

DIAGRAMMI DELLE CLASSI E DEGLI OGGETTI INGEGNERIA DEL SOFTWARE

Università degli Studi di Padova
Facoltà di Scienze MM. FF. NN.

Corso di Laurea in Informatica, A.A. 2011 – 2012

rcardin@math.unipd.it

SOMMARIO

- Introduzione
- Proprietà e Operazioni
- Concetti base e avanzati
- Diagrammi degli oggetti

2

Ingegneria del software mod. A

Riccardo Cardin

SOMMARIO

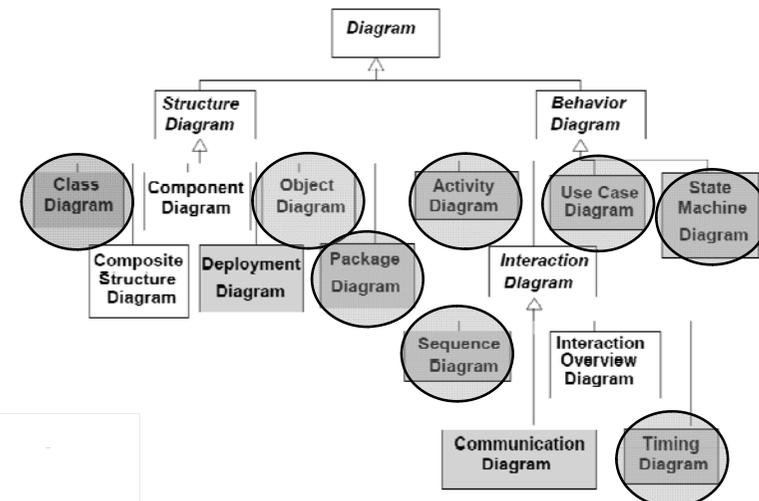
- Introduzione
- Proprietà e Operazioni
- Concetti base e avanzati
- Diagrammi degli oggetti

3

Ingegneria del software mod. A

Riccardo Cardin

DIAGRAMMI DELLE CLASSI



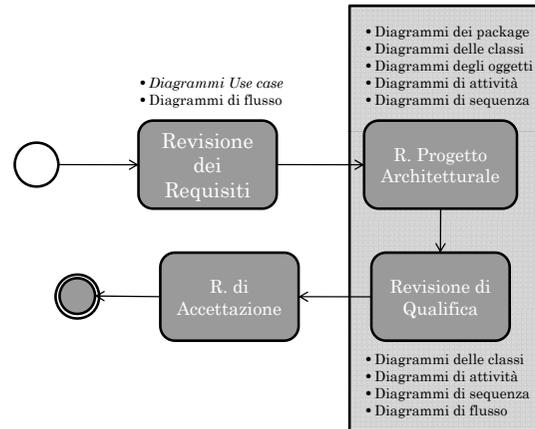
4

Ingegneria del software mod. A

Riccardo Cardin

DIAGRAMMI DELLE CLASSI

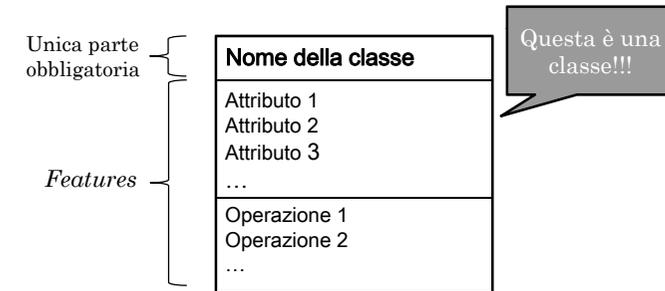
o Specifica Tecnica, Definizione di Prodotto



DIAGRAMMI DELLE CLASSI

o Definizione

- Descrizione del tipo degli oggetti che fa parte di un sistema
 - o Relazioni statiche fra i tipi degli oggetti

