



POLITECNICO
MILANO 1863

Ripasso UML

26 Marzo 2019

Scaglione San Pietro

RESPONSABILI

Giovanni Meroni
Amarildo Likmeta

TUTOR

Marco Bacis
Valentina Deda

Introduzione UML

- Acronimo di Unified Modeling Language
- Permette di modellare graficamente la struttura ed il comportamento di un'applicazione
- È uno standard OMG
- È usato unicamente per documentare un progetto, nelle varie fasi del ciclo di sviluppo del software
- Non permette di generare direttamente codice eseguibile
- Comprende diversi tipi di diagramma
 - Per il progetto ci concentreremo unicamente sul class diagram

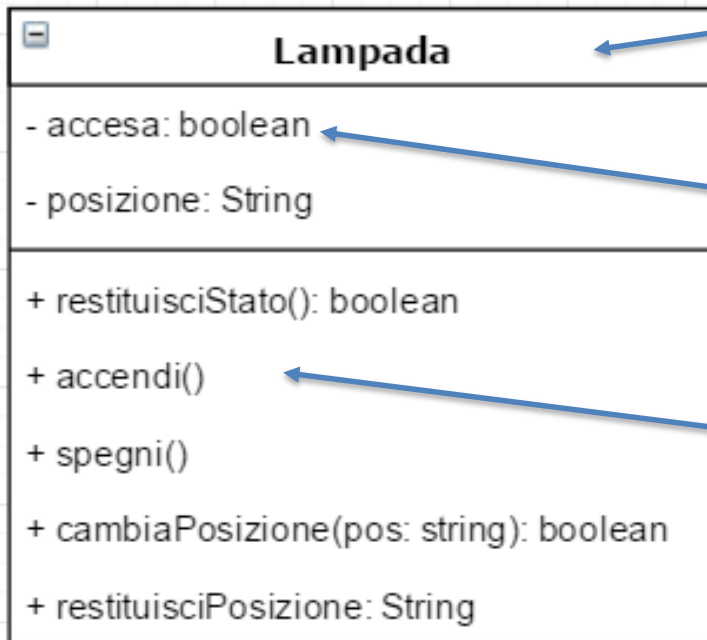
Modellazione classi in UML

- UML Class Diagram permettono di modellare graficamente:
 - La struttura delle classi
 - Le relazioni tra classi
- Data una classe
 - Gli attributi tengono traccia dello stato dell'oggetto
 - I metodi definiscono il comportamento dell'oggetto (l'invocazione di un metodo può alterare lo stato dell'oggetto)
 - Nella rappresentazione grafica di una classe, deve esserci una netta separazione tra attributi e metodi

Modellazione classi in UML

- A metodi e attributi è assegnato un modificatore di visibilità:
 - - Private: metodo/attributo accessibile solo all'interno della classe
 - # Protected: metodo/attributo accessibile anche dalle classi che ereditano da quella corrente
 - + Public: metodo/attributo accessibile anche dalle altre classi
- Per ogni attributo va indicato modificatore, cardinalità (se diversa da 1-1) e tipo di dato:
 - <modificatore> <nome attributo> [<cardinalità>]: <tipo di dato>
- Per ogni metodo vanno indicati modificatore, parametri di input e tipo di dato in output:
 - <modificatore> <nome metodo>(<nome parametro 1>: <tipo parametro 1>, <nome parametro 2>: <tipo parametro 2>, ...): <tipo output>

Modellazione classi in UML



Classe

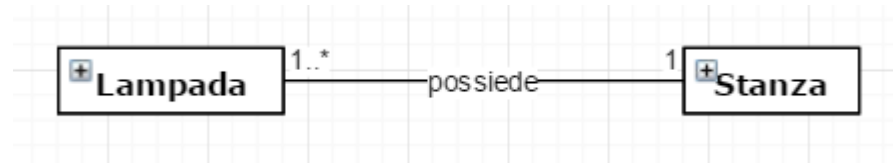
Attributi

Metodi

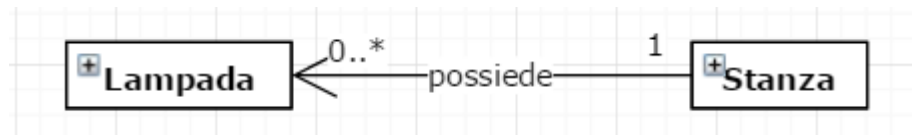
Associazioni tra classi in UML

- Se due classi devono poter interagire, è necessario introdurre una relazione
- La relazione è rappresentata con un arco
- La relazione si definisce con un verbo
- Ogni capo della relazione ha una cardinalità minima ed una massima
- * indica un numero massimo arbitrario di elementi
- La cardinalità si riferisce alla classe ad essa più vicina
- Se si assegna un verso, la relazione è navigabile solo dalla classe uscente verso la classe entrante (quella “puntata” dal verso)

