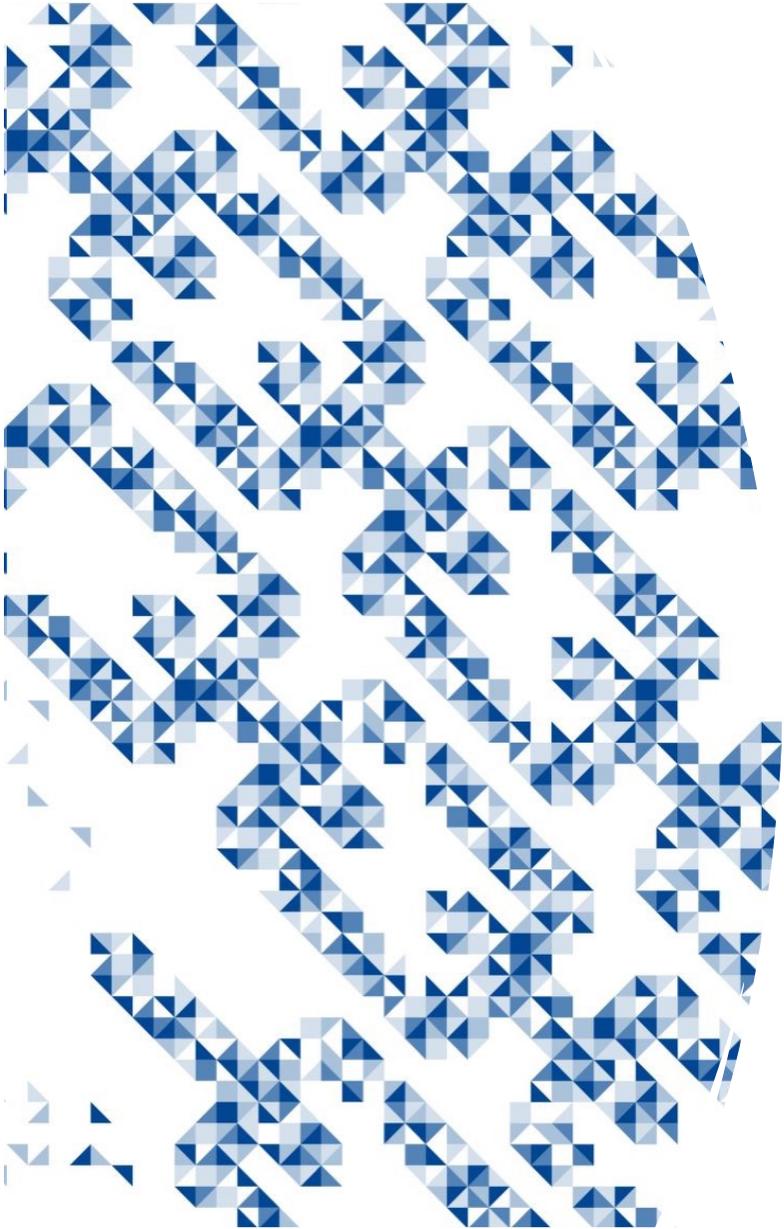
- Qualsiasi classe di associazione può essere trasformata in una classe e associazioni semplici



RUOLI

- Ciascuna estremità di un'associazione può essere etichettata con una stringa chiamata ruolo
- I ruoli dell'associazione tra le classi *EmergencyReport* e *FieldOfficer* sono autore (*author*) e rapporto generato (*reportGenerated*)
 - Etichettare le estremità delle associazioni con i ruoli consente di distinguere tra le multiple associazioni che si originano da una classe.
 - Inoltre, i ruoli chiariscono lo scopo dell'associazione