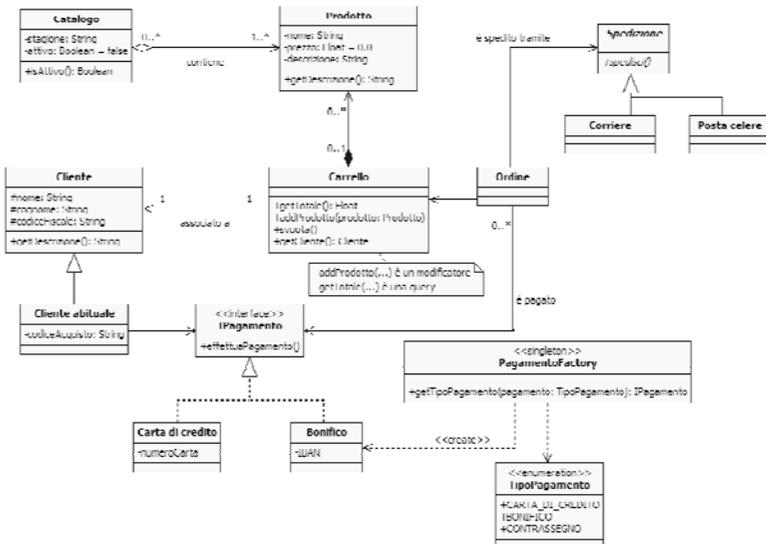


DIAGRAMMI DELLE CLASSI

o Esempio principale

Esempio

Il cliente sfoglia il catalogo ed aggiunge i prodotti desiderati al carrello della spesa. Quando il cliente termina l'acquisto e deve pagare, lo stesso fornisce le informazioni sulla consegna dei prodotti e sulla carta di credito. Il sistema verifica l'autorizzazione al pagamento con carta di credito e conferma l'acquisto immediatamente e mediante una successiva mail.



SOMMARIO

- Introduzione
- Proprietà e Operazioni
- Concetti base e avanzati
- Diagrammi degli oggetti

9

PROPRIETÀ

- Caratteristiche strutturali

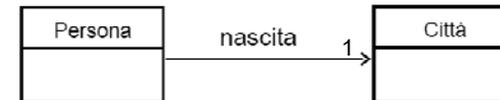
- Attributo

Definizione

```
visibilità nome : tipo molteplicità = default {proprietà aggiuntive}
```

- Visibilità: + pubblica, - privata, # protetta
- Associazione

- Linea continua e orientata fra due classi



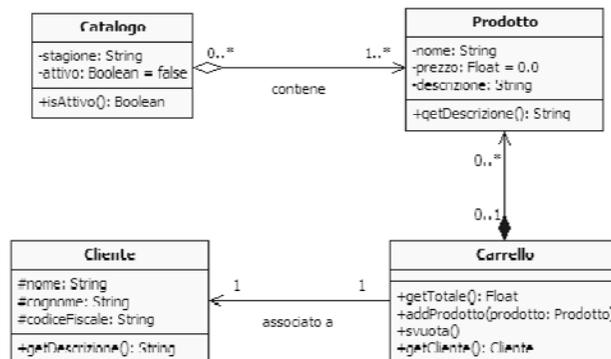
- Molteplicità

- Quanti oggetti possono far parte dell'
 - 1, 0..1, 0..*, *,...
- Spesso interscambiabile con un attributo

10

PROPRIETÀ

- Esempio 1



11

PROPRIETÀ

- ...nel linguaggio di programmazione

- Attributi

- Membri di classe (privati, se possibile)
- Molteplicità
 - Se *ordered*: Array o vettori
 - Se *unordered*: insiemi
- Convenzioni dei gruppi di programmazione
 - Esempio: *Getter* e *setter* per ogni attributo

