



# DIAGRAMMI DEI PACKAGE

## INGEGNERIA DEL SOFTWARE

Università degli Studi di Padova

Dipartimento di Matematica

Corso di Laurea in Informatica, A.A. 2015– 2016

rcardin@math.unipd.it



# SOMMARIO

- o Cos'è un package
- o Diagrammi dei package

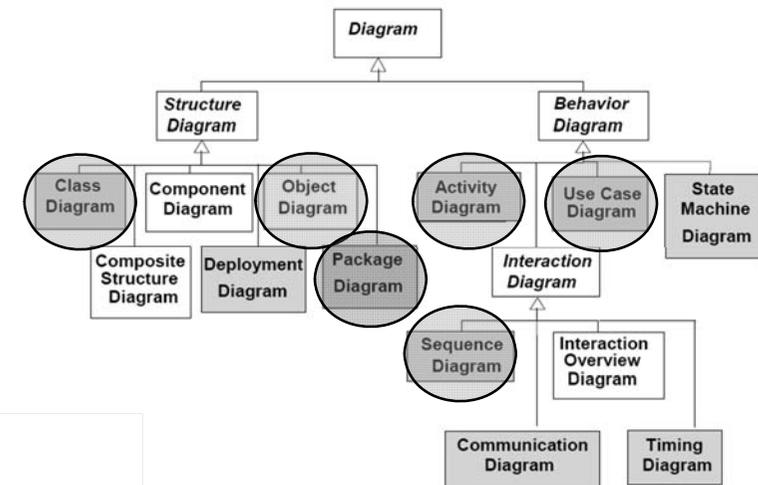


# SOMMARIO

- o Cos'è un package
- o Diagrammi dei package



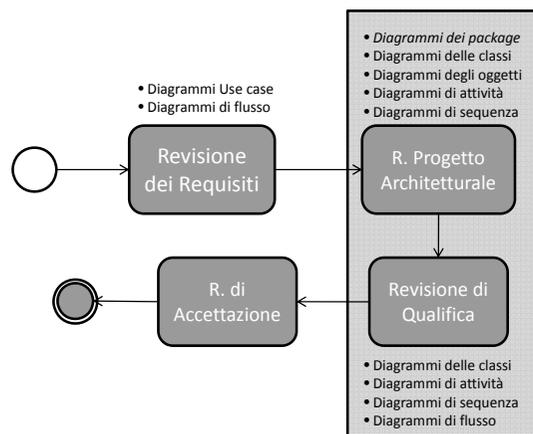
# DIAGRAMMI DEI PACKAGE



# DIAGRAMMI DEI PACKAGE



## o Specifica Tecnica

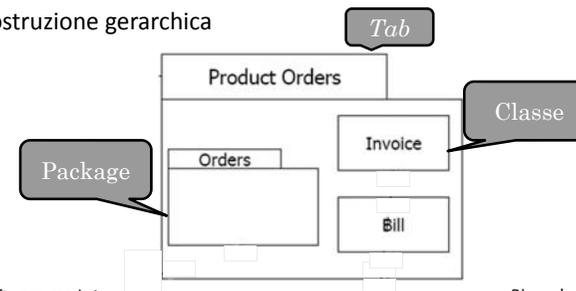


# COS'È UN PACKAGE



Raggruppamento di un numero arbitrario di elementi UML in una unità di livello più alto

- Praticamente si raggruppano solo classi
- Ogni classe appartiene ad un solo *package*
- Un *package* può contenere un altro *package*
  - o Costruzione gerarchica

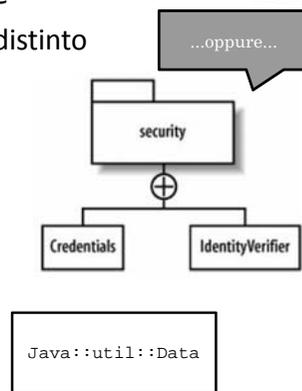
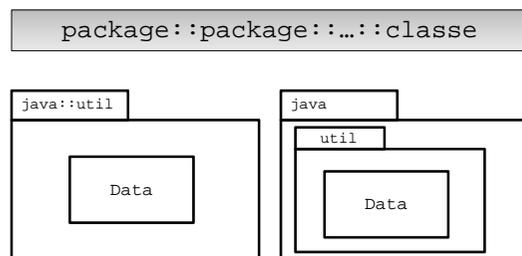


# COS'È UN PACKAGE



## o Il *package* individua un *namespace*

- Ogni elemento deve avere un nome distinto all'interno dello "spazio di nomi"
- Nome completamente qualificato

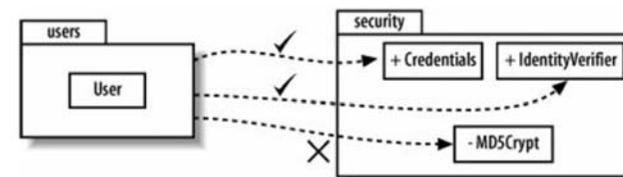


# COS'È UN PACKAGE



## o Visibilità

- Gli elementi del *package* possono avere visibilità pubblica (+) o privata (-)



