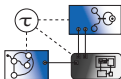


Il pattern Façade

a cura di **Angelo Furfaro**
da “Design Patterns”, Gamma et al.
“Patterns in Java”, Grand



Laboratorio di Ingegneria del Software
Dipartimento di Ingegneria Informatica, Elettronica Modellistica e Sistemistica
Università della Calabria, 87036 Rende(CS) - Italy
Email: a.furfaro@deis.unical.it
Web: <http://www.lis.deis.unical.it/~furfaro>

Classificazione

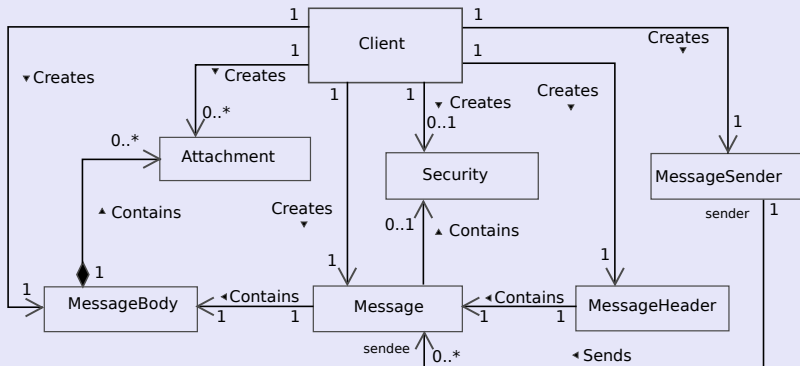
- Scopo: strutturale
- Raggio d'azione: basato oggetti

Scopo

- Fornire un'interfaccia unificata per un insieme di interfacce presenti in un sottosistema.
- Façade definisce un'interfaccia di livello più alto che rende il sistema più semplice da utilizzare

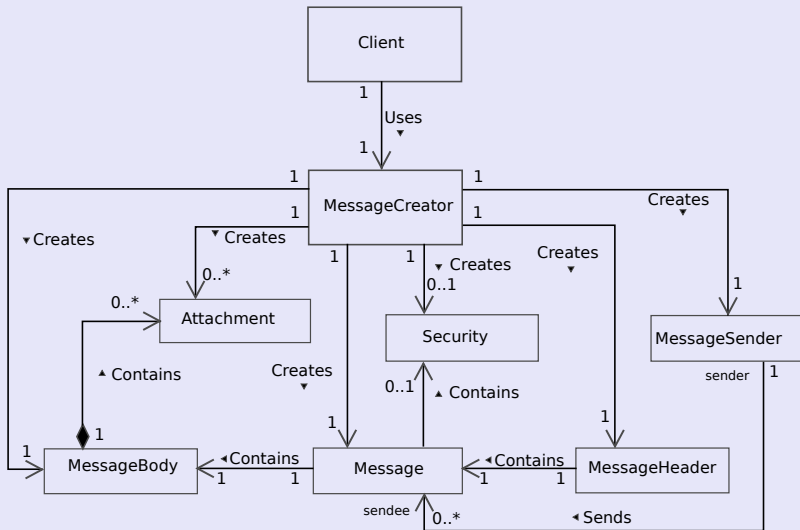
Esempio

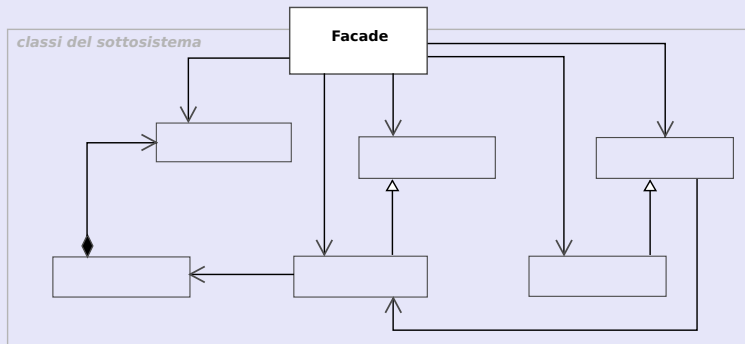
Creazione di un messaggio e-mail



Esempio

Creazione di un messaggio e-mail (Façade)





Partecipanti

- **Facade** (MessageCreator):
conosce le classi nel sottosistema che sono responsabili di gestire una richiesta.
- **Classi del sottosistema** (Message, MessageBody, Attachment, etc.):
 - Implementano le funzionalità del sottosistema.
 - Non hanno alcuna conoscenza dell'esistenza del Facade: non hanno alcun riferimento ad esso.

Conseguenze

- ☺ Nasconde ai client i componenti del sottosistema, riducendo il numero degli oggetti con cui i client interagiscono.
- ☺ Promuove un basso accoppiamento tra un sottosistema ed i suoi client. I componenti del sottosistema sono molto accoppiati tra di loro (alta coesione interna). La riduzione delle dipendenze è di fondamentale importanza nei sistemi di grandi dimensioni.
- ☺ In particolare, riduce le dipendenze di compilazione: se le classi del sottosistema sono modificate non è necessario ricompilare il codice dei client.
- ☺ Non impedisce alle applicazioni client di utilizzare le classi del sottosistema qualora sia necessario.

Pattern correlati

- Abstract Factory
- Mediator
- Singleton