

Classi e Oggetti

Ingegneria del Software

Gli oggetti: concetti fondamentali

- ~ Ovunque guardiamo vediamo oggetti: persone, animali, piante, automobili, computer, etc.
- ~ Le persone pensano in termini di oggetti (o di categorie di oggetti)
- ~ I programmi per computer (ed i programmi Java) possono essere visti come oggetti, costituiti a loro volta da altri oggetti software interconnessi.
- ~ Gli oggetti (nel mondo reale) si dividono in due categorie: animati ed inanimati.

Gli oggetti: concetti fondamentali

- ~ Gli oggetti animati sono vivi, si muovono, quelli inanimati sembrano non far nulla -> tutti hanno qualcosa in comune.
- ~ Possiedono **attributi** (come dimensione, forma, peso, colore) e tutti mostrano un **comportamento** (es. la palla rimbalza, il bambino piange, la radio suona...).
- ~ Al livello software è quindi importante studiare i tipi di attributi e i comportamenti che gli oggetti software mostrano
- ~ Gli uomini imparano a conoscere gli oggetti studiando/osservando i loro attributi ed il loro comportamento.

Gli oggetti: concetti fondamentali

~ Gli oggetti del mondo reale condividono due caratteristiche, essi hanno:

- § • uno stato
- § • un comportamento
- § • un cane ad esempio:
 - § • Lo stato è identificato dai valori dati a: nome, colore, razza, ...
 - § • Il comportamento potrebbe essere uno dei seguenti: abbaia, scodinzola, odora, ...
- § • Per una bicicletta:
 - § • Stato, valori dati a: marcia ingranata, cadenza della pedalata, velocità attuale
 - § • Comportamento: cambia marcia, cambia cadenza di pedalata, frena

Alcune di queste slide sono ispirate dal tutorial sun: <http://java.sun.com/docs/books/tutorial/java/index.html>

Gli oggetti: concetti fondamentali

- ~ In altre parole in un ciascun oggetto che ci circonda possiamo trovare:
- §• Degli attributi a cui assegnare dei valori (che determinano lo stato)
- §• Dei comportamenti, descritti da metodi

Cosa è un oggetto

- ~ Gli oggetti software sono concettualmente simili agli oggetti del mondo reale: anch'essi hanno uno stato e un comportamento.
- ~ Un oggetto memorizza il suo stato in variabili dette anche **attributi (attribute)** o campi (field)
- ~ Il comportamento viene esposto tramite funzioni o **metodi (method)**.
- ~ I metodi agiscono sullo stato dell'oggetto a cui appartengono e sono il meccanismo di comunicazione principale tra oggetti (**invocazione di metodi**)
- ~ Esempio: automobile
 - §• **Attributi** (che descrivono lo stato): motore_acceso, velocità, luci_accese
 - §• **Metodi**: accendi, accelera, frena, accendiLuci

Cosa è un oggetto

- ~ Perché non mettere tutto il codice in un unico contenitore (oggetto, funzione, unità, modulo, ...)?
- ~ **Modularità**: il codice di un oggetto può essere scritto e mantenuto indipendentemente dal resto del codice
- ~ **Information-hiding**: limitando l'interazione con i soli metodi di un oggetto i dettagli della sua struttura interna sono nascosti al mondo esterno
- ~ **Riuso**: se un oggetto esiste già, può essere riusato in altre parti del programma o in un altro programma.
- ~ **Componibilità e debug**: se un oggetto crea problemi può essere rimosso dalla applicazione e al suo posto se ne può mettere un altro che svolge le stesse funzioni (anche nel mondo reale funziona così: se il motore di un'auto si rompe, si cambia il solo motore non l'intera auto)

Cosa è una classe

- ~ Nel mondo reale si trovano spesso parecchi oggetti dello stesso tipo.
- ~ Per esempio esistono molte biciclette, tutte dello produttore e modello.
 - Ogni bicicletta contiene gli stessi componenti e nasce dallo stesso progetto.
- ~ In termini object-oriented si dice che una certa bicicletta (la mia ad esempio) è una istanza della classe (*instance of class*) di oggetti di nome Bicicletta.
- ~ La classe Bicicletta è il progetto da cui la mia bicicletta è stata creata.



