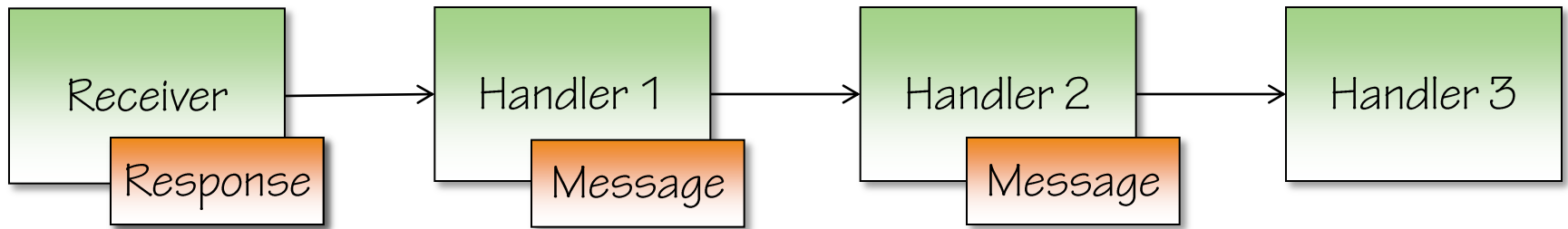


# Chain of Responsibility

Design Patterns



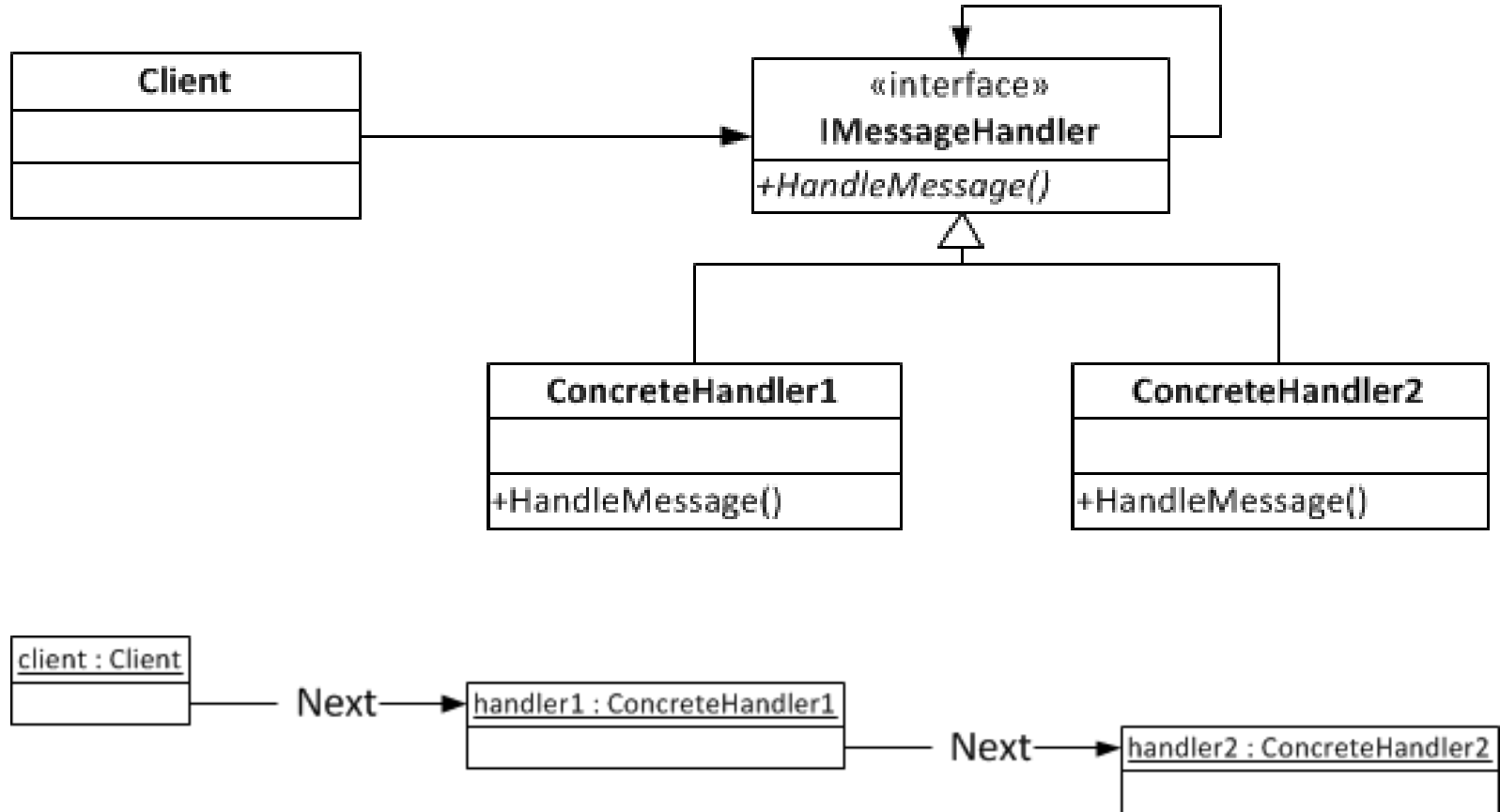
# Overview



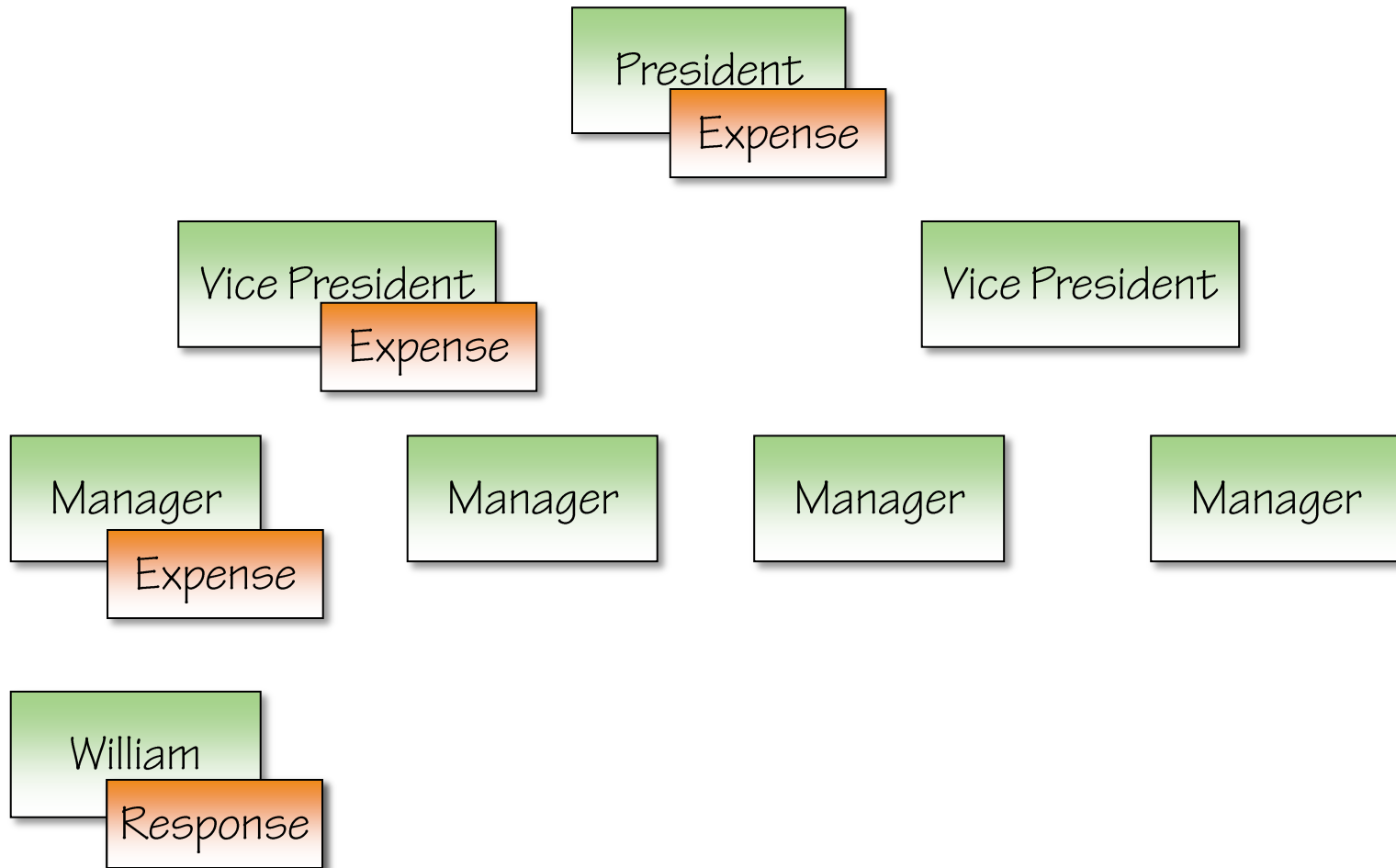
# Chain of Responsibility Traits

- Receiver is aware of only one handler
- Each handler is only aware of the next handler
- Handlers either process the message or send it to the next handler
- The receiver does not know who handled the message
- The first handler to process the message terminates the chain
- The order of the handler list matters

# Traditional Usage



# Motivating Example



# Poker Ranking

Hand

Royal Flush

Straight Flush

Four of a Kind

Full House

Flush

*etc...*

Hand

Hand

Hand

# Summary

- **Use the Chain of Responsibility when...**
  - More than one message handler for a message
  - The appropriate handler is not explicitly known by the sender
  - The set of handlers can be dynamically defined
- **Benefits of the Chain of Responsibility**
  - Reduced coupling
  - Dynamically manage the message handlers
  - End of chain behavior can be defined appropriately
- **Related Patterns**
  - Composite
  - Tree of Responsibility

For more in-depth **online** developer **training** visit



**on-demand** content from authors you **trust**