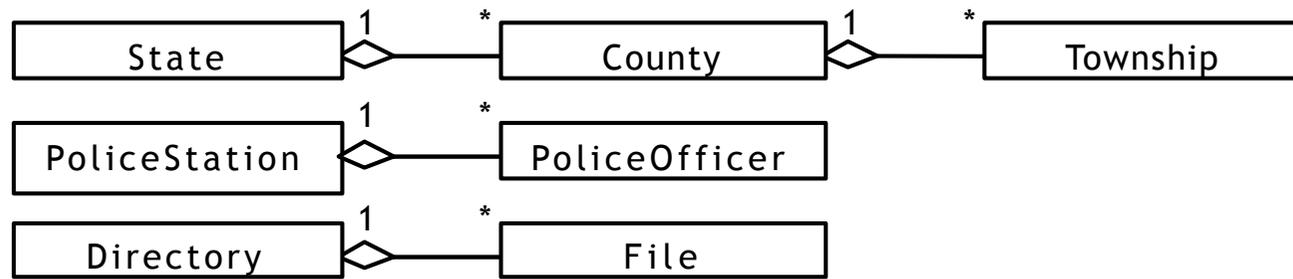




AGGREGAZIONE

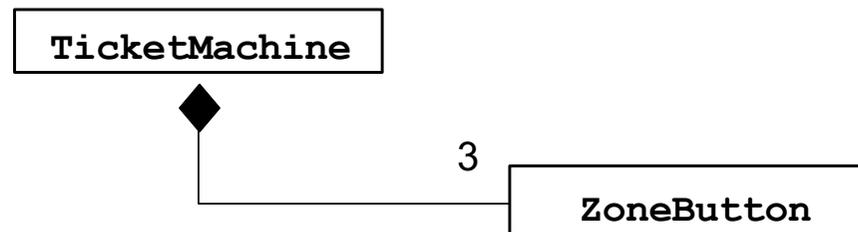
- Le associazioni sono usate per rappresentare un'ampia gamma di connessioni tra un insieme di oggetti
- Un tipo speciale di associazione si presenta frequentemente: le **aggregazioni** (denotate con una linea con testa di diamante)
 - Esempi:
 - uno *stato* contiene molti *paesi* che a loro volta contengono molte *città*
 - una *stazione di polizia* è costituita di un certo numero di *poliziotti*
 - una *directory* contiene un certo numero di *file*
 - Tali relazioni possono essere modellate con associazioni uno-a-molti
 - UML, invece, fornisce il concetto di aggregazione che consente di denotare aspetti gerarchici della relazione che può avere molteplicità sia uno-a-molti che molti-a-molti

ESEMPI DI AGGREGAZIONI



COMPOSIZIONE

- Un rombo solido denota una composizione
 - Una forma forte di aggregazione dove il tempo di vita delle istanze component è controllato dall'aggregato
 - Le parti non hanno esistenza autonoma ("il tutto controlla /distrugge le parti")