Esempio di codice JAVA

```
class Bicycle {  
    int cadenza = 0;  
    int velocita = 0;  
    int marcia = 1;  
    void cambiaCadenza (int nuovoValore) {  
        cadenza = nuovoValore;  
    }  
    void cambiaMarcia(int nuovoValore) {  
        marcia = nuovoValore;  
    }  
    void accelera (int incremento) {  
        velocita = velocita + incremento;  
    }  
}
```

OOP – concetti fondamentali

- ~ Oggetti diversi possono avere attributi simili e mostrare comportamenti simili.
- ~ La **progettazione** orientata agli oggetti **modella** il software in termini simili a quelli che gli uomini usano per descrivere gli oggetti del mondo reale.
- §• Viene sfruttato il concetto di **classe** -> gli oggetti di una stessa classe hanno le stesse caratteristiche.
- §• Il concetto di **ereditarietà** (anche multipla) con cui è possibile derivare nuove classi “assorbendo” le caratteristiche di classi già esistenti ed aggiungendone di nuove
- ~ Es: un oggetto della classe ‘spider’ ha le caratteristiche della classe più generale ‘automobile’ ma in più ha il tettuccio apribile

OOP – concetti fondamentali

- ~ La progettazione orientata agli oggetti modella i componenti software proprio come vengono descritti gli oggetti del mondo reale, usando i loro attributi ed il loro comportamento.
- ~ Gli oggetti nel mondo reale interagiscono tra di loro comunicando -> La OOP modella anche la comunicazione tra gli oggetti tramite **messaggi**.
- ~ La OOP **incapsula** gli attributi e le funzionalità degli oggetti -> gli oggetti hanno la proprietà di nascondere le informazioni:
 - §• sebbene essi possano comunicare tra loro attraverso specifiche **interfacce**, non possono sapere come altri oggetti sono implementati.
- ~ I dettagli dell'implementazione sono nascosti all'interno degli oggetti stessi.

OOP – concetti fondamentali

- ~ Nascondere le informazioni (Information Hiding) è cruciale nell'ambito della progettazione software.
- ~ Nella programmazione procedurale (linguaggi tipo C, Pascal) la programmazione tende ad essere orientata all'azione- > l'unità di programmazione è la **funzione**, gruppi di azioni che svolgono qualche compito comune vengono trasformate in funzioni e queste ultime a loro volta raggruppate a formare programmi.
- ~ In Java l'unità di programmazione è la **classe** da cui gli **oggetti** vengono **istanziati** (creati); le classi Java contengono **metodi** (che implementano delle **funzionalità**) e **campi** (che implementano **attributi**).



OOP – concetti fondamentali

- ~ Creare un programma in Java significa creare delle classi, ogni classe contiene campi ed un insieme di metodi per manipolare questi campi per fornire determinati servizi ai clienti.
- ~ Le classi vengono usate come mattoni per costruire nuove classi.
- ~ Le classi hanno anche relazioni con altre classi chiamate associazioni.
- ~ Un software ad oggetti viene scritto come aggregati di classi, ogni 'aggregato' può essere riusato per creare altri programmi.
- ~ Parola d'ordine RIUSO.



...passiamo ora ad
UML

Cosa sono gli oggetti?

- ~ Un oggetto è (UML Reference Manual - Rumbaugh): un'entità discreta, con confini ben definiti che incapsula stato e comportamento
- ~ Ogni oggetto è istanza della classe che definisce l'insieme delle caratteristiche comuni (attributi e operazioni) condivise da tutte le sue istanze
- ~ Un oggetto ha:
 - §. attributi - rappresentano la parte dati
 - §. Operazioni - rappresentano la parte comportamento

Cosa sono gli oggetti?