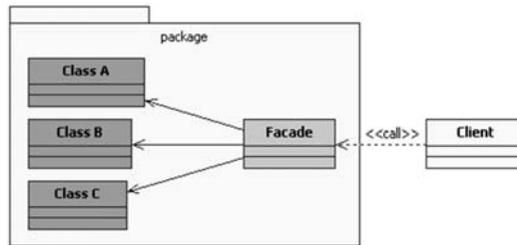
- Visibilità pubblica
  - o `public class Credential {}`
- Visibilità privata
  - o `class MD5Crypt {}`

# COS'È UN PACKAGE



## ◦ Interfaccia del *package*

- Insieme dei tipi pubblici in un *package*
  - Possono essere definite anche classi private
- Design pattern Façade



# COS'È UN PACKAGE



## ◦ Principi di progettazione

- *Common Closure Principle*
  - Classi dello stesso *package* condividono la stessa causa di cambiamento.
- *Common Reuse Principle*
  - Classi dello stesso *package* dovrebbero essere sempre riusate insieme.

# SOMMARIO



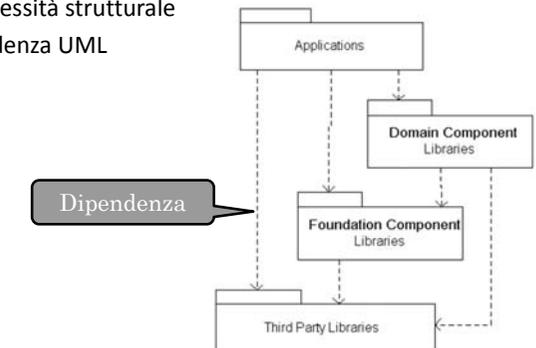
- Cos'è un package
- Diagrammi dei package

# PACKAGE E DIPENDENZE



## ◦ Diagramma dei *package*

- Documenta le dipendenze fra le classi
  - In sistemi medio-grandi è utilissimo per tenere sotto controllo la complessità strutturale
  - ... qualsiasi dipendenza UML

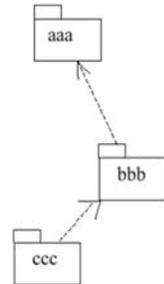


## PACKAGE E DIPENDENZE



### o Caratteristiche

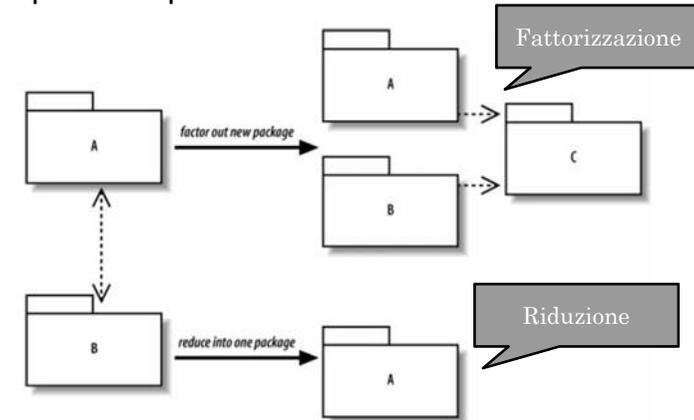
- Tutte le dipendenze dovrebbero seguire la stessa direzione
  - o A meno di isolamento voluto di sottostrutture
- Evitare le dipendenze circolari
  - o *Acyclic Dependency Principle*
- Relazioni di dipendenza non sono transitive
  - o Se modifico aaa, non necessariamente devo modificare ccc
- Più dipendenze entranti, più il *package* dovrebbe essere stabile
  - o *Package «global»*



## PACKAGE E DIPENDENZE



### o Rompere le dipendenze circolari

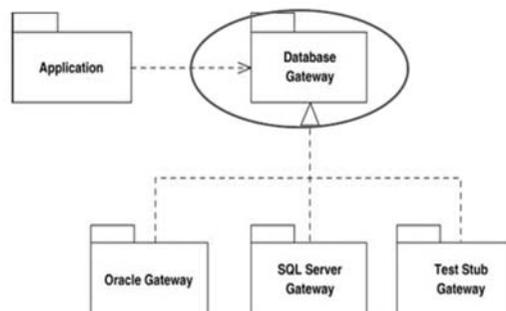


## PACKAGE E DIPENDENZE



### o *Package* di interfaccia

- Contiene interfacce e classi astratte
- Design Pattern “Interfaccia separata”



## RIFERIMENTI



- o OMG Homepage – [www.omg.org](http://www.omg.org)
- o UML Homepage – [www.uml.org](http://www.uml.org)
- o UML Distilled, Martin Fowler, 2004, Pearson (Addison Wesley)
- o Learning UML 2.0, Kim Hamilton, Russell Miles, O'Reilly, 2006