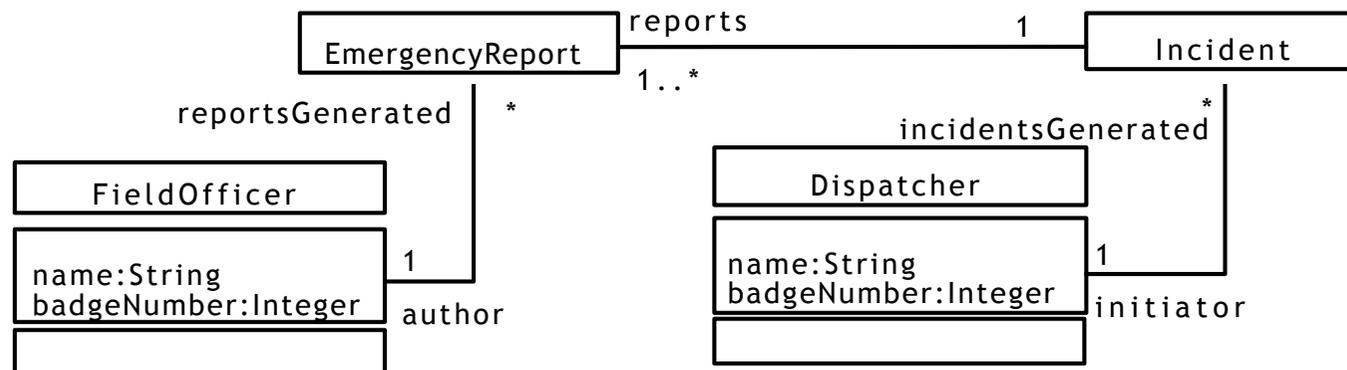


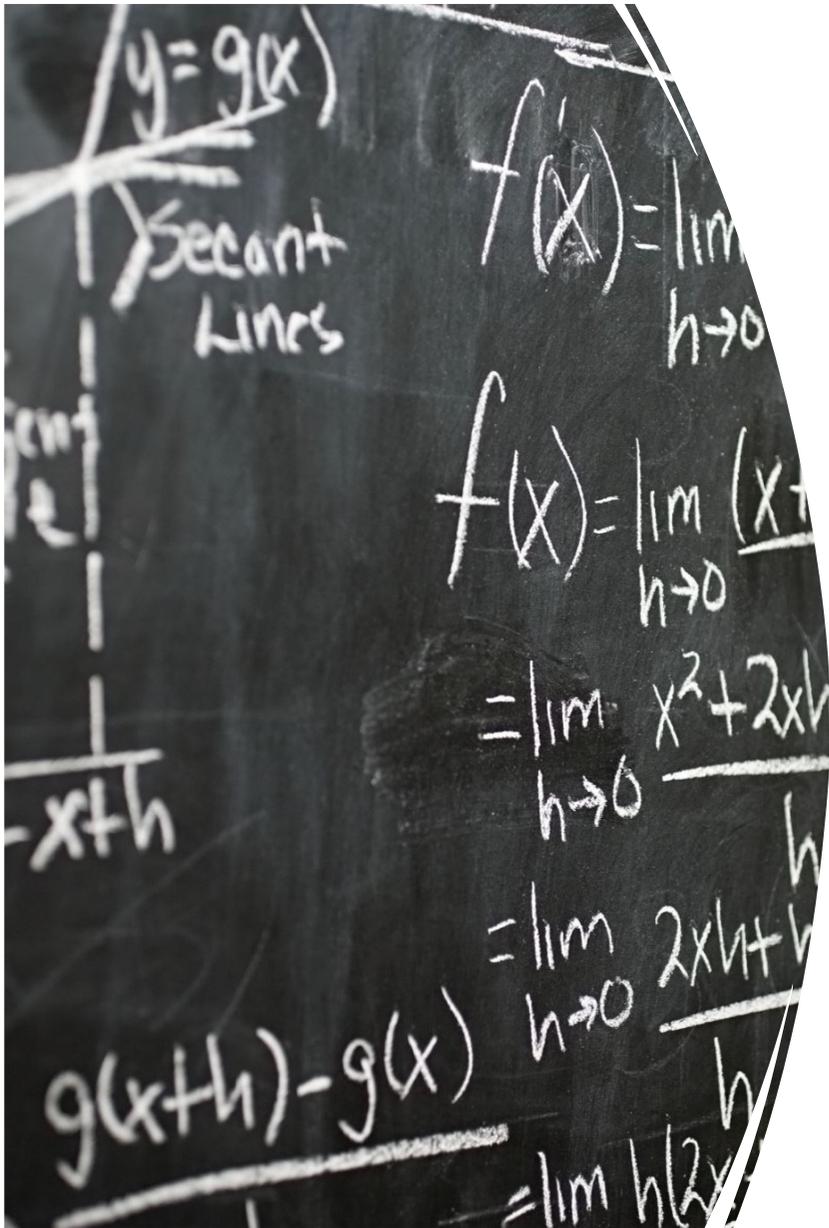
MOLTEPLICITÀ

- Ogni estremità di un'associazione può essere etichettata con un insieme di interi che indicano il numero di link che si originano da un'istanza della classe connessa all'estremità dell'associazione
- Questo insieme di interi è chiamato **molteplicità** dell'estremità dell'associazione