- ~ un oggetto è un insieme coeso di dati e funzioni
- ~ Per avere accesso alla parte dati di un oggetto si deve chiamare una delle funzioni rese disponibili dall'oggetto stesso
 - §• Nell'analisi queste funzioni si chiamano *operazioni*
- ~ Mascherare i dati di un oggetto sotto uno strato di funzioni si chiama *incapsulazione*.
- ~ L'incapsulazione non è imposta da UML
 - §• Alcuni linguaggi di programmazione non la richiedono
 - §• Comunque buona pratica di programmazione

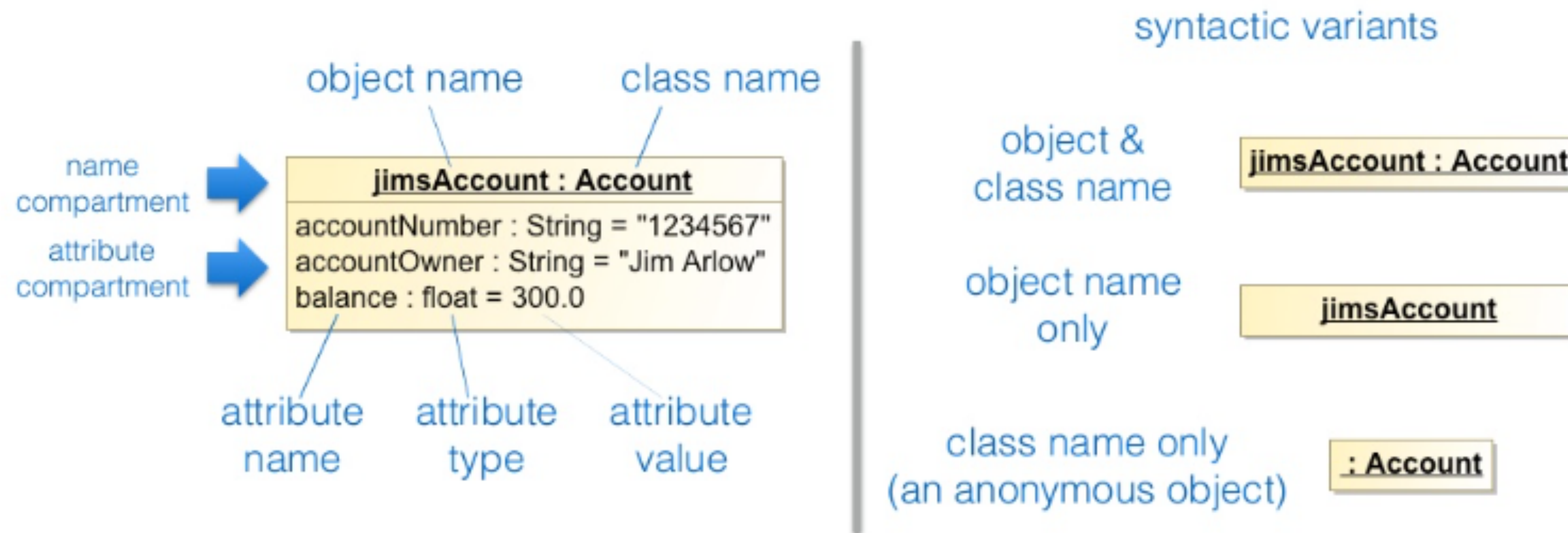
Cosa sono gli oggetti?

~ Tutti gli oggetti hanno:

- §. Identità: ogni oggetto ha la sua identità unica può essere acceduto mediante un riferimento all'oggetto
- §. Stato: un insieme significativo di valori degli attributi e delle relazioni dell'oggetto in un dato istante
- §. Comportamento: l'insieme delle operazioni che un oggetto può fare o i servizi che l'oggetto offre agli altri oggetti

Cosa sono gli oggetti?