Associazioni tra classi in UML



- Ogni stanza ha una o più lampade (lampadari)
- Ogni lampada risiede in una sola stanza
- Data una stanza si può risalire alle lampade presenti
- Data una lampada si può risalire alla stanza che la contiene

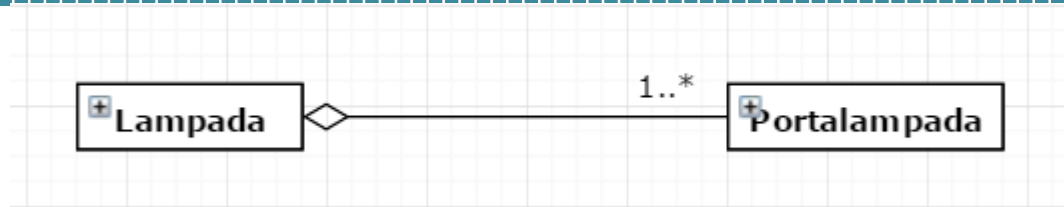


- Ogni stanza ha zero o più lampade
- Ogni lampada risiede in una sola stanza
- Data una stanza si può risalire alle lampade presenti
- Data una lampada **non** si può risalire alla stanza che la contiene

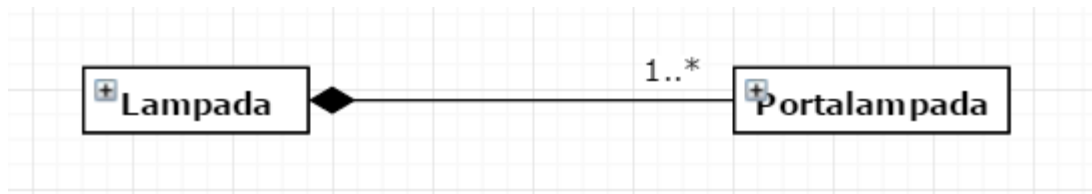
Associazioni tra classi in UML

- Se una classe è concettualmente parte di un'altra, è necessario introdurre una relazione di aggregazione
- L'aggregazione è rappresentata con un arco terminante in un rombo vuoto
- Il rombo deve essere collegato alla classe contenitore
- La cardinalità della classe contenitore è sempre 1
- Non è possibile creare cicli di aggregazioni
- Se la classe parte non può esistere senza la classe contenitore, la relazione è di composizione
- La composizione è rappresentata con un arco terminante in un rombo pieno
- Per il resto, segue le stesse regole della relazione di aggregazione

Associazioni tra classi in UML



- Ogni lampada ha almeno un portalampada
- Il portalampada è parte della lampada
- Il portalampada può esistere senza lampada (p.e., come ricambio)