ESEMPIO MOLTEPLICITÀ

- L'estremità dell'associazione *author* ha una molteplicità pari ad 1
 - Significa che tutti gli *EmergencyReport* sono scritti esattamente da un *FieldOfficer* vale a dire, ogni oggetto *EmergencyReport* ha esattamente un link ad un oggetto della classe *FieldOfficer*
 - La molteplicità dell'estremità dell'associazione *reportsGenerated* è "molti" ed indicata con un asterisco (*) che indica 0..n

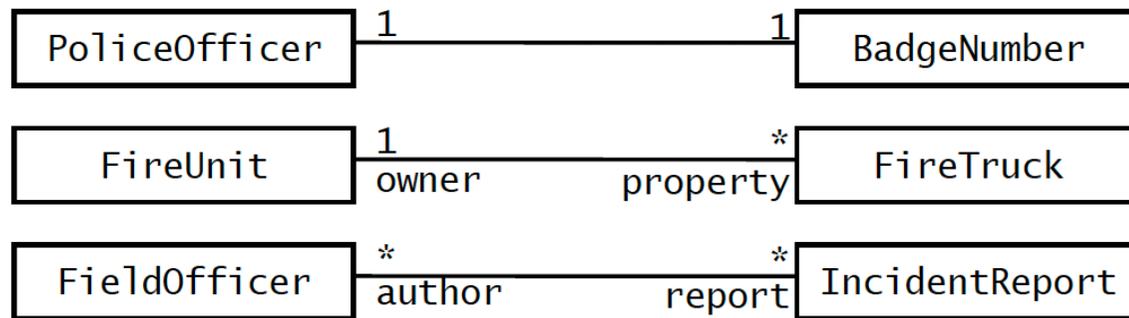




TIPI DI MOLTEPLICITÀ

- Le associazioni solitamente usate nei diagrammi sono di tre tipi:
 - **Associazione uno-a-uno**
 - Ha molteplicità 1 su entrambe le estremità: esiste esattamente un link tra le istanze di ogni classe
 - **Associazione uno-a-molti**
 - Ha molteplicità 1 su di una estremità e 0..n o 1..n dall'altra estremità: denota il rapporto di composizione tra due classi
 - **Associazione molti-a-molti**
 - Ha molteplicità 0..n o 1..n su ambo i lati: indica che un numero arbitrario di link possono esserci tra le istanze di due classi

