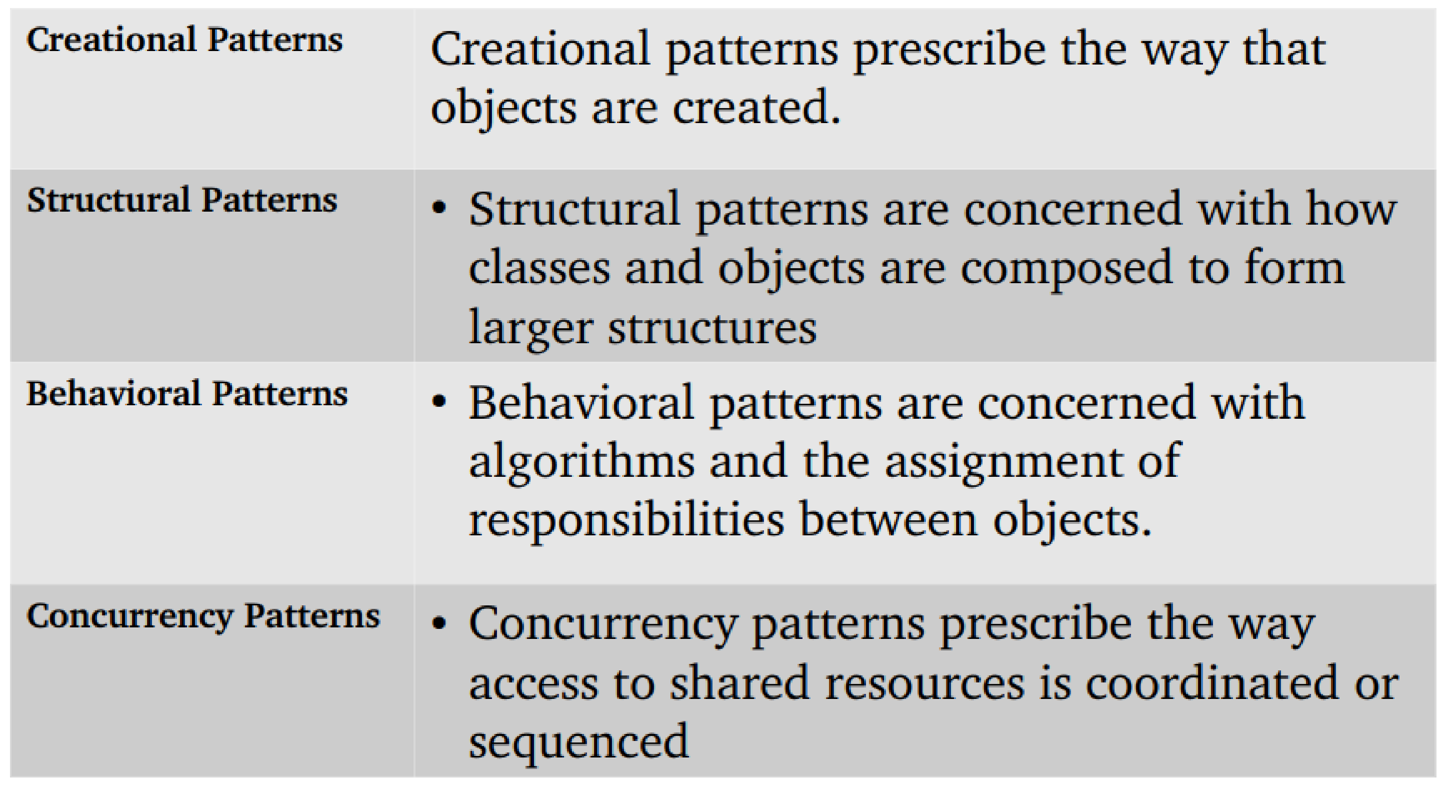
Pola Desain Perangkat Lunak

[Week] 11 – Behavioral Pattern, Chain of Responsibility
Prepared by: Tifanny Nabarian

Design Patterns Category

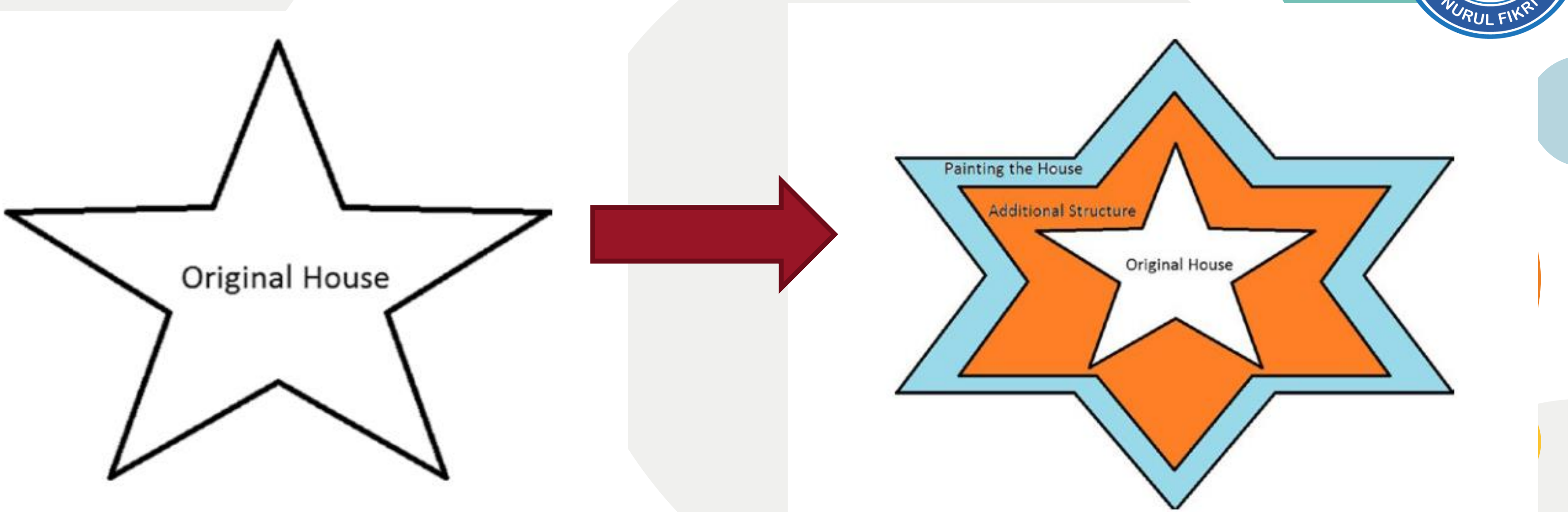


Creational Patterns	Creational patterns prescribe the way that objects are created.
Structural Patterns	• Structural patterns are concerned with how classes and objects are composed to form larger structures
Behavioral Patterns	• Behavioral patterns are concerned with algorithms and the assignment of responsibilities between objects.
Concurrency Patterns	• Concurrency patterns prescribe the way access to shared resources is coordinated or sequenced

Design Patterns Scope

		Purpose		
		Creational	Structural	Behavioral
Scope	Class	• Factory method	• Adapter	• Interpreter • Template method
	Object	• Abstract factory • Builder • Prototype • Singleton	• Adapter • Bridge • Composite • Decorator • Fasad • Flyweight • Proxy	• Chain of responsibility • Command • Iterator • Mediator • Memento • Observer • State • Strategy • Visitor

Review - Decorator



Original House

Painting the House

Additional Structure

Original House

Review - Decorator