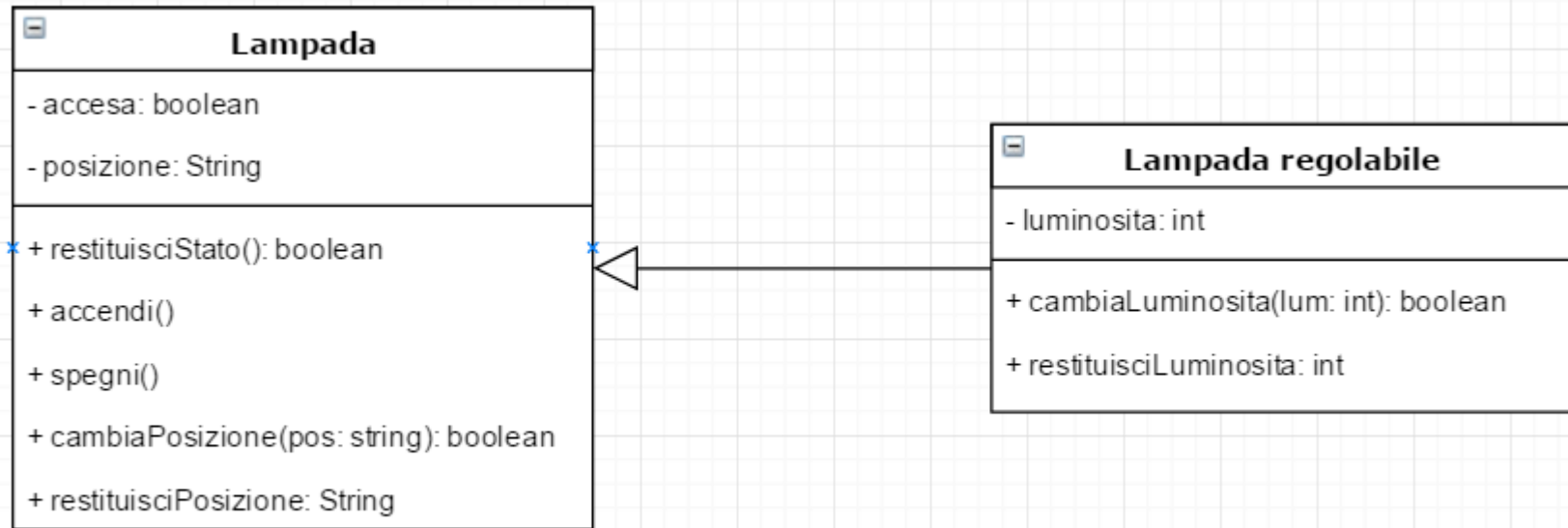


- Ogni lampada ha almeno un portalampada
- Il portalampada è parte della lampada
- Il portalampada **non** può esistere senza lampada

Ereditarietà in UML

- Per esprimere la relazione padre-figlio tra due classi si utilizza la relazione di generalizzazione
- La generalizzazione è rappresentata con un arco terminante in un triangolo vuoto
- Il triangolo deve essere collegato alla classe padre

Ereditarietà in UML

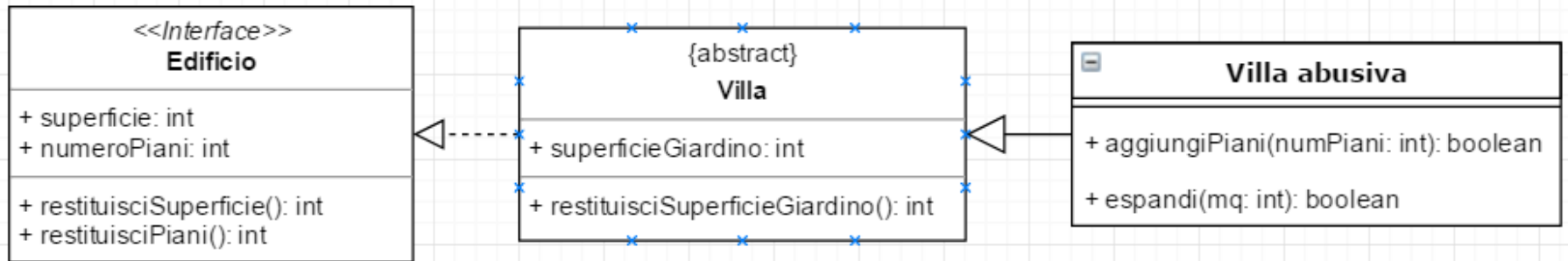


- Lampada regolabile eredita tutti gli attributi e i metodi di Lampada

Ereditarietà in UML

- Le classi astratte si modellano come classi normali, anteponendo lo stereotipo {abstract} al nome della classe
- Le interfacce si modellano come classi normali, anteponendo lo stereotipo <<interface>> al nome della classe
- Per indicare che una classe realizza un'interfaccia, la si collega a quest'ultima mediante la relazione di implementazione
- La relazione di implementazione si rappresenta come un arco tratteggiato terminante in un triangolo
- Il triangolo deve essere collegato all'interfaccia

Ereditarietà in UML



- La classe astratta Villa realizza l'interfaccia Edificio
- La classe Villa abusiva estende la classe astratta Villa

UML: consigli pratici

- Cominciare a modellare le strutture dati
- Considerare tutte le informazioni che è necessario memorizzare per mantenere lo stato dell'applicazione
- Ragionare sulle relazioni tra tali informazioni:
 - Informazioni direttamente correlate andranno inserite nella stessa classe
 - Informazioni scorrelate andranno inserite in classi diverse
 - Informazioni simili possono essere definite come figli di una classe padre, contenente gli aspetti comuni
- Prestare attenzione alle cardinalità e al verso delle relazioni
 - Non lasciare le cardinalità non specificate

UML: consigli pratici

- È buona prassi definire gli attributi di una classe privati o protected, utilizzando metodi pubblici per leggerli o cambiarne il valore
- Per definire attributi settabili solo una volta, modificare il costruttore della classe che li contiene
- Se non devono esistere attributi non inizializzati, modificare il costruttore della classe che li contiene, oppure definire un valore di default.
- Fare uso dei design pattern
- Se si definisce una classe per un elemento del gioco (plancia, sacchetto dei dadi,...), utilizzare il nome inglese (WindowFrame, DiceBag)
- Non trattare le classi come semplici strutture dati
 - Ogni classe dovrebbe includere sia attributi, sia la logica per la gestione di questi ultimi