

Encapsulation



Encapsulation

- L'obiettivo principale degli OOP è sviluppare codice riutilizzabile in diversi contesti.
Per fare ciò è necessario nascondere i dettagli implementativi ed esporre solo le informazioni utilizzabili dall'utente
- Un programma è ben incapsulato quando i dati sono confinati nei metodi che li usano.
- Buona encapsulation = codice meglio mantenibile



Packages

- Permettono di raggruppare le classi in unità
- Il nome è strutturato come un url al contrario (dal generale allo specifico) e va inserito nella prima riga del file della classe
- Se si vuole usare una classe appartenente ad un altro package, bisogna importarla

```
it
└─ unive
    └─ dais
        ...
            └─ Class.java
```

```
package it.unive.dais....

import it.unive.dais.package1.Class1
import it.unive.dais.package2.*

class Class{...}
```



Information hiding

- Permette una corretta encapsulation, esponendo solamente l'interfaccia utilizzabile da un utente
- Ottenuta utilizzando i modificatori di accesso
- I vantaggi sono:
 - Riduzione dipendenze
 - Semplificazione interfacce
 - Riduzione di errori d'uso



Modificatori di accesso

	Classe	Pacchetto	Sottoclasse	Ovunque
public	✓	✓	✓	✓
protected	✓	✓	✓	✗
<default>	✓	✓	✗	✗
private	✓	✗	✗	✗

I campi e metodi private sono accessibili in altri oggetti appartenenti alla stessa classe

- **public:** interfaccia della classe
- **private:** informazioni non interessanti per gli utenti
- **default:** dati e funzioni utili agli sviluppatori

Getters e setters

- Il problema principale dei campi pubblici che possono essere modificati in maniera arbitraria
- Si può rendere il campo privato e creare delle funzioni pubbliche di *get* e *set*, che permettono di aggiungere logica di controllo sull'input e l'output

Come incapsulare

1. Definire i componenti dell'interfaccia
 2. Impostare il resto come private
 3. Decidere se per certi componenti si può rilassare la visibilità o bisogna migliorare l'encapsulation
- Evitare dipendenze circolari, se non assolutamente necessarie