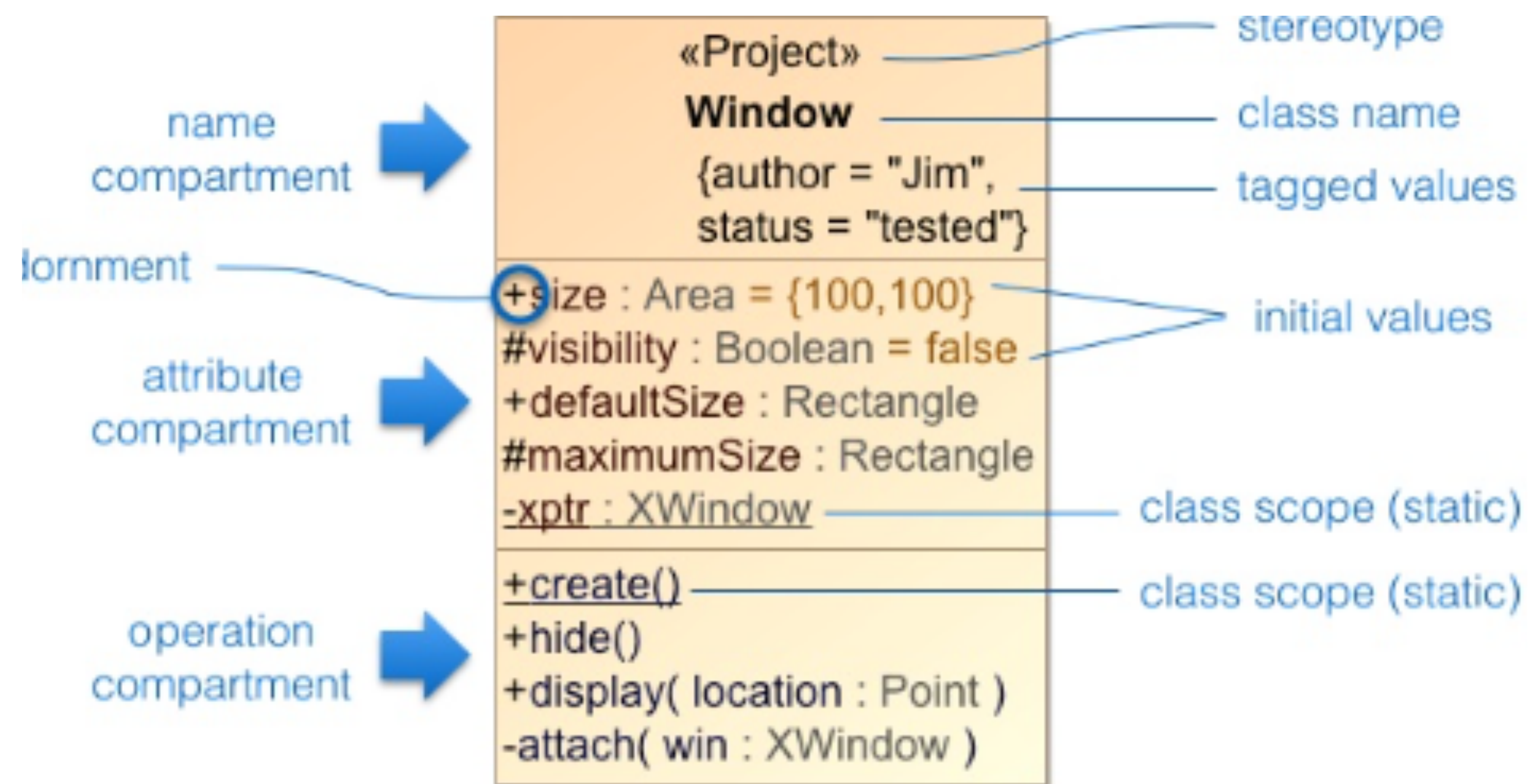
- ~ Gli oggetti interagiscono fra di loro per generare il comportamento del sistema
- ~ L'interazione comporta che gli oggetti si scambino messaggi
 - §• Un messaggio viene eseguito con il metodo corrispondente
 - §• Può produrre una transizione di stato



Sintassi UML per gli oggetti

Cosa sono le Classi

- ~ una classe definisce le caratteristiche (attributi, operazioni, metodi, relazioni e comportamento) di un insieme di oggetti.
- §• Ogni oggetto è un'istanza di una, ed una sola, classe
- §• Oggetti diversi appartenenti alla stessa classe hanno lo stesso insieme di attributi, ma possono avere valori differenti per questi attributi
- ~ Esistono molte classificazioni possibili per descrivere il mondo reale
- §• Trovare quella giusta è una delle chiavi per produrre una valida analisi OO



Sintassi UML per le classi