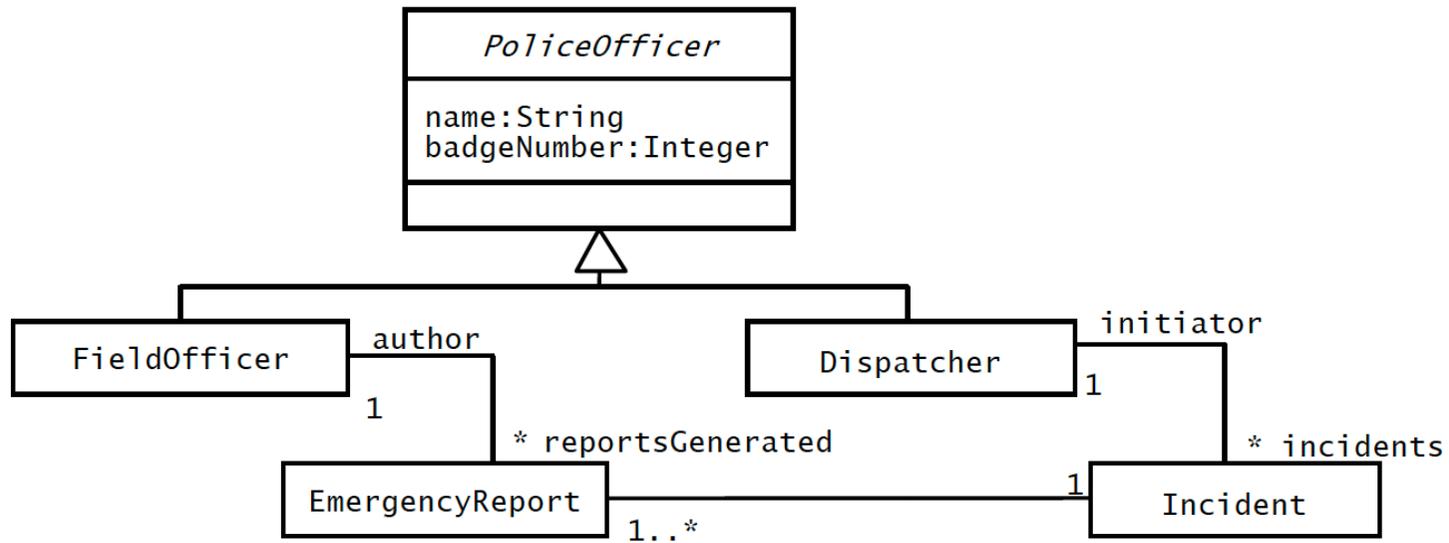
ESEMPI DI MOLTEPLICITÀ



EREDITARIETÀ

- L'**ereditarietà** è la relazione tra una classe generale ed una o più classi specializzate
- L'ereditarietà consente di descrivere tutti gli attributi e le operazioni che sono comuni ad un insieme di classi
 - Esempio: *FieldOfficer* e *Dispatcher* hanno entrambi gli attributi *name* e *badgeNumber*. Tuttavia, *FieldOfficer* ha un'associazione con *EmergencyReport*, mentre *Dispatcher* ha un'associazione con *Incident*. Gli attributi comuni di *FieldOfficer* e *Dispatcher* possono essere modellati introducendo una classe *PoliceOfficer* che è specializzata da *FieldOfficer* e *Dispatcher*
 - *PoliceOfficer* è chiamata la **superclasse**; *FieldOfficer* e *Dispatcher*
 - sono chiamate **sottoclassi**
- Le sottoclassi ereditano gli attributi e le operazioni della loro superclasse

ESEMPIO DI GENERALIZZAZIONE



CLASSI ASTRATTE

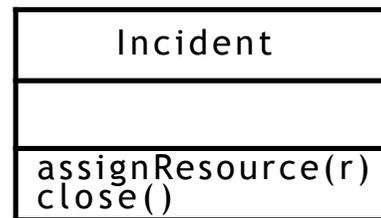
- La classe *PoliceOfficer* è una classe astratta. Per distinguerla dalle classi concrete si scrive il nome in corsivo
 - Le classi astratte sono utilizzate nella modellazione orientata agli oggetti per classificare concetti collegati riducendo, quindi, la complessità totale del modello
 - Eliminano la ridondanza

OGGETTI E OPERAZIONI

- Il comportamento di un oggetto è specificato dalle **operazioni**
- Un oggetto richiede l'esecuzione di un'operazione ad un altro oggetto inviandogli un messaggio
- Il messaggio è confrontato con il metodo definito dalla classe a cui l'oggetto ricevente appartiene o da una qualsiasi sua superclasse
- I metodi di una classe in un linguaggio di programmazione orientato agli oggetti sono le implementazioni di queste operazioni

ESEMPIO DI OPERAZIONI DELLA CLASSE

Incident



APPLICAZIONE DEI DIAGRAMMI DELLE CLASSI

- Usati per descrivere la struttura del sistema
- Durante la fase di analisi gli ingegneri (del software) costruiscono diagrammi delle classi per formalizzare la conoscenza del dominio dell'applicazione
- Le classi rappresentano gli oggetti partecipanti individuati nei diagrammi dei casi d'uso e di interazione
 - Descrivono i loro attributi e le operazioni
- Lo scopo dei modelli di analisi è di descrivere il proposito del sistema e scoprire i suoi confini
 - Ad esempio, un analista può esaminare la molteplicità dell'associazione tra *FieldOfficere* e *EmergencyReport* e chiedere all'utente se ciò è corretto
 - E' possibile avere più di un autore per l'*EmergencyReport*?
 - Sono previsti rapporti anonimi?
- La fase di analisi tiene fuori concetti implementativi
 - I diagrammi di classe sono rifiniti durante la fase di progettazione del sistema e degli oggetti