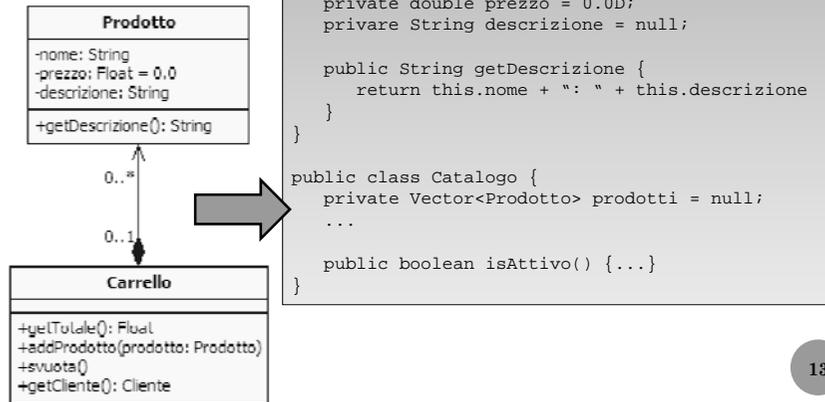
- Associazioni

- Anche se etichettata con verbo, meglio renderla con un nome
- Evitare le associazioni bidirezionali
 - Di chi è la responsabilità di aggiornare la relazione?

12

PROPRIETÀ

◦ Esempio 2



13

OPERAZIONI

◦ Le azioni che la classe “sa eseguire”

- Aspetto comportamentale
- Servizio che può essere richiesto ad ogni istanza della classe

Definizione

Visibilità nome (lista-parametri) : tipo-ritorno {proprietà aggiuntive}

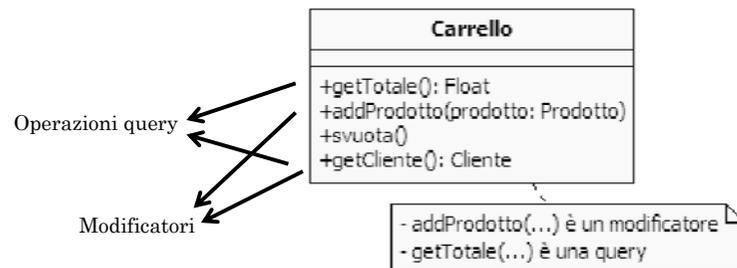
Lista-parametri := direzione nome : tipo = default

- Direzione: **in**, **out**, **inout** (*default in*)
- Visibilità: + pubblica, - privata, # protetta
- Query
- Modificatori
- Operazione ≠ metodo

14

OPERAZIONI

◦ Esempio 3



15

SOMMARIO

- Introduzione
- Proprietà e Operazioni
- Concetti base e avanzati
- Diagrammi degli oggetti

16

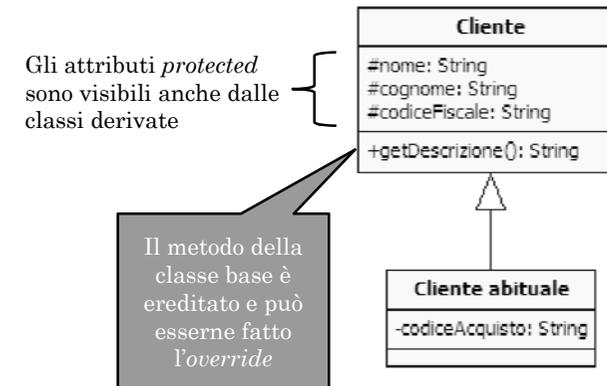
GENERALIZZAZIONE

- A generalizza B, se ogni oggetto di B è anche un oggetto di A
 - Equivale all'ereditarietà dei linguaggi di programmazione
 - Le proprietà della superclasse non si riportano nel diagramma della sottoclasse
 - A meno di *override*
 - Sostituibilità
 - Sottotipo \neq sottoclasse

17

GENERALIZZAZIONE

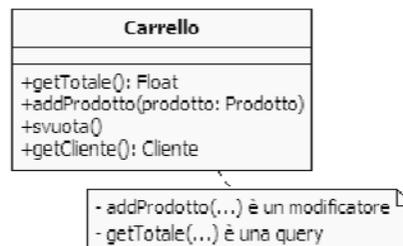
- Esempio 4



18

COMMENTI E NOTE

- Informazioni aggiuntive
 - Singole e solitarie
 - Legate a qualsiasi elemento grafico
 - Linea tratteggiata
 - Esempio 5



19

RELAZIONE DI DIPENDENZA

- Definizione

Si ha dipendenza tra due elementi di un diagramma se la modifica alla definizione del primo (fornitore) può cambiare la definizione del secondo (client)

- UML permettere di modellare ogni sorta di dipendenza
 - Non è una proprietà transitiva!
- Le dipendenze vanno minimizzate!
 - *Loose coupling*
- Da inserire solo quando danno valore aggiunto
 - Troppe dipendenze creano confusione nel diagramma

20

RELAZIONE DI DIPENDENZA

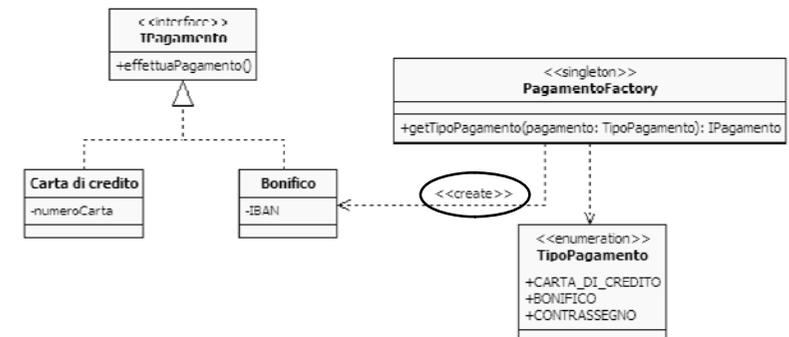
o Dipendenze UML

Parola chiave	Significato
«call»	La sorgente invoca un'operazione della classe destinazione.
«create»	La sorgente crea istanze della classe destinazione.
«derive»	La sorgente è derivata dalla classe destinazione
«instantiate»	La sorgente è una istanza della classe destinazione (meta-classe)
«permit»	La classe destinazione permette alla sorgente di accedere ai suoi campi privati.
«realize»	La sorgente è un'implementazione di una specifica o di una interfaccia definita dalla sorgente
«refine»	Raffinamento tra differenti livelli semantici.
«substitute»	La sorgente è sostituibile alla destinazione.
«trace»	Tiene traccia dei requisiti o di come i cambiamenti di una parte di modello si colleghino ad altre
«use»	La sorgente richiede la destinazione per la sua implementazione.

21

RELAZIONE DI DIPENDENZA

o Esempio 6

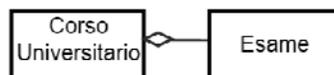


22

AGGREGAZIONE E COMPOSIZIONE

o Aggregazione

- Relazione di tipo "parte di" (*part of*)



o Composizione

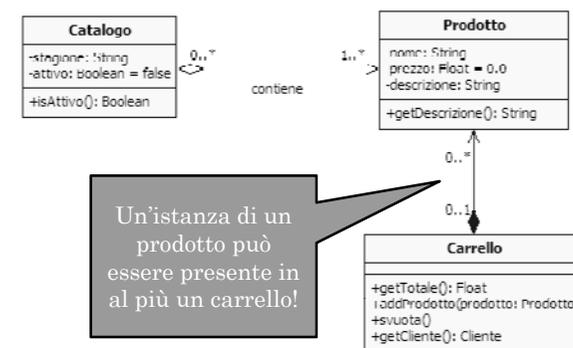
- Come associazione, ma:
 - o Ogni istanza deve essere componente di un solo oggetto
 - o Solo l'oggetto intero può creare e distruggere le sue parti



23

AGGREGAZIONE E COMPOSIZIONE

o Esempio 7

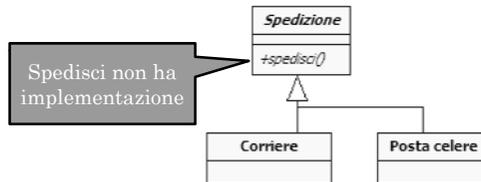


24

INTERFACCE E CLASSI ASTRATTE

o Classe Astratta {abstract}

- Classe che non può essere istanziata
 - o Operazione astratta non ha implementazione



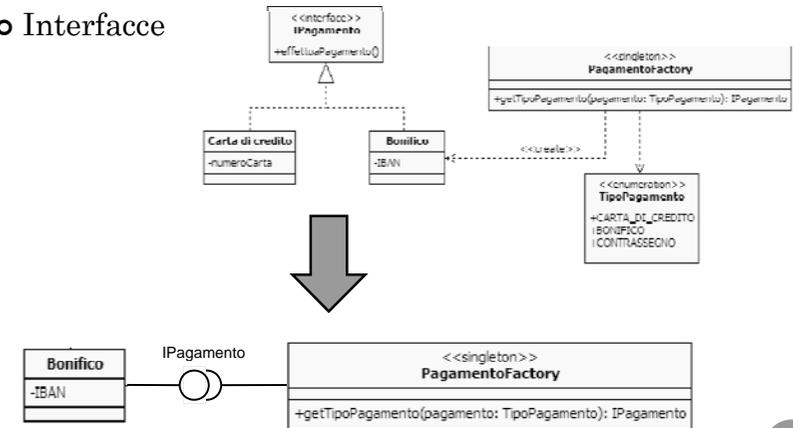
o Interfaccia «interface»

- Classe priva di implementazione
 - o Una classe realizza un'interfaccia se ne implementa le operazioni

25

INTERFACCE E CLASSI ASTRATTE

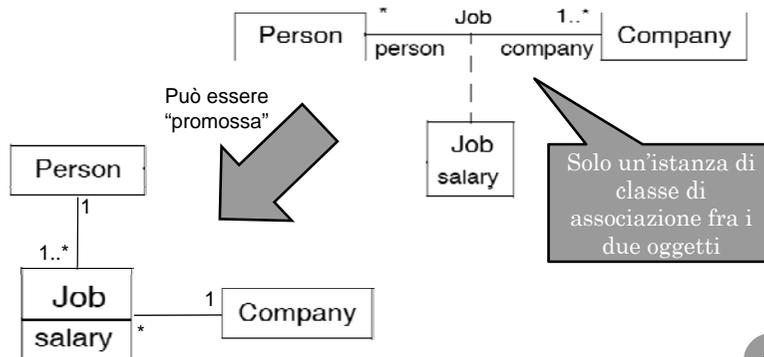
o Interfacce



26

CLASSI DI ASSOCIAZIONE

o Aggiungono attributi e operazioni alle associazioni



27

CLASSIFICAZIONE E GENERALIZZAZIONE

o Sottotipo ≠ “è un” (IS A)

Generalizzazione

- *Un Border Collie è un cane*
- *I cani sono animali*
- *I cani sono una specie*

Classificazione

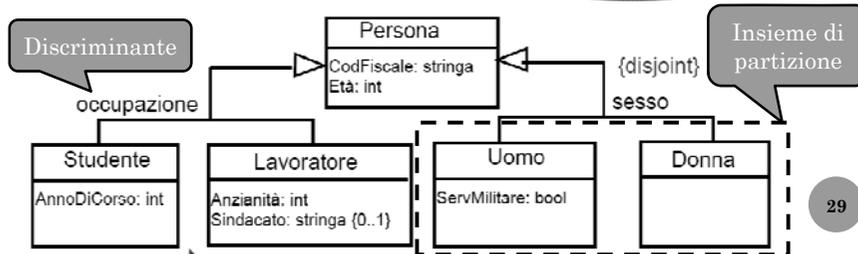
- *Shep è un Border Collie*
- *Border Collie è una razza*

- Generalizzazione
 - o Proprietà transitiva
 - o La classificazione non lo è!
- Classificazione
 - o Dipendenza «instantiated»

28

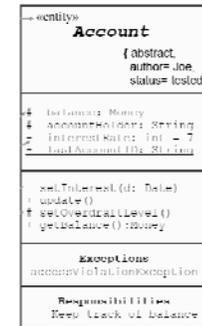
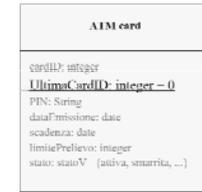
CLASSIFICAZIONE MULTIPLA

- Un oggetto può essere descritto da più tipi
 - Non necessariamente collegati da ereditarietà...
 - Non viene definito un nuovo tipo
- Insieme di generalizzazione
 - Appartenenza al più ad un tipo da ogni insieme
 - (Studente, Uomo); (Lavoratore, Donna); (Studente, Uomo, Donna)



CARATTERISTICHE VARIE

- Operazioni e attributi statici
 - Applicabili alla classe, non all'oggetto
 - Sottolineati sul diagramma
- Parole chiave
 - Estensione della semantica UML
 - Costrutto simile + parola chiave!
 - «interface»
 - {abstract}
- Responsabilità
 - Funzionalità offerte
 - Aggiunta alla classe con commento



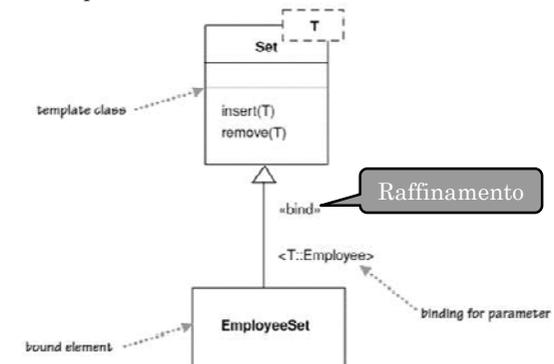
CARATTERISTICHE VARIE

- Proprietà derivate
 - Possono essere calcolate a partire da altri valori
 - Definiscono un vincolo fra valori
 - Si indicano con "r" che precede il nome della proprietà
- Proprietà *read only* e *frozen*
 - {readOnly}
 - {frozen}
 - Immutabile, non può variare nel suo ciclo di vita
- Enumerazioni
 - Insiemi di valori che non hanno altre proprietà oltre il valore simbolico
 - «enumeration»



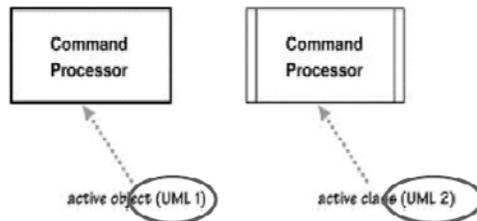
CARATTERISTICHE VARIE

- Classi Parametriche
 - T è detto "segnaposto"
 - Come *template* C++



CARATTERISTICHE VARIE

- Classi Attive
 - Eseguono e controllano il proprio *thread*



33

CONSIGLI UTILI

- Diagrammi molto ricchi di concetti
 - Non cercare di utilizzare tutte le notazioni disponibili
 - Cominciare dapprima con i concetti semplici
- Una prospettiva concettuale permette di esplorare il linguaggio di un particolare *business*
 - Mantenere la notazione semplice e non introdurre concetti legati al *software*
- Concentrarsi nel disegno dei diagrammi delle parti più importanti
 - Disegnare ogni cosa è sinonimo di diagrammi non fondamentali che diventano obsoleti molto presto!

34

SOMMARIO

- Introduzione
- Proprietà e Operazioni
- Concetti base e avanzati
- Diagrammi degli oggetti

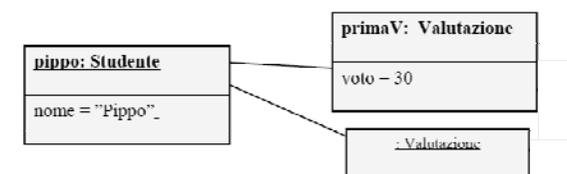
35

DIAGRAMMI DEGLI OGGETTI

- Grafo delle istanze, comprensivo di associazioni e valori delle proprietà

nome dell'istanza : nome della classe

- Fotografia degli oggetti che compongono un sistema
- Non ci sono parti obbligatorie
- Specifica di istanza
 - Anche di classi astratte, omissione dei metodi, ecc...



36

RIFERIMENTI

- OMG Homepage – www.omg.org
- UML Homepage – www.uml.org
- UML Distilled, Martin Fowler, 2004, Pearson (Addison Wesley)