

INGEGNERIA DEL SOFTWARE - 2025-26

DIAGRAMMA DELLE CLASSI: COSTRUZIONE DEL DIAGRAMMA DELLE CLASSI

LEZIONE 14
12/11/2025
VINCENZO RICCIO



- I diagrammi UML possono essere creati in diverse fasi del ciclo di vita, con diversi scopi e livelli di dettaglio
- **System domain model (Modello Concettuale del Dominio):**
 - Modella le entità del dominio del problema che saranno presenti nel sistema (i dati di interesse)
 - Possono contenere le responsabilità delle classi (opzionale)
- **System model (Modello del Sistema):**
 - Include anche le classi che saranno usate per costruire il sistema completo: classi per l'interfaccia utente (es. menu), classi associate a parti dell'architettura (client, server, file, database), classi di utilità
 - Contiene metodi e informazioni sulla navigabilità



COSTRUZIONE DEL SYSTEM DOMAIN MODEL

PROCESSO DI ASTRAZIONE DEL SYSTEM DOMAIN MODEL



1. Identifica un primo insieme di **classi** candidate
2. Aggiungi **associazioni** ed **attributi** a queste classi
3. Trova le **generalizzazioni**
4. Trova le principali **responsabilità** di ogni classe
5. Itera il processo finché il modello ottenuto è soddisfacente

1. IDENTIFICA LE CLASSI



- Bisogna porsi la domanda *"Quali sono le classi che fanno parte del dominio?"*
- In generale, le classi dovrebbero corrispondere a entità del dominio del problema
- In pratica, bisogna applicare i concetti generali di modellazione Object Oriented: una classe identifica un tipo di dato astratto

1. IDENTIFICA LE CLASSI



- Una buona classe:
 - ⦿ Ha un nome che ne rispecchia l'intento
 - ⦿ È un'astrazione che modella un elemento del dominio del problema
 - ⦿ Ha un insieme ridotto e ben definito di responsabilità

Libro
titolo autore genere ISBN disponibile
--Controlla Disponibilità() --Presta Libro()

"Libro" rispecchia immediatamente l'intento della classe, ovvero rappresentare un singolo volume all'interno del sistema di biblioteca

"Libro" è un'astrazione del dominio, che rappresenta l'entità fisica o digitale dei libri disponibili in biblioteca

Contiene informazioni sul titolo, autore, genere e ISBN, ma non gestisce operazioni non pertinenti, come la gestione degli utenti

1. IDENTIFICA LE CLASSI: ANALISI DEI NOMI



- Semplice tecnica per scoprire le classi del dominio
 1. Analizza la documentazione di partenza (es. specifica dei requisiti)
 2. Elenca i nomi
 3. Elimina nomi che:
 - Sono ridondanti (rappresentano la stessa classe)
 - Rappresentano o caratterizzano istanze e non classi
 - Sono vaghi, troppo generici
 - Corrispondono a classi che non sono necessarie al livello considerato

1. IDENTIFICA LE CLASSI: ANALISI DEI NOMI



- *Il sistema consente a un utente di aprire un conto corrente, diventandone il titolare. Il titolare può consultare il proprio saldo attuale.*
- I sostantivi possono indicare classi, attributi e ruoli
- **Classe:** il sostantivo indica un insieme di elementi distinti
 - *Es: Conto Corrente*
- **Attributo:** indica una proprietà di un elemento di un insieme
 - *Es.: Saldo (proprietà di un Conto Corrente)*
- **Ruolo:** il sostantivo indica il ruolo che un'istanza di una classe assume quando la si considera come parte di un legame con altri oggetti
 - *Es.: La Persona che ha aperto un Conto Corrente ne diventa il titolare*

1. IDENTIFICA LE CLASSI: ANALISI DEI NOMI



- Il **sistema** consente agli **utenti** di cercare **libri** per **titolo**, **autore** e **genere**.
 - Gli **utenti** possono prendere in prestito e restituire **libri**.
 - Gli **amministratori** del sistema possono aggiungere, modificare e rimuovere libri dal **catalogo**.
 - Il **sistema** traccia lo stato di disponibilità di ciascun **libro**.
 - Gli **utenti** possono iscriversi al **sistema**, creare un **profilo** e accedere alla **lista** dei **libri** in prestito.
-
- Elenco dei nomi: Sistema, Utente, Libro, Titolo, Autore, Genere, Catalogo, Amministratore, Profilo, Cronologia, Lista dei libri in prestito

1. IDENTIFICA LE CLASSI: ANALISI DEI NOMI



- Il sistema consente agli **utenti** di cercare libri per titolo, autore e genere.
 - Gli **utenti** possono prendere in prestito e restituire **libri**.
 - Gli **amministratori** del sistema possono aggiungere, modificare e rimuovere libri dal **catalogo**.
 - Il **sistema** traccia lo stato di disponibilità di ciascun **libro**.
 - Gli **utenti** possono iscriversi al **sistema**, creare un **profilo** e accedere alla **lista** dei **libri** in prestito.
-
- Sistema identifica l'intero sistema e non classi interne ad esso
 - Libro, Titolo, Autore, Genere caratterizzano l'istanza di libro
 - La lista dei libri può essere un attributo del profilo utente

1. IDENTIFICA LE CLASSI: ANALISI DEI NOMI



- Il **sistema** consente agli utenti di cercare libri per titolo, autore e genere.
 - Gli utenti possono prendere in prestito e restituire **libri**.
 - Gli **amministratori** del sistema possono aggiungere, modificare e rimuovere libri dal **catalogo**.
 - Il **sistema** traccia lo stato di disponibilità di ciascun **libro**.
 - Gli utenti possono iscriversi al **sistema**, creare un **profilo** e accedere alla **lista** dei **libri** in prestito.
-
- Profilo e utente sono ridondanti, possiamo utilizzare un'unica classe Profilo Utente che rispecchia l'informazione utile al sistema

1. IDENTIFICA LE CLASSI: ANALISI DEI NOMI



- ▶ Gli amministratori del sistema possono aggiungere, modificare e rimuovere libri dal **catalogo**.
- ▶ Gli utenti possono iscriversi al sistema, creare un **profilo** e accedere alla **lista** dei **libri** in prestito.
- ▶ Un errore comune consiste nel considerare il sistema e tutti gli utenti del sistema come se fossero classi
- ▶ Attenzione a classi nel domain model che rappresentano tipi di utente o altri attori
- ▶ Bisogna includere tali classi solo se occorre manipolare o salvare loro informazioni (Profilo Utente), altrimenti sono esterni al sistema (Amministratore)

2. IDENTIFICA ATTRIBUTI

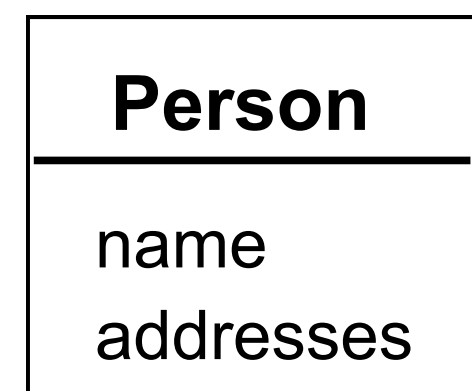


- Il sistema consente agli utenti di cercare libri per **titolo, autore e genere**.
- Cercare le informazioni che devono essere conservate per ciascuna classe
- È possibile che nomi che sono stati scartati come classi al passo 1, possano ora essere considerati attributi
 - Nel nostro esempio, titolo, autore e genere sono attributi di libro di tipo stringa

2. IDENTIFICA ATTRIBUTI

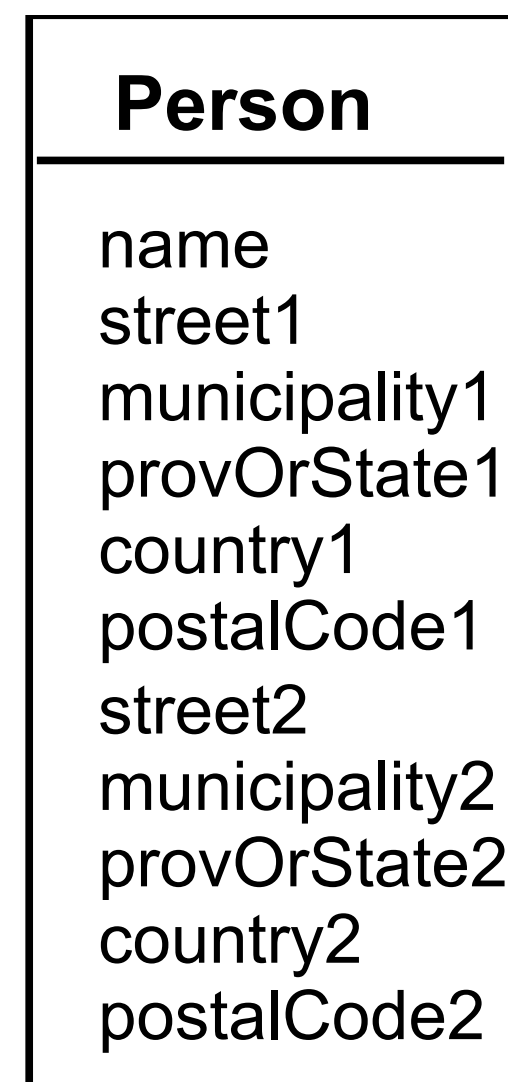


- Una classe non dovrebbe avere troppi attributi
- Bisogna stare attenti quando un attributo contiene molteplici valori
- Delicato trovare un equilibrio tra classi e attributi delle classi

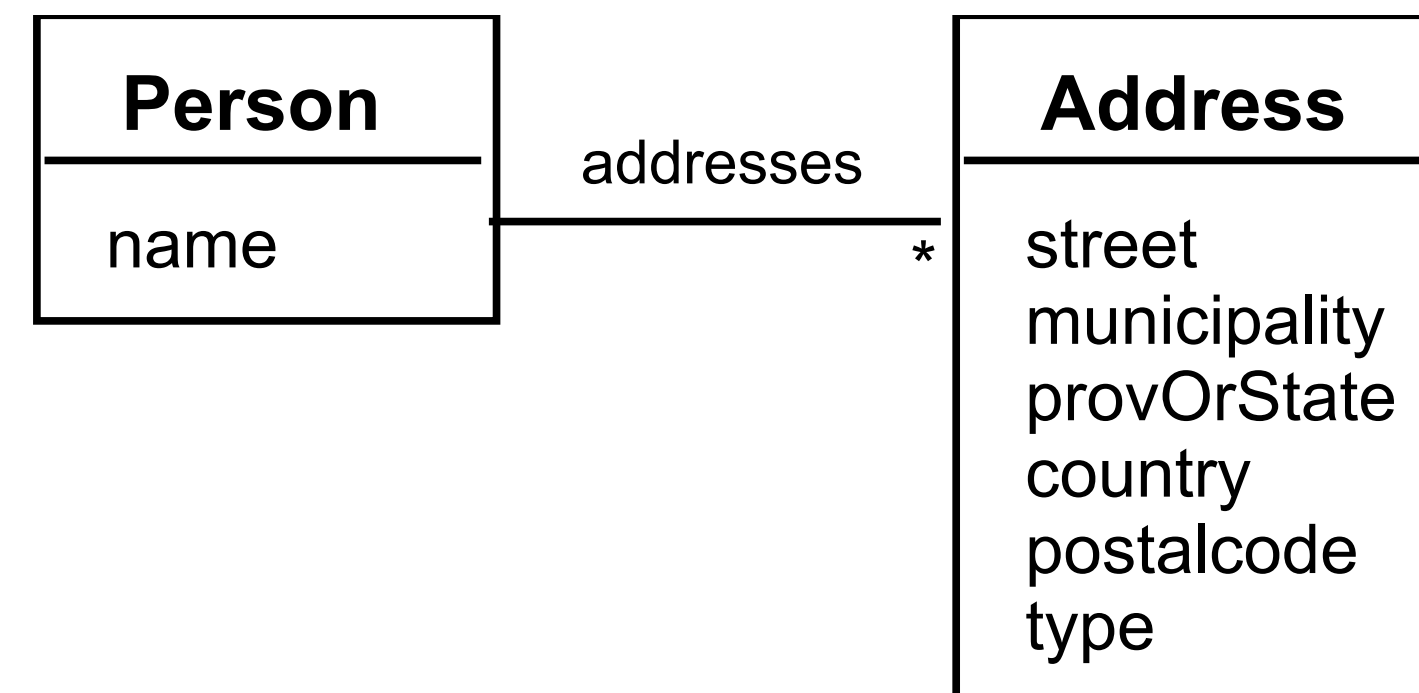


Bad due to
a plural
attribute

**NO ATTRIBUTI
RIPETUTI!**



Bad due to too many
attributes, and inability
to add more addresses



Good solution. The
type indicates whether
it is a home address,
business address etc.

**SE UNA CLASSE
POSSIEDE TROPPI
ATTRIBUTI E UN
SOTTOINSIEME DI ESSI
FORMA UN GRUPPO
COERENTE, PUOI
CREARE UNA CLASSE
DISTINTA PER QUESTI
ATTRIBUTI**

2. IDENTIFICA ASSOCIAZIONI



- Le espressioni verbali che coinvolgono più classi indicano possibili relazioni tra esse
- Un'associazione dovrebbe esistere se una classe:
 - ⊙ possiede o controlla
 - ⊙ è collegata a, oppure si riferisce a
 - ⊙ è parte di, oppure ha come parti
 - ⊙ è membro di, oppure ha come membri qualche altra classe del modello
- Cercare nell'ordine:
 1. **Gen-Spec:** Espressioni del tipo "è un"
 2. **Contenimento:** Espressioni del tipo "è fatto di", "comprende"
 3. **Associazione:** Ogni altra espressione

2. IDENTIFICA ASSOCIAZIONI

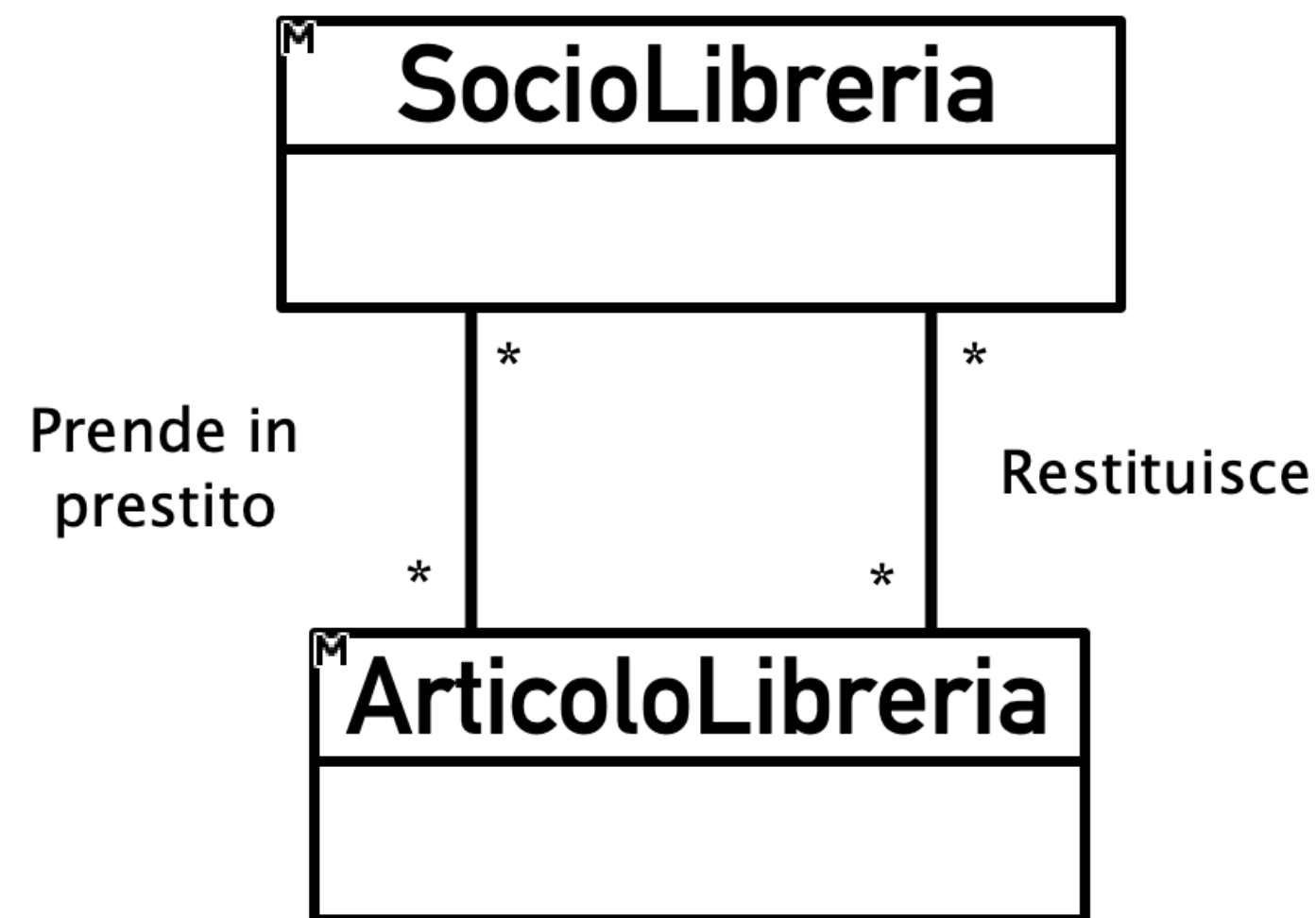


- Una volta identificata un'associazione:
 - ◎ Specificare le molteplicità da entrambi i lati
 - ◎ Assegnare un nome chiaro all'associazione e/o definire i ruoli delle entità che partecipano all'associazione
- **Nota:** È possibile che nomi che sono stati scartati come classi al passo 1 e come attributi, possano ora essere considerati ruoli

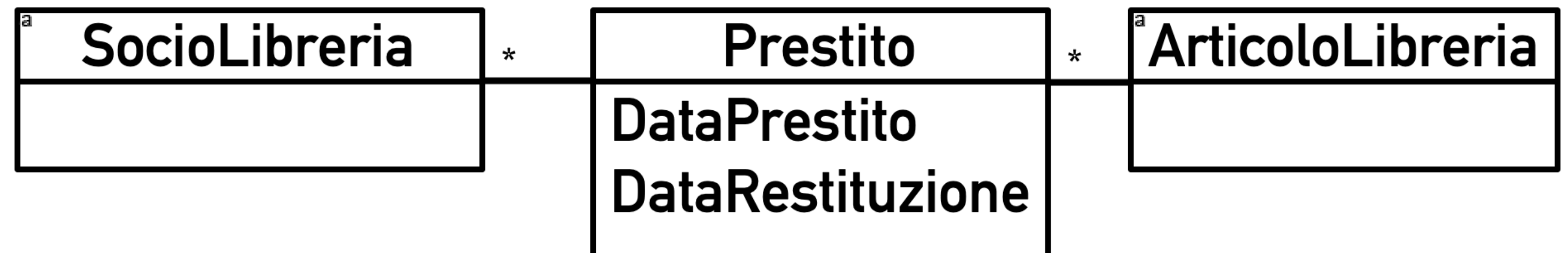
2. IDENTIFICA ASSOCIAZIONI: ERRORE COMUNE



- Un errore comune consiste nel considerare azioni come se fossero associazioni



L'OPERAZIONE
PRENDI_IN_PRESTITO() CREA
UN PRESTITO E SETTA LA DATA DI
PRESTITO E L'OPERAZIONE
RESTITUISCI() SETTA LA DATA DI
RESTITUZIONE



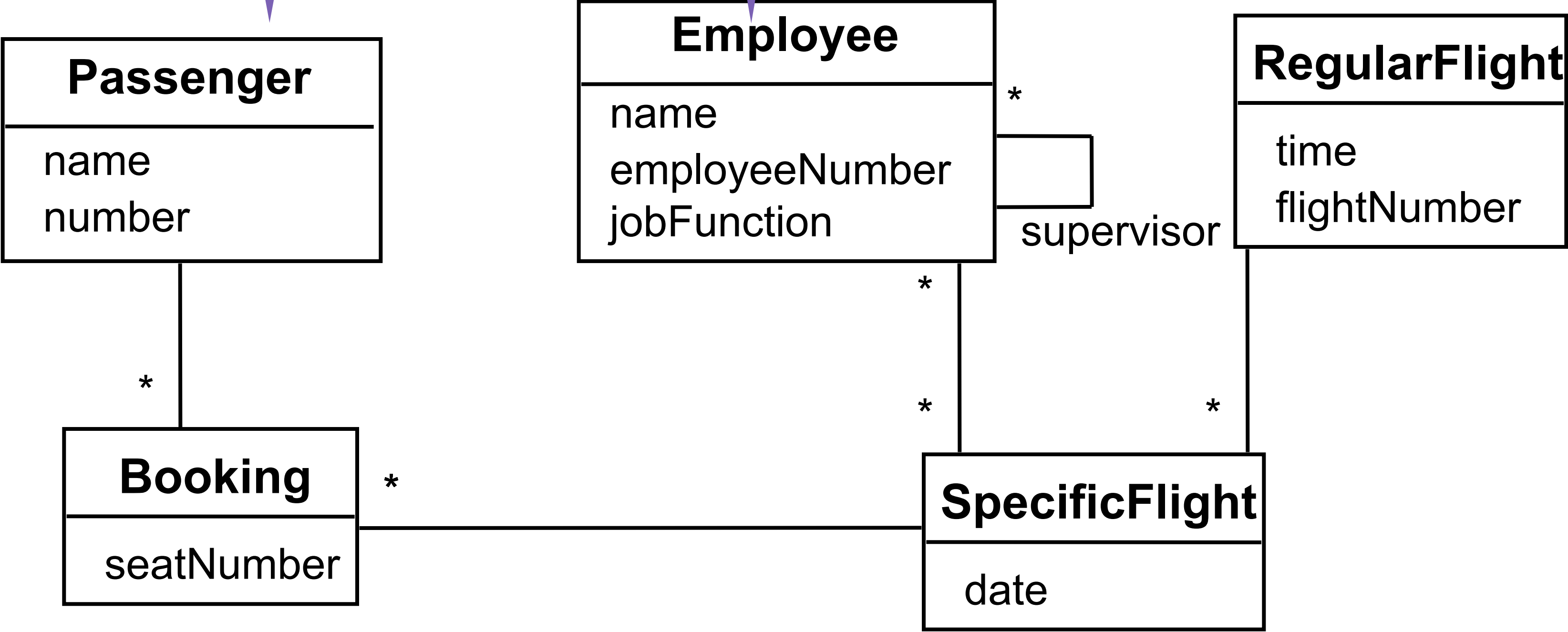
3. IDENTIFICA GENERALIZZAZIONI E INTERFACCE



- Due modi per trovare le **generalizzazioni**
 - ⦿ Bottom-Up: raggruppare classi simili creando una nuova super-classe
 - ⦿ Top-Down: cercare prima le classi più generali, e poi specializzare
- Crea un'**interfaccia**, invece di una superclasse se:
 - ⦿ Le classi sono molto diverse fra loro, tranne che per poche operazioni in comune
 - ⦿ Potrebbero essere disponibili diverse implementazioni della stessa classe
- **Nota:** Vedremo meglio le interfacce nella prossima lezione

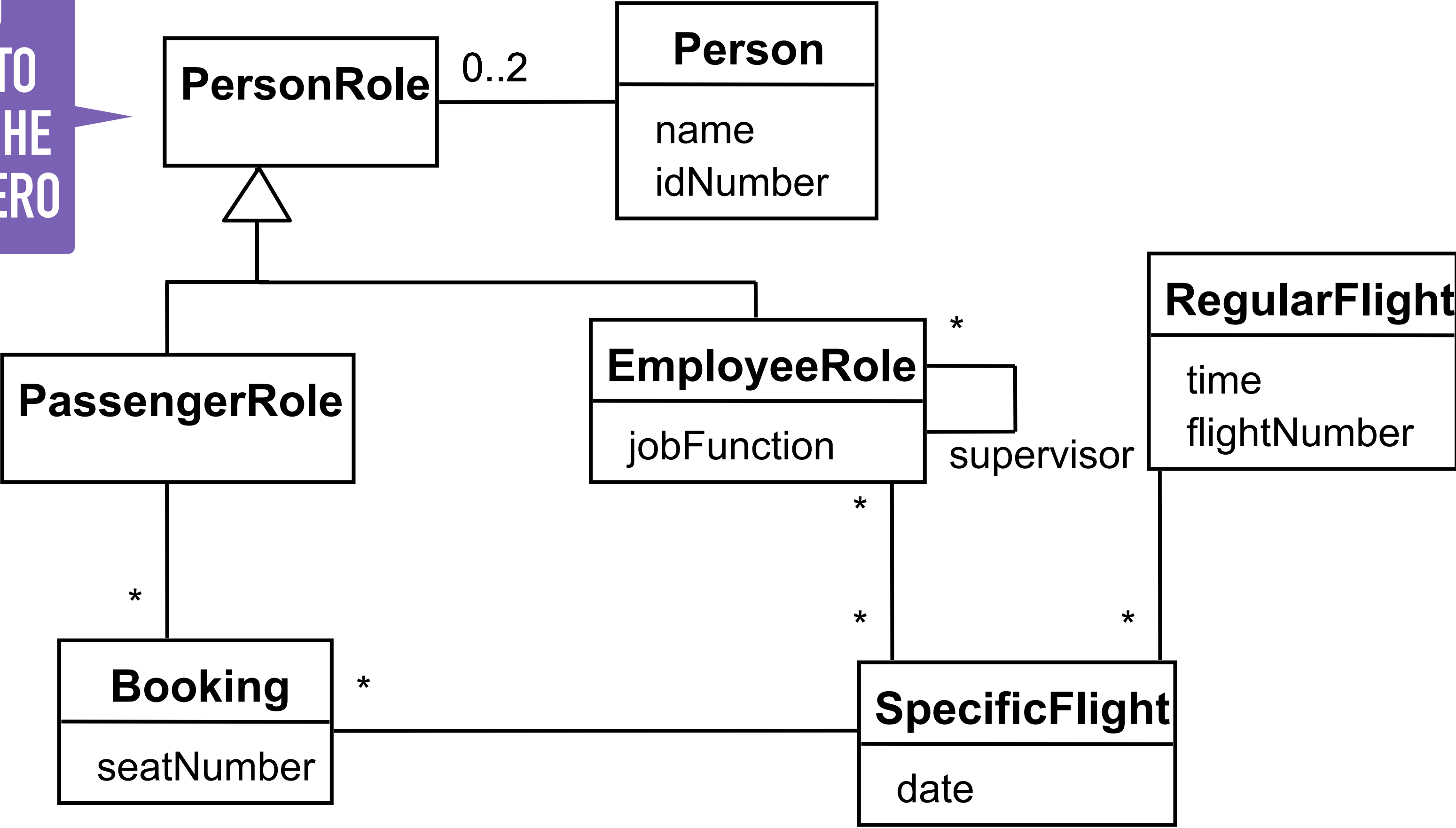
ESEMPIO: COMPAGNIA AEREA

PROBLEMA: EMPLOYEE E PASSENGER HANNO ALCUNI ATTRIBUTI SIMILI. INOLTRE, UN IMPIEGATO PUÒ PRENOTARE COME PASSEGGERO



ESEMPIO: COMPAGNIA AEREA CON GENERALIZZAZIONE

CLASSE PERSON CHE PUÒ
ASSUMERE DUE RUOLI, VISTO
CHE UN IMPIEGATO PUÒ ANCHE
PRENOTARE COME PASSEGGERO



4. TROVA LE PRINCIPALI RESPONSABILITÀ DI OGNI CLASSE



- Una **responsabilità** corrisponde ad una funzionalità richiesta al sistema software
- La responsabilità di ogni requisito funzionale deve essere attribuita ad una delle classi, anche se nella pratica tale requisito potrà essere svolto mediante una collaborazione fra più classi
- **Consigli:**
 - Se una classe ha troppe responsabilità, valutare l'ipotesi di dividerla in più classi
 - Se una classe non ha responsabilità, probabilmente è inutile
 - Quando una responsabilità non può essere attribuita a nessuna delle classi esistenti, dovrebbe essere creata una nuova classe

4. TROVA LE PRINCIPALI RESPONSABILITÀ DI OGNI CLASSE

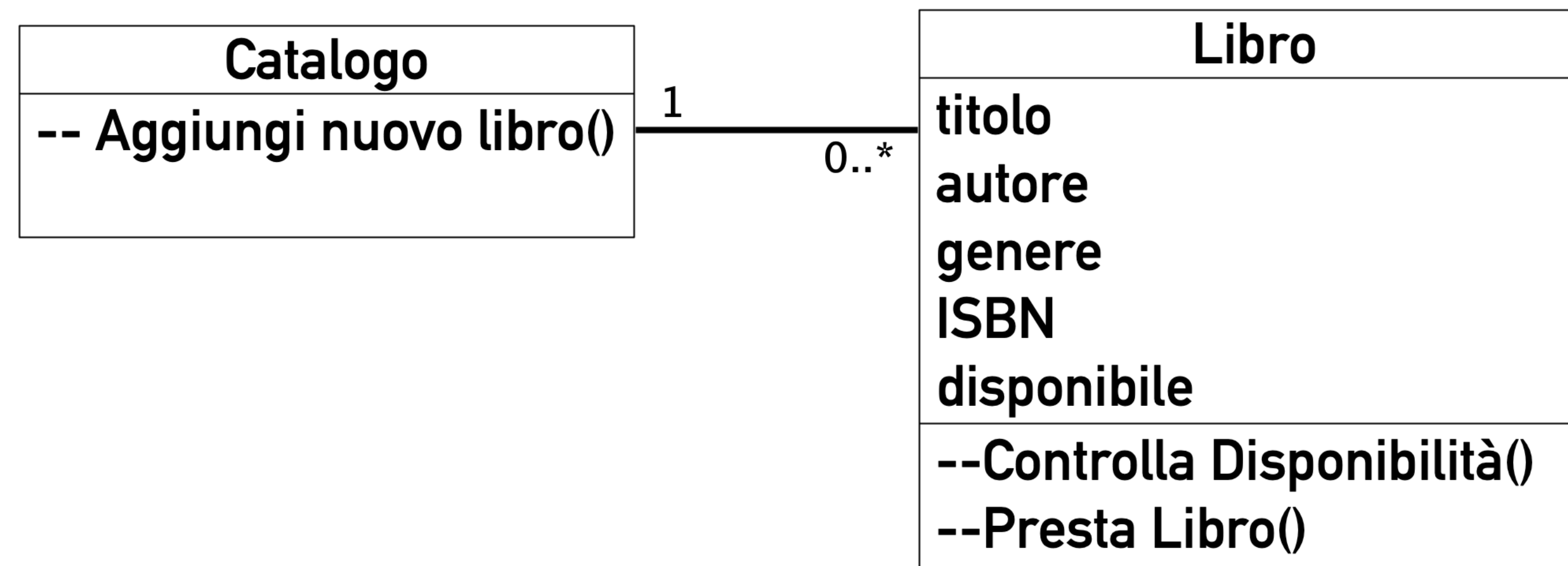


- Per stabilire le responsabilità dei requisiti funzionali:
 - Svolgere l'analisi dei casi d'uso
 - Cercare verbi e nomi che descrivono azioni nella descrizione del sistema
- Esistono inoltre alcune categorie di responsabilità generiche in un class diagram:
 - Accedere e modificare valori degli attributi
 - Creare ed inizializzare nuove istanze
 - Prelevare da o memorizzare dati in una memoria persistente
 - Distruggere istanze
 - Aggiungere e cancellare istanze di associazioni
 - Copiare, convertire, trasformare, trasmettere o fornire dati in output
 - Calcolare risultati numerici
 - Navigare e cercare dati di particolari istanze

4. TROVA LE PRINCIPALI RESPONSABILITÀ DI OGNI CLASSE



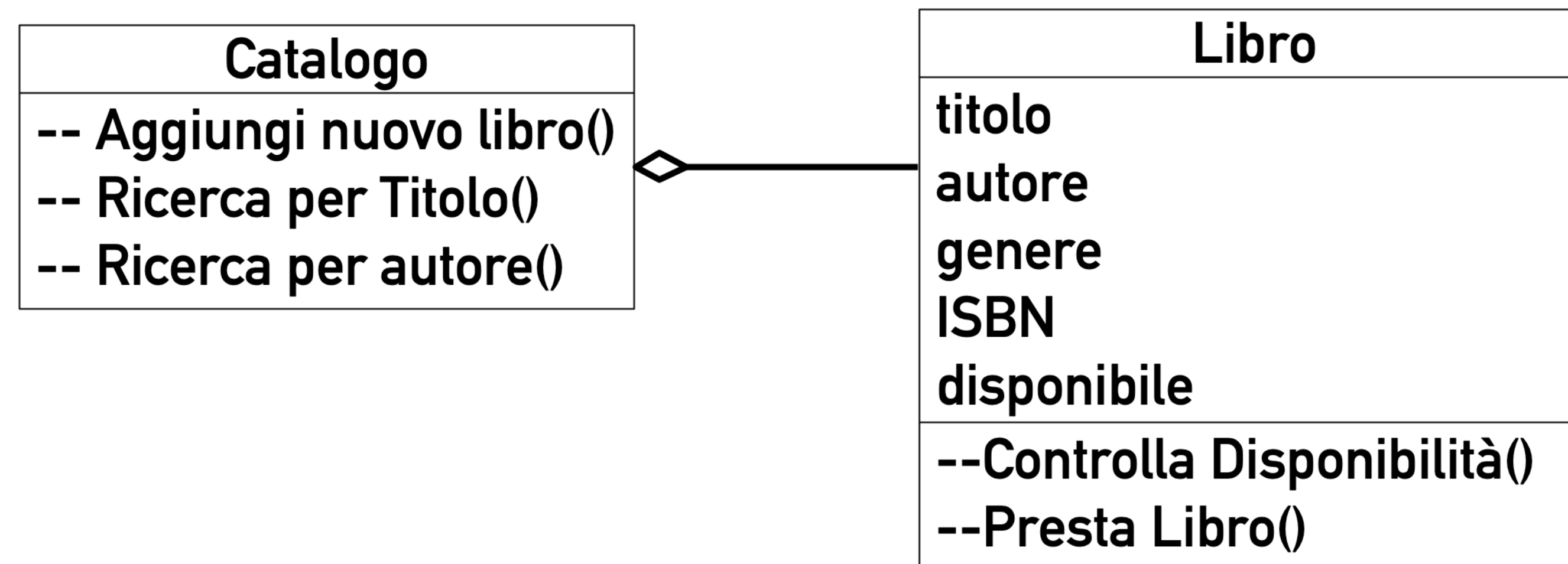
- La responsabilità di creare istanze di una classe non può essere attribuita alla classe stessa, ma ad una classe collegata ad essa
- Il sistema permette ai rappresentanti dell'editore di aggiungere un nuovo libro disponibile agli utenti nel catalogo



4. TROVA LE PRINCIPALI RESPONSABILITÀ DI OGNI CLASSE

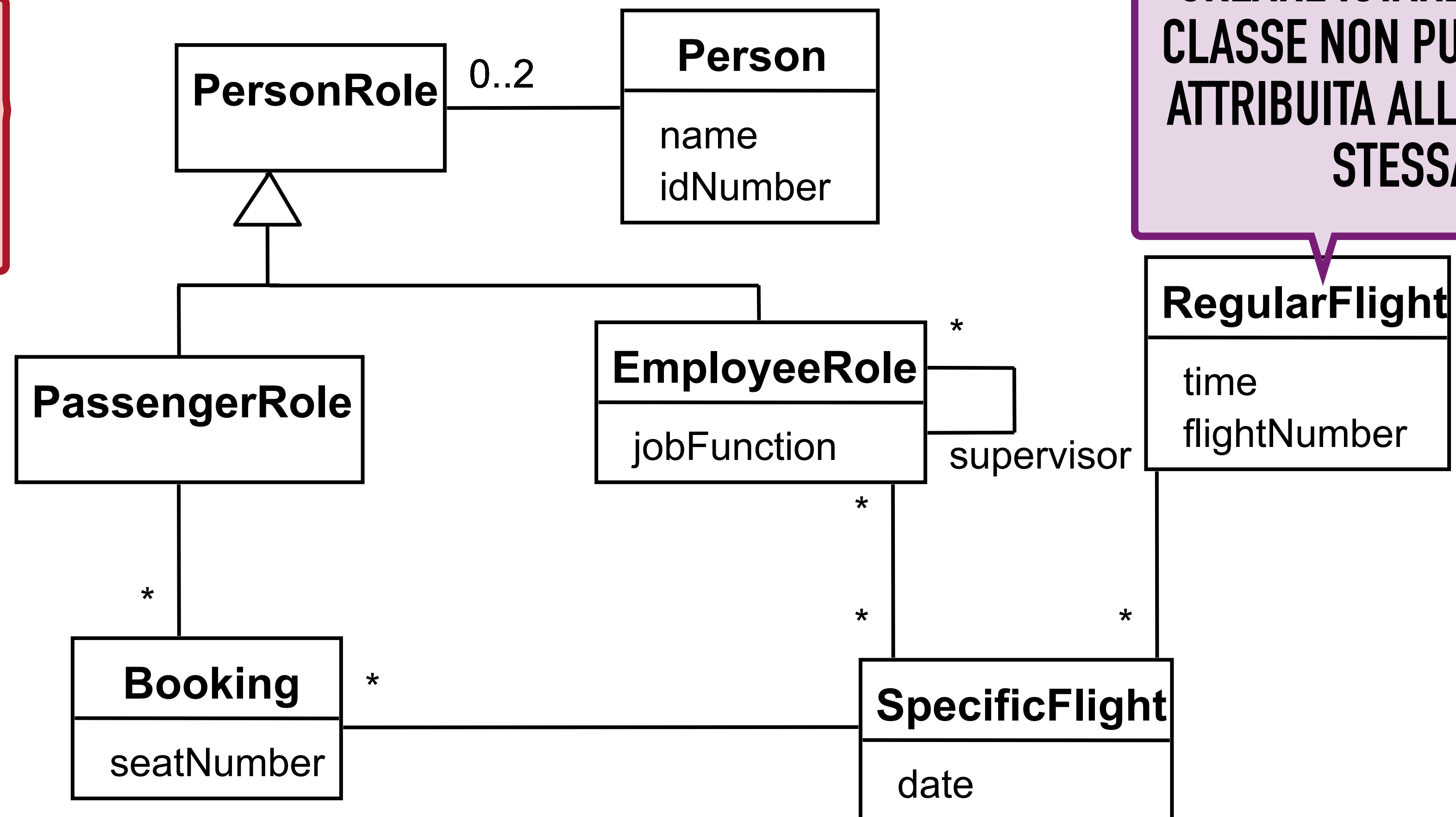


- La responsabilità di cercare istanze di una classe che fanno parte di una collection (es. lista, catalogo, etc.) non può essere attribuita alla classe stessa, ma alla classe collection
- L'utente può cercare libri nel catalogo per nome o per autore



ESEMPIO: COMPAGNIA AEREA CON RESPONSABILITÀ

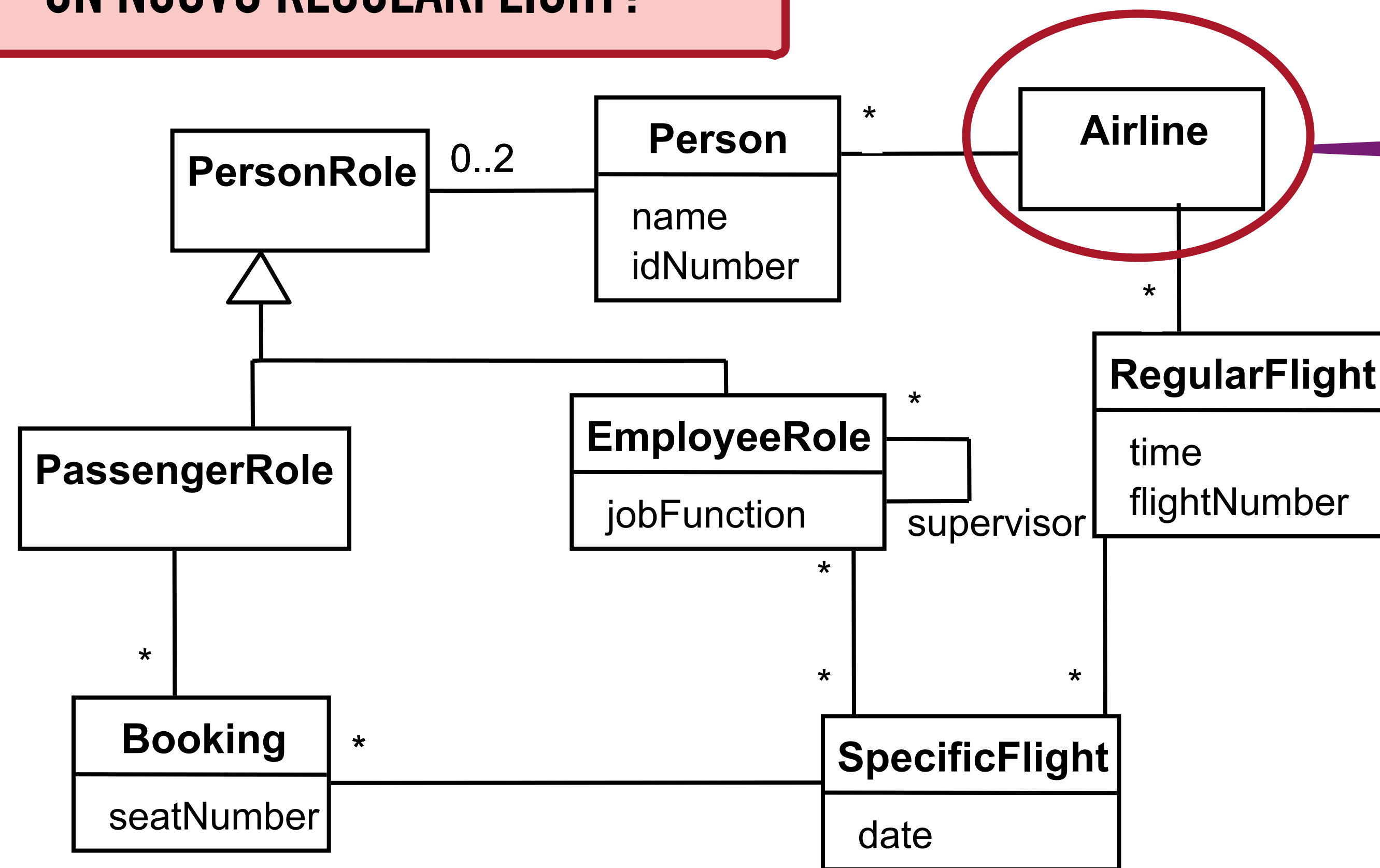
CHI HA LA
RESPONSABILITÀ DI
CREARE UN NUOVO
REGULARFLIGHT?



LA RESPONSABILITÀ DI
CREARE ISTANZE DI UNA
CLASSE NON PUÒ ESSERE
ATTRIBUITA ALLA CLASSE
STESSA

L'ESEMPIO DEI VOLI (CON RESPONSABILITÀ)

CHI HA LA RESPONSABILITÀ DI CREARE UN NUOVO REGULARFLIGHT?

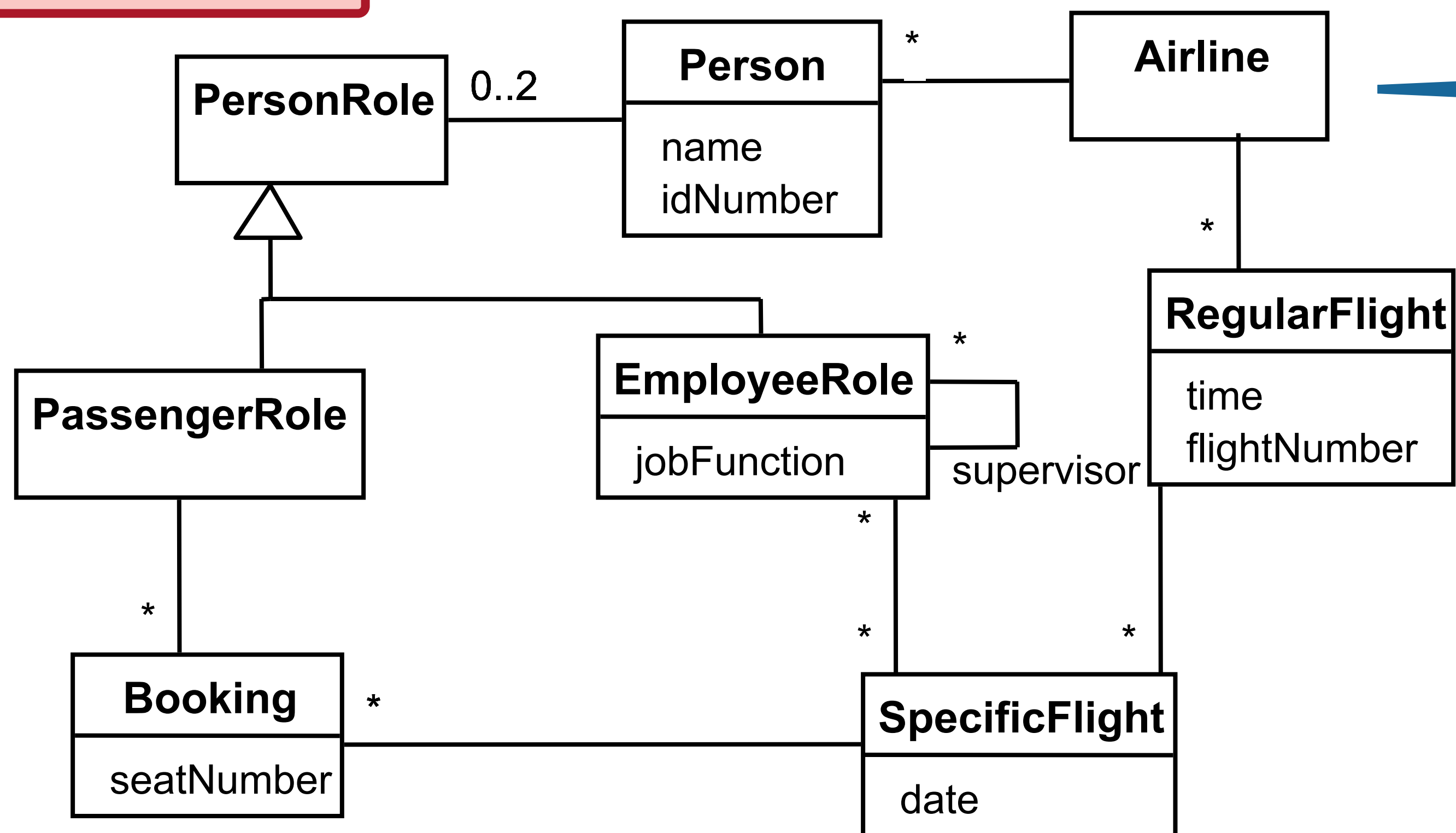


QUANDO UNA RESPONSABILITÀ NON PUÒ ESSERE ATTRIBUITA A NESSUNA DELLE CLASSI ESISTENTI, DOVREBBE ESSERE CREATA UNA NUOVA CLASSE

LA NUOVA CLASSE AIRLINE HA LA RESPONSABILITÀ DI CREARE UN NUOVO REGULARFLIGHT

ESEMPIO: COMPAGNIA AEREA CON RESPONSABILITÀ

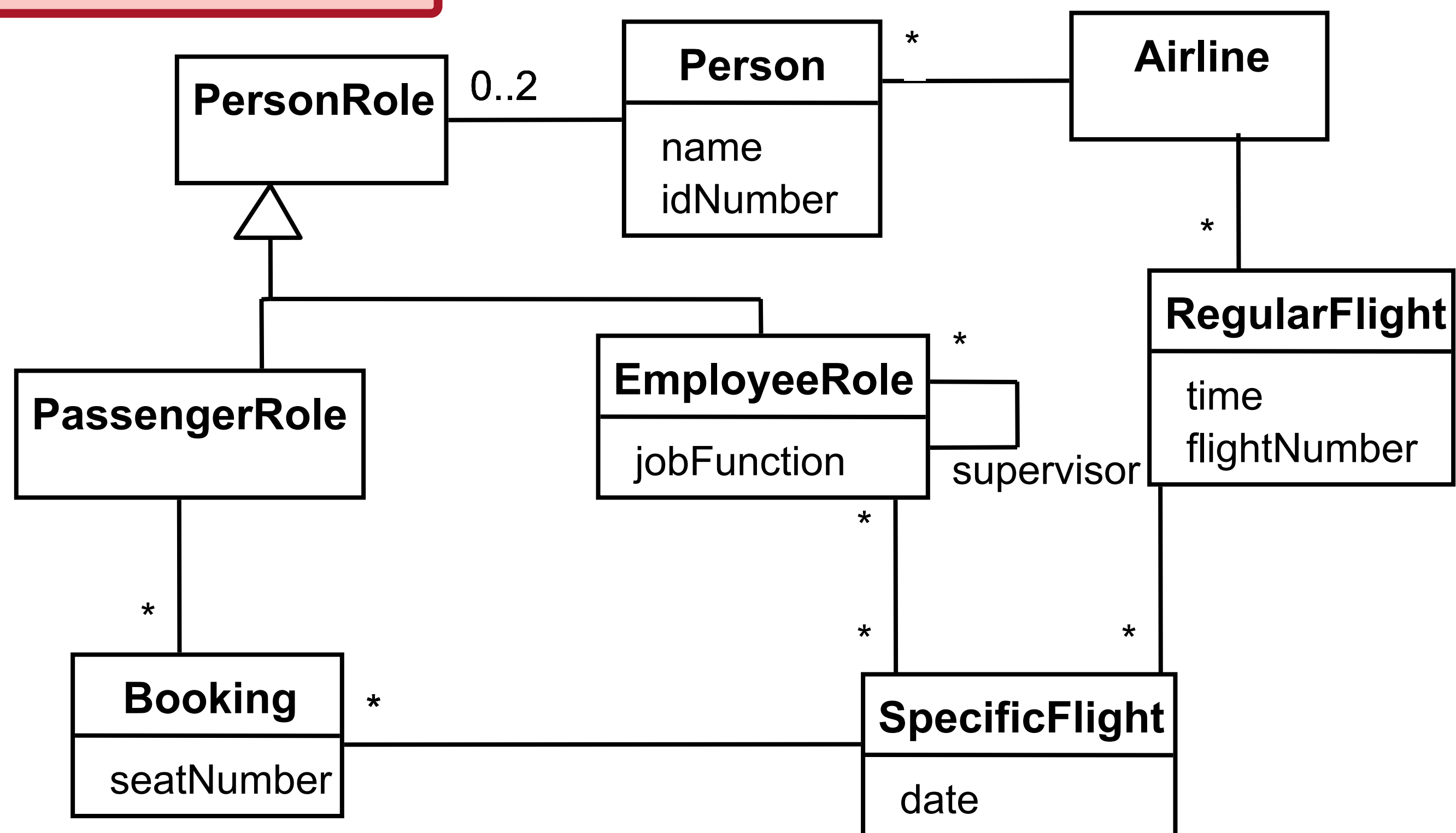
CHI HA LA
RESPONSABILITÀ DI
CERCARE UN
REGULARFLIGHT?



AIRLINE HA LA
RESPONSABILITÀ DI
CERCARE UN
REGULARFLIGHT PERCHÉ
HA LA VISIBILITÀ DI TUTTI I
REGULARFLIGHT

ESEMPIO: COMPAGNIA AEREA CON RESPONSABILITÀ

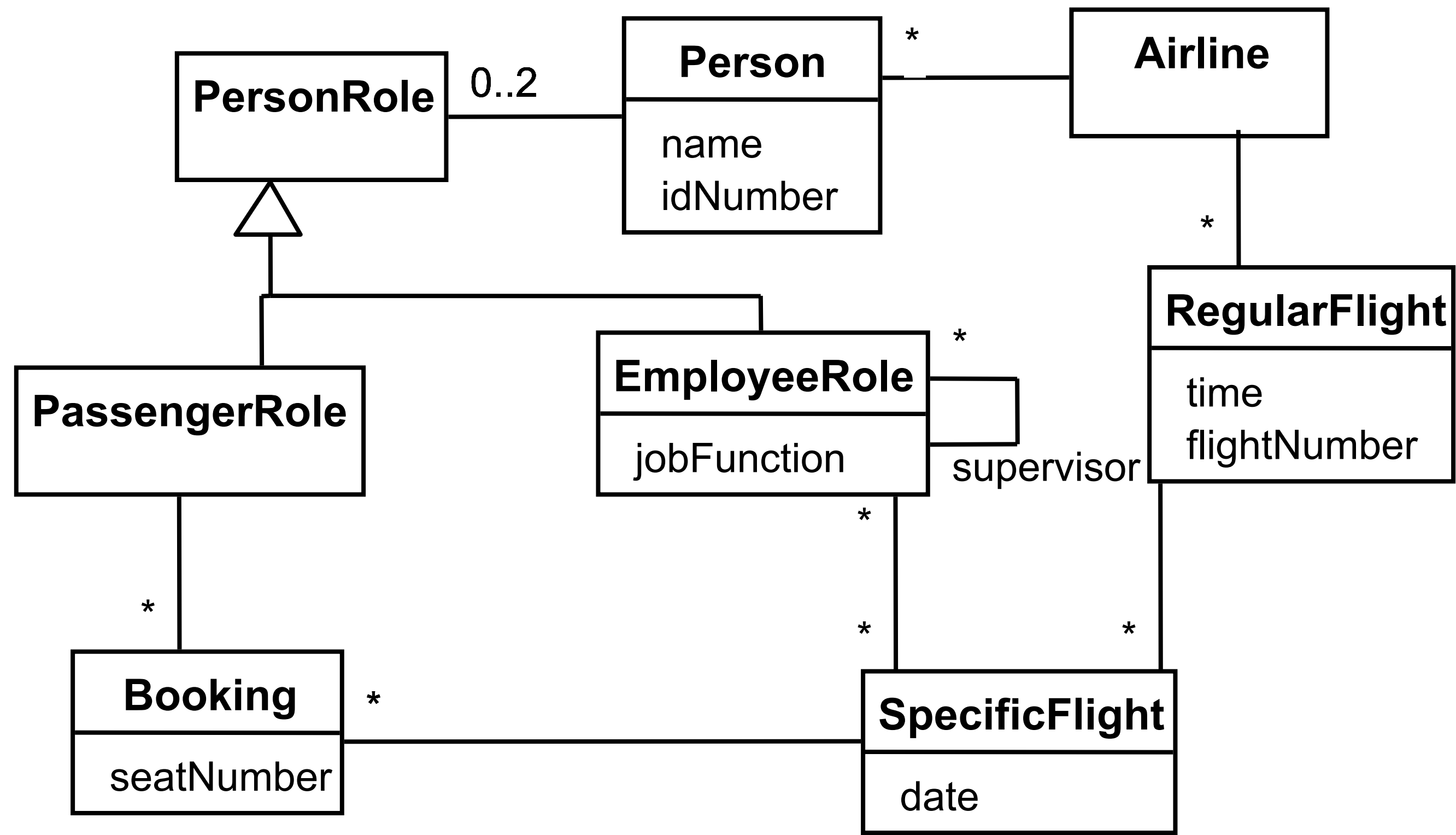
CHI HA LA RESPONSABILITÀ
DI MODIFICARE GLI
ATTRIBUTI DI UN
REGULARFLIGHT?



REGULARFLIGHT! POICHÉ
OGNI CLASSE É
RESPONSABILE DI
CAMBIARE I PROPRI
ATTRIBUTI

ESEMPIO: COMPAGNIA AEREA CON RESPONSABILITÀ

CHI HA LA RESPONSABILITÀ
DI CREARE UNO SPECIFIC
FLIGHT?

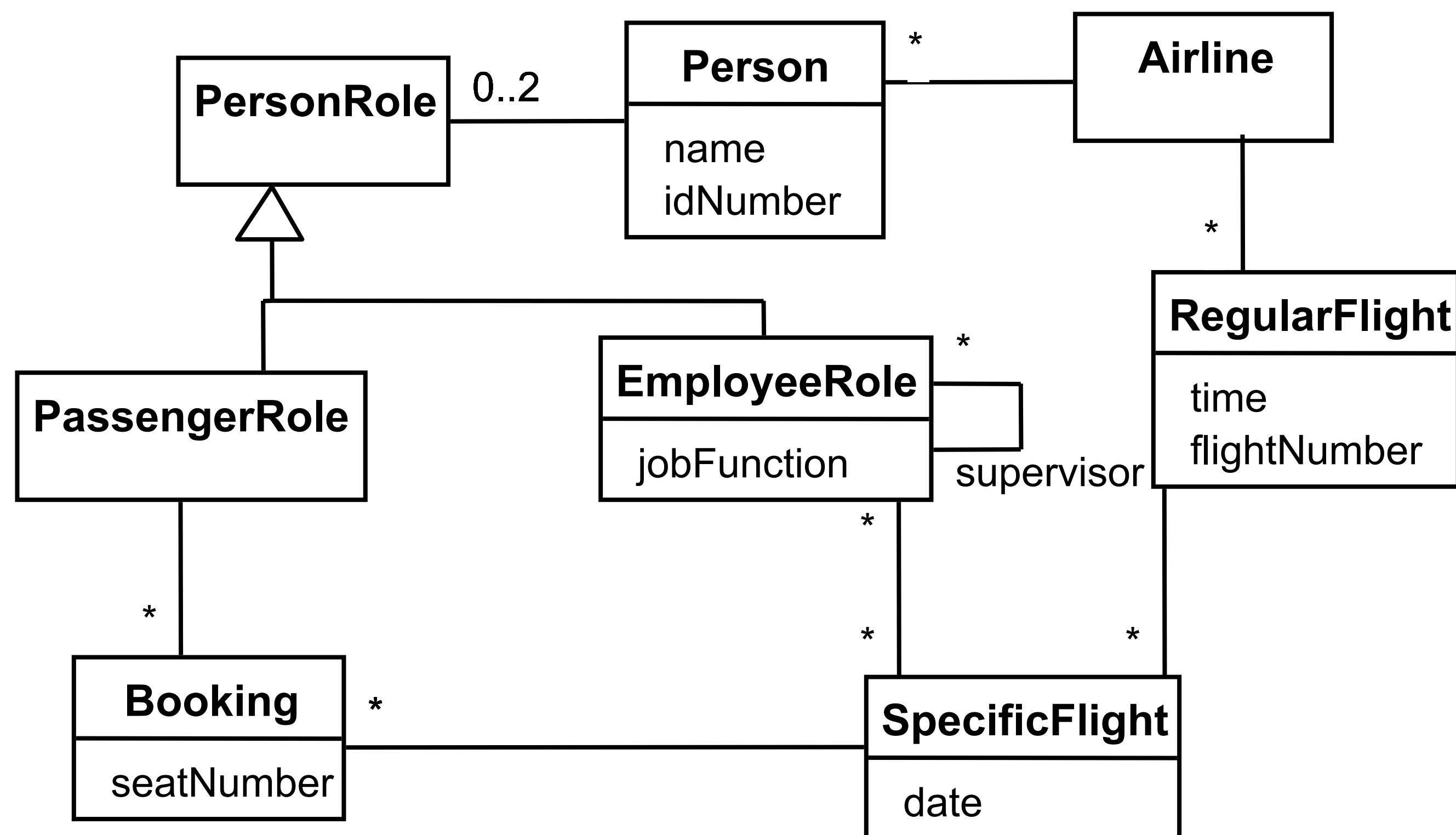


REGULARFLIGHT! POICHÉ IL
VOLO SPECIFICO É
UN'ISTANZA DEL VOLO
REGOLARE

ESEMPIO: COMPAGNIA AEREA CON RESPONSABILITÀ

PRENOTARE UN
PASSEGGERO SU UN VOLO
SPECIFICO (OSSIA CREARE
UN BOOKING)

SIA SPECIFIC FLIGHT CHE
PASSENGER ROLE
POTREBBERO CREARE UN
BOOKING. MEGLIO
PASSENGER ROLE POICHÉ É
IL PASSEGGERO A
DECIDERE DI PRENOTARE



CLASS RESPONSIBILITY COLLABORATION (CRC) CARDS



- Metodo alternativo per ottenere il diagramma delle classi:
 1. Per ogni classe identificata, porre il nome della classe su una scheda (Card)
 2. Man mano che vengono individuati attributi e responsabilità, elencarli sulle Card
 3. Sistemare le card su una lavagna per creare il Class diagram
 4. Disegnare le linee corrispondenti ad associazioni e generalizzazioni
- L'utilizzo di card (eventualmente fisiche) serve per forzare l'analista a non realizzare classi con un numero troppo elevato di attributi e metodi. Se la card è piena allora probabilmente bisogna dividere la classe in due o più classi più semplici

ESEMPIO DI CRC CARDS



Class Name	Collaborators <ul style="list-style-type: none">• Partner• Components
Responsibility <ul style="list-style-type: none">• Operations may go across several lines.	

Order		Class
Check if item is in stock	Order Line	Responsibility
Determine Price		Collaboration
Check for valid payment	Customer	
Dispatch to delivery address		

SYSTEM MODEL: IDENTIFICA LE OPERAZIONI



- Nel progetto di dettaglio bisogna passare dalle responsabilità alle operazioni, necessarie a realizzare le responsabilità di ciascuna classe
- Una responsabilità può essere realizzata attraverso molteplici operazioni
- Le operazioni che implementano una responsabilità sono normalmente dichiarate public
- Altri metodi che collaborano a realizzare una responsabilità devono essere dichiarate private se possibile



ESEMPIO DI PROCESSO DI ASTRAZIONE DEL SYSTEM DOMAIN MODEL

ESEMPIO: AZIENDA ALIMENTARE



- Un'azienda produttrice di prodotti alimentari, vuole organizzare un sistema informativo aziendale. Tutti gli utenti dell'applicazione devono essere in grado di cercare prodotti per nome nel catalogo. Inoltre i dipendenti devono essere in grado di accedere ad informazioni relative alle loro mansioni.
- I prodotti sono organizzati in linee di prodotto che accomunano prodotti dello stesso tipo: ad esempio, due linee possono essere pasta e sughi. I prodotti possono essere confezionati in diversi stabilimenti.
- I dipendenti si suddividono in diverse categorie: i manager, che sono responsabili di una o più linee di produzione (ogni linea é però gestita esattamente da tre manager, per assicurare una gestione equa), i supervisori della produzione che sono responsabili di tutti i prodotti di una specifica linea in uno specifico stabilimento, e gli operai che lavorano su uno specifico prodotto in un determinato stabilimento.
- L'applicazione deve prevedere un sistema di controllo accessi che consenta ai manager di accedere alle informazioni relative ai dipendenti di cui sono responsabili, e che consenta a tutti di accedere alle informazioni sui prodotti.

ESEMPIO: RICERCA CLASSI



- Un'azienda produttrice di **prodotti** alimentari, vuole organizzare un sistema informativo aziendale. Tutti gli utenti dell'applicazione devono essere in grado di cercare **prodotti** per nome nel **catalogo**. Inoltre i **dipendenti** devono essere in grado di accedere ad informazioni relative alle loro mansioni.
- I **prodotti** sono organizzati in **linee di produzione** che accomunano **prodotti** dello stesso tipo: ad esempio, due linee possono essere pasta e sughi. I **prodotti** possono essere confezionati in diversi **stabilimenti**.
- I **dipendenti** si suddividono in diverse categorie: i **manager**, che sono responsabili di una o più **linee di produzione** (ogni **linea** é però gestita esattamente da tre **manager**, per assicurare una gestione equa), i **supervisor** della produzione che sono responsabili di tutti i **prodotti** di una specifica **linea di produzione** in uno specifico **stabilimento**, e gli **operai** che lavorano su uno specifico **prodotto** in un determinato **stabilimento**.
- L'applicazione deve prevedere un sistema di controllo accessi che consenta ai **manager** di accedere alle informazioni relative alle **linee di produzione** di cui sono responsabili, e che consenta a tutti di accedere alle informazioni sui **prodotti**.

ESEMPIO: RICERCA CLASSI



- Un'azienda produttrice di **prodotti** alimentari, vuole organizzare un sistema informativo aziendale. Tutti gli utenti dell'applicazione devono essere in grado di cercare **prodotti** per nome nel **catalogo**. Inoltre i **dipendenti** devono essere in grado di accedere ad informazioni relative alle loro mansioni.

- Classi identificate:
 - Prodotto
 - Catalogo (contiene prodotti)
 - Dipendenti (poiché il sistema deve salvare le informazioni relative alle mansioni)

ESEMPIO: RICERCA CLASSI



- I **prodotti** sono organizzati in **linee di produzione** che accomunano **prodotti** dello stesso tipo: ad esempio, due linee possono essere pasta e sughi. I **prodotti** possono essere confezionati in diversi **stabilimenti**.
- Classi identificate:
 - Linea di produzione
 - Stabilimento
- Invece pasta e sughi non sono classi ma solo esempi

ESEMPIO: RICERCA CLASSI



- I **dipendenti** si suddividono in diverse categorie: i **manager**, che sono responsabili di una o più **linee di produzione** (ogni **linea** é però gestita esattamente da tre **manager**, per assicurare una gestione equa), i **supervisori** della produzione che sono responsabili di tutti i **prodotti** di una specifica **linea di produzione** in uno specifico **stabilimento**, e gli **operai** che lavorano su uno specifico **prodotto** in un determinato **stabilimento**.
- Classi identificate:
 - Manager
 - Operaio
 - Supervisore

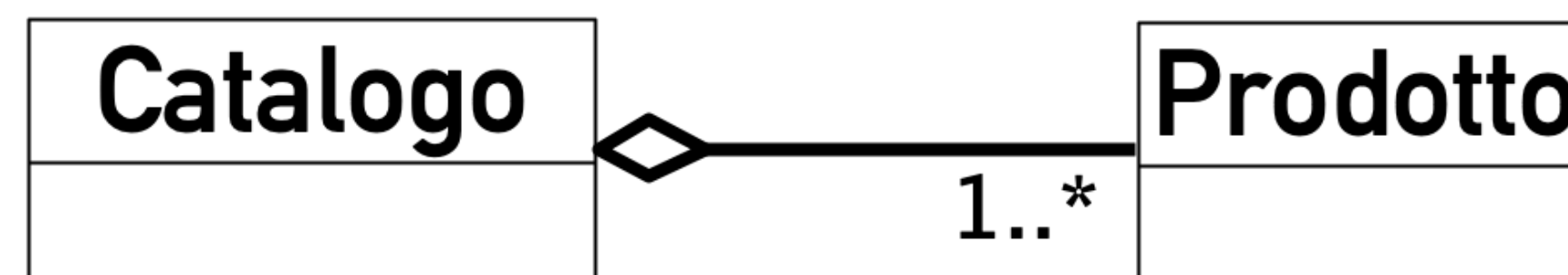


- L'applicazione deve prevedere un sistema di controllo accessi che consenta ai **manager** di accedere alle informazioni relative alle linee di produzione di cui sono responsabili, e che consenta a tutti di accedere alle informazioni sui **prodotti**.
- Responsabile non é una classe ma un ruolo del manager rispetto ai dipendenti
- Il sistema di controllo degli accessi é un termine usato in maniera impropria, si riferisce ad una funzionalità

ESEMPIO: RICERCA ASSOCIAZIONI



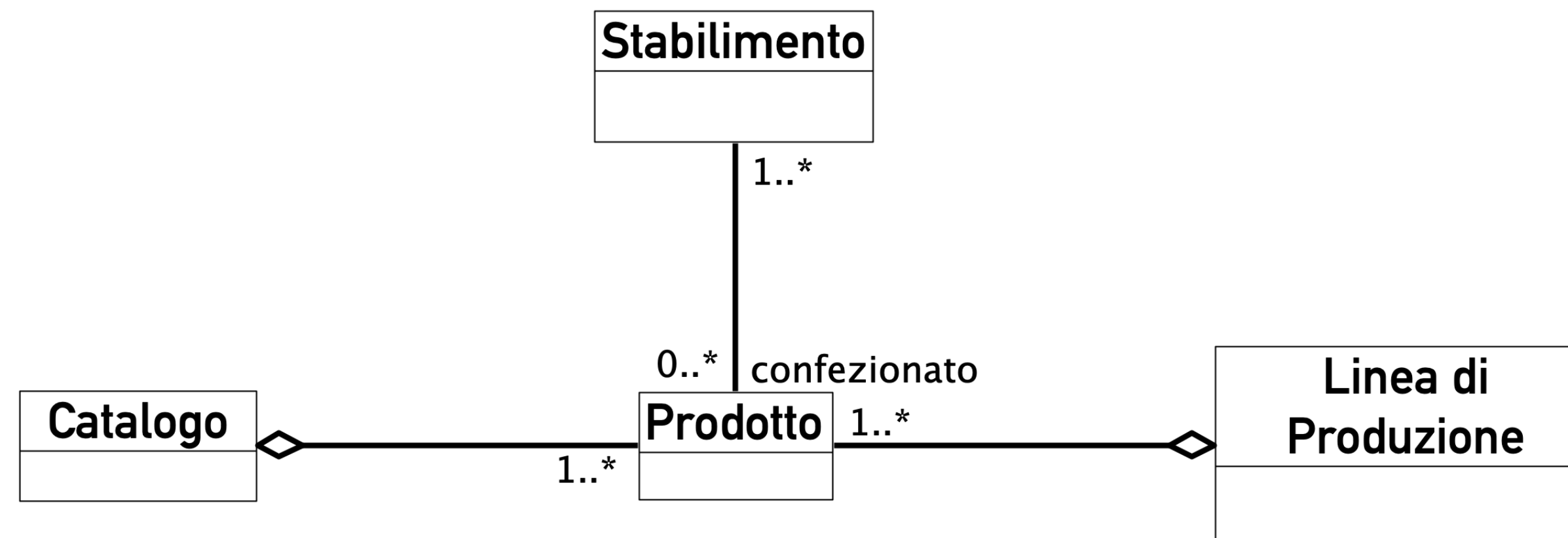
- Un'azienda produttrice di **prodotti** alimentari, vuole organizzare un sistema informativo aziendale. Tutti gli utenti dell'applicazione devono essere in grado di cercare **prodotti** per nome nel **catalogo**. Inoltre i **dipendenti** devono essere in grado di accedere ad informazioni relative alle loro mansioni.
- Il catalogo é una composizione di molteplici prodotti (aggregazione). Sarebbe meglio se ce ne fosse almeno uno altrimenti il sistema non avrebbe molto senso
- Per ora non rileviamo associazioni tra dipendenti e prodotti/catalogo



ESEMPIO: RICERCA ASSOCIAZIONI



- I **prodotti** sono organizzati in **linee di produzione** che accomunano **prodotti dello stesso tipo**: ad esempio, due linee possono essere pasta e sughi. I **prodotti** possono essere confezionati in diversi **stabilimenti**.
- Ciascun prodotto è organizzato in una linea di produzione, la linea di produzione organizza più prodotti dello stesso tipo
- Ciascun prodotto può essere confezionato in uno o più stabilimenti. Ciascuno stabilimento produce molteplici prodotti.
- Non emerge ancora la relazione tra linee di produzione e stabilimenti

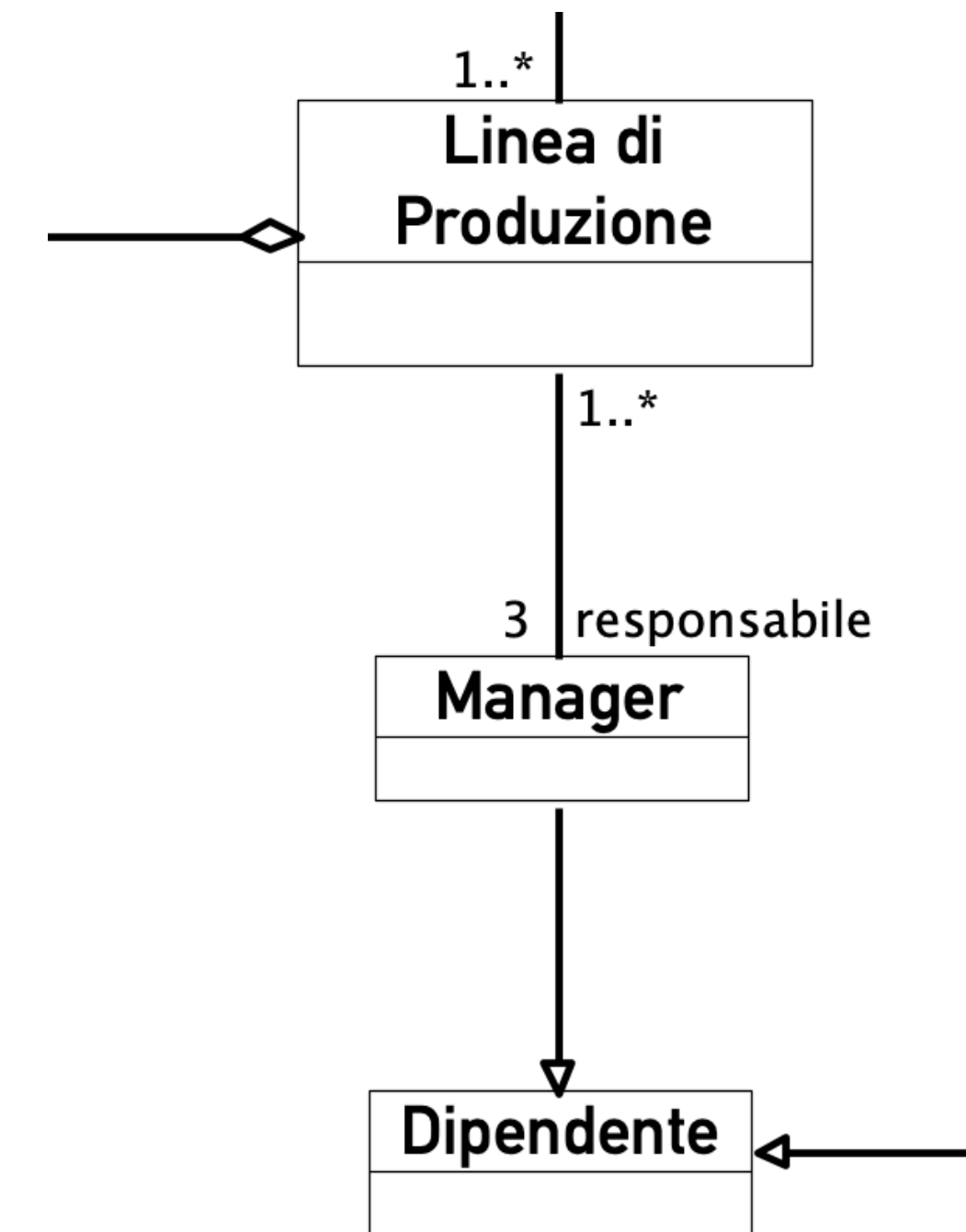


ESEMPIO: RICERCA ASSOCIAZIONI



- I **dipendenti** si suddividono in diverse categorie: i **manager**, che sono responsabili di una o più **linee di produzione** (ogni **linea** é però gestita esattamente da tre **manager**, per assicurare una gestione equa), i **supervisor** della produzione che sono responsabili di tutti i **prodotti** di una specifica **linea di produzione** in uno specifico **stabilimento**, e gli **operai** che lavorano su uno specifico **prodotto** in un determinato **stabilimento**.

- Il Manager è una sottoclasse di dipendente
- I manager assumono il ruolo di responsabili della linea di produzione
- Ciascun manager é responsabile di 1..* linee di produzione
- Ciascuna linea di produzione ha un numero esatto (3) di manager responsabili

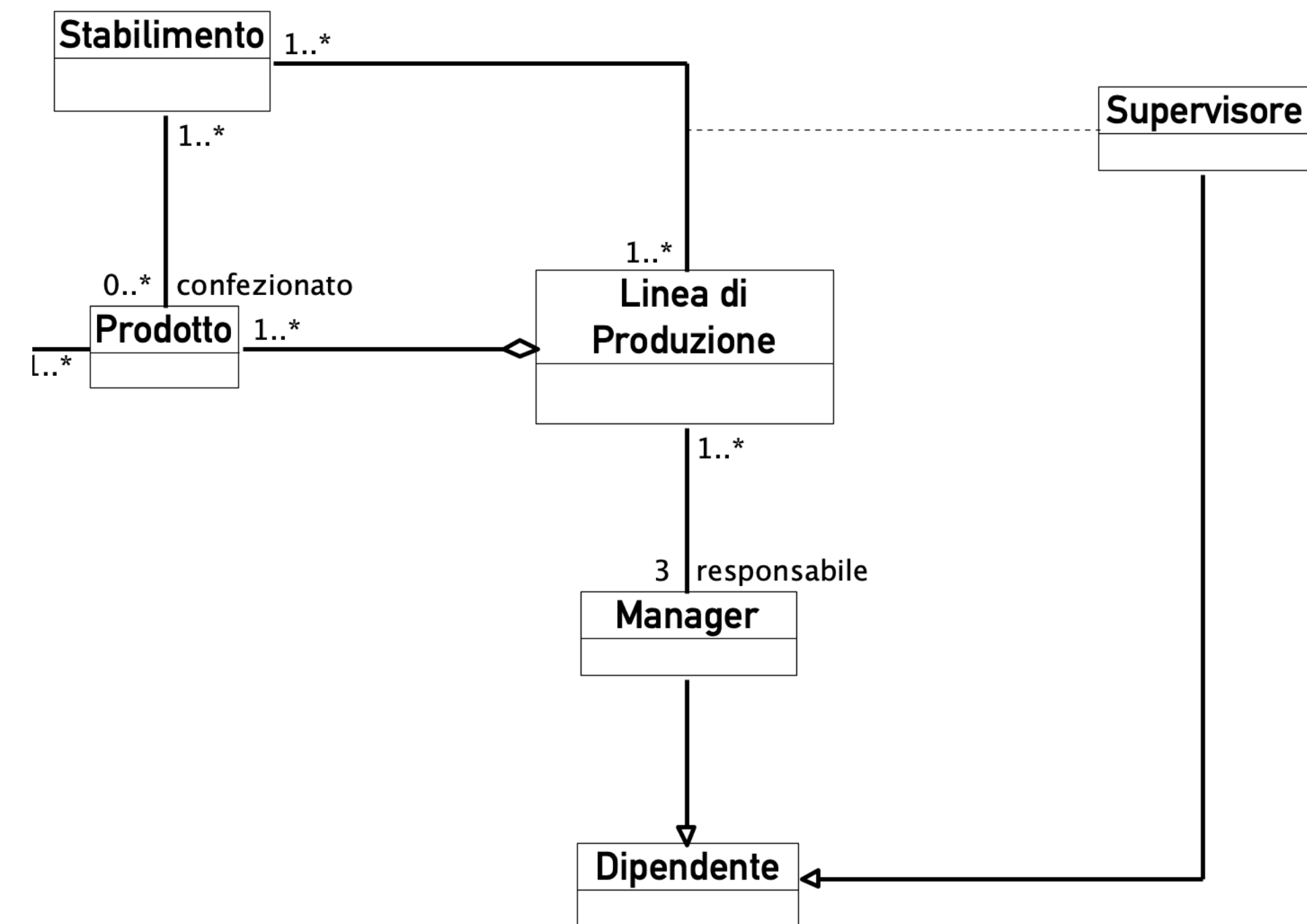


ESEMPIO: RICERCA ASSOCIAZIONI



- I **dipendenti** si suddividono in diverse categorie: i **manager**, che sono responsabili di una o più **linee di produzione** (ogni **linea** é però gestita esattamente da tre **manager**, per assicurare una gestione equa), i **supervisori** della produzione che sono responsabili di tutti i **prodotti** di una specifica **linea di produzione** in uno specifico **stabilimento**, e gli **operai** che lavorano su uno specifico **prodotto** in un determinato **stabilimento**.

- Il Supervisore è una sottoclasse di dipendente
- A una linea di produzione é associato un supervisore per uno specifico stabilimento: supervisore é una **classe associativa**

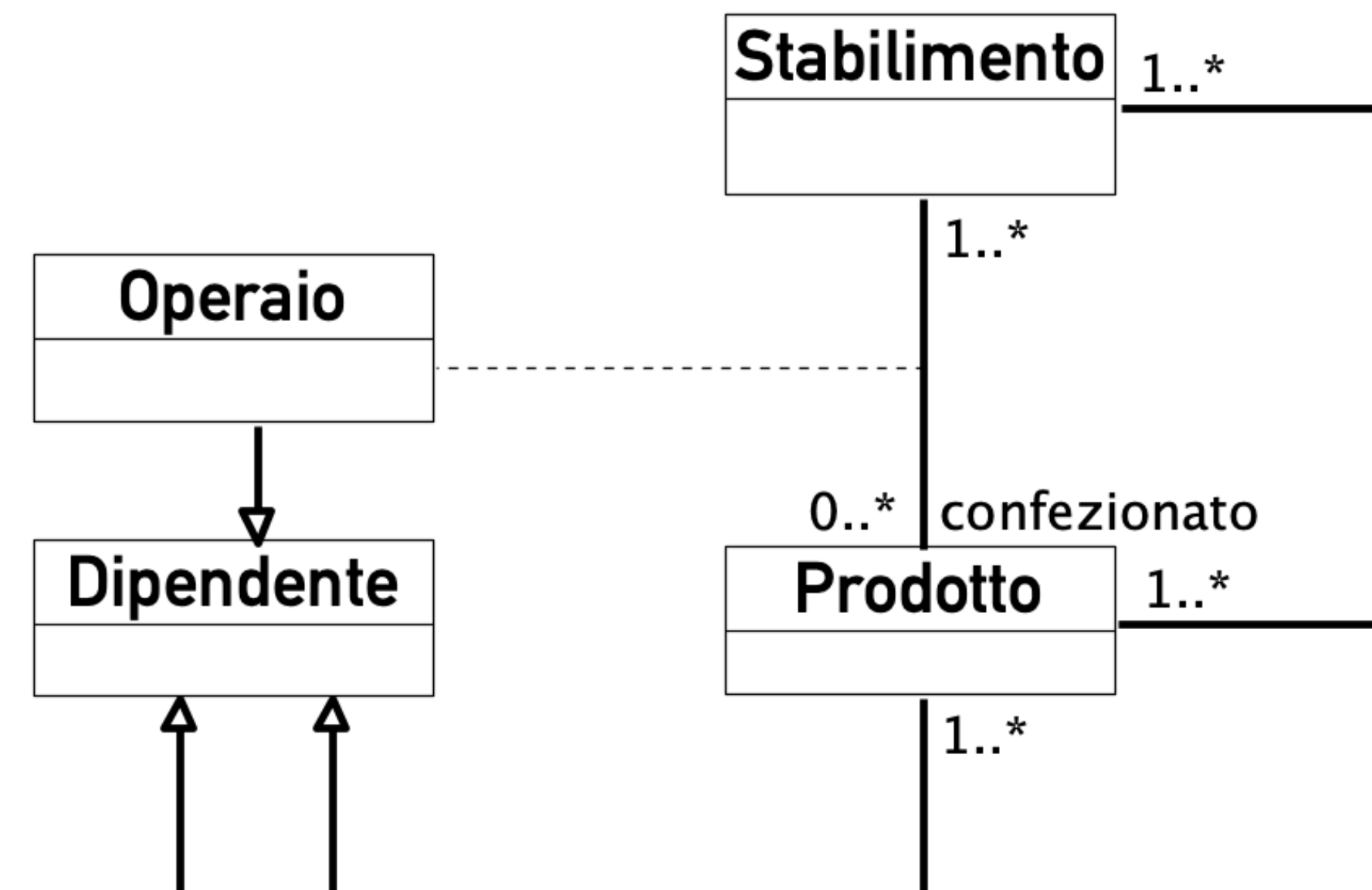


ESEMPIO: RICERCA ASSOCIAZIONI



- I **dipendenti** si suddividono in diverse categorie: i **manager**, che sono responsabili di una o più **linee di produzione** (ogni **linea** é però gestita esattamente da tre **manager**, per assicurare una gestione equa), i **supervisor** della produzione che sono responsabili di tutti i **prodotti** di una specifica **linea di produzione** in uno specifico **stabilimento**, e gli **operai** che lavorano su uno specifico **prodotto** in un determinato **stabilimento**.

- Operario è una sottoclasse di dipendente
- A uno stabilimento é associato un operaio per una specifica classe di prodotto: operaio é una **classe associativa**



ESEMPIO: ASSEGNAZIONE RESPONSABILITÀ



- Un'azienda produttrice di prodotti alimentari, vuole organizzare un sistema informativo aziendale. Tutti gli utenti dell'applicazione devono essere in grado di cercare prodotti per nome nel catalogo. Inoltre i dipendenti devono essere in grado di accedere ad informazioni relative alle loro mansioni.
- La ricerca dei prodotti é responsabilità del catalogo perché ha visibilità di tutti i prodotti
- La classe Dipendente é responsabile di accedere e visualizzare la mansione poiché tale responsabilità é ereditata da tutte le sue sottoclassi

ESEMPIO: RICERCA ASSOCIAZIONI



- L'applicazione deve prevedere un sistema di controllo accessi che consenta ai **manager** di accedere alle informazioni relative alle linee di produzione di cui sono responsabili, e che consenta a tutti di accedere alle informazioni sui **prodotti**.
- I manager possono accedere alle informazioni delle linee di cui sono responsabili, le linee non consentono a tutti di accedere alle proprie informazioni. Quindi la responsabilità é dei manager
- I prodotti permettono a tutti gli utenti di visualizzare le proprie informazioni. I prodotti hanno la responsabilità di esporre i propri dati

ESEMPIO: SYSTEM DOMAIN MODEL DI AZIENDA ALIMENTARE

