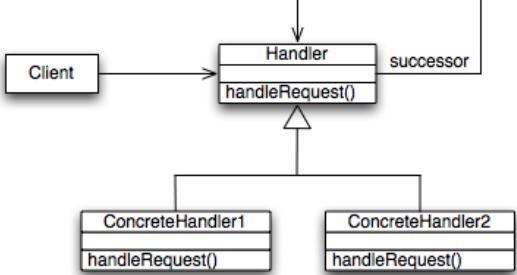


# Chain of Responsibility

- Intento
  - Evitare di accoppiare il mandante di una richiesta con il ricevente, dando così la possibilità a più di un oggetto di gestire la richiesta. Concatena gli oggetti riceventi e passa la richiesta tra questi fino a che un oggetto la gestisce
- Motivazione
  - Consideriamo una interfaccia utente dove l'utente può richiedere aiuto su parti dell'interfaccia. Per es. un bottone può fornire informazioni di aiuto. Se non esiste un'informazione specifica allora il sistema dovrebbe fornire il messaggio d'aiuto del contesto più vicino (ad es. la finestra)
  - Il problema: l'oggetto che fornirà il messaggio d'aiuto non è conosciuto dall'oggetto (es. Button) che inizia la richiesta
  - Disaccoppiare mandante e ricevente

# Soluzione

- Struttura
- Partecipanti
  - Handler definisce l'interfaccia per gestire le richieste
  - ConcreteHandler gestisce la richiesta per cui è responsabile, può accedere al suo successore, inoltra la richiesta se non può gestirla
  - Client inizia effettuando la richiesta ad un ConcreteHandler
- La richiesta si propaga lungo la catena finché un ConcreteHandler la può gestire

# Chain of Responsibility

- Conseguenze
  - Riduce l'accoppiamento: chi fa una richiesta non conosce il ricevente e viceversa
  - Un oggetto della catena non deve conoscere la struttura della catena
  - Gli oggetti handler anziché mantenere i riferimenti a tutti i candidati riceventi mantengono solo il riferimento al successore
  - Si aggiunge flessibilità nella distribuzione di responsabilità agli oggetti. Si può cambiare o aggiungere responsabilità nella gestione di una richiesta cambiando la catena a runtime
  - Non c'è garanzia che una richiesta venga gestita, poiché non c'è un ricevente esplicito, la richiesta potrebbe arrivare alla fine della catena senza essere gestita

# Chain of Responsibility

- Implementazione
  - Handler o ConcreteHandler possono definire i link per avere il successore della catena
  - Una gerarchia già esistente può essere usata, anziché ridefinire i link (vedi Composite ed il riferimento al parent), se la gerarchia riflette la catena, altrimenti i link vanno definiti
  - Tipi di richieste
    - Ciascun tipo di richiesta può corrispondere ad una operazione. In questo caso il set di richieste è definito dall'Handler
    - In alternativa, a ciascun tipo di richiesta corrisponde un codice e solo una operazione è definita. Il set di richieste non è fissato, richiedente e riceventi prendono accordi sulla codifica delle richieste. Più controlli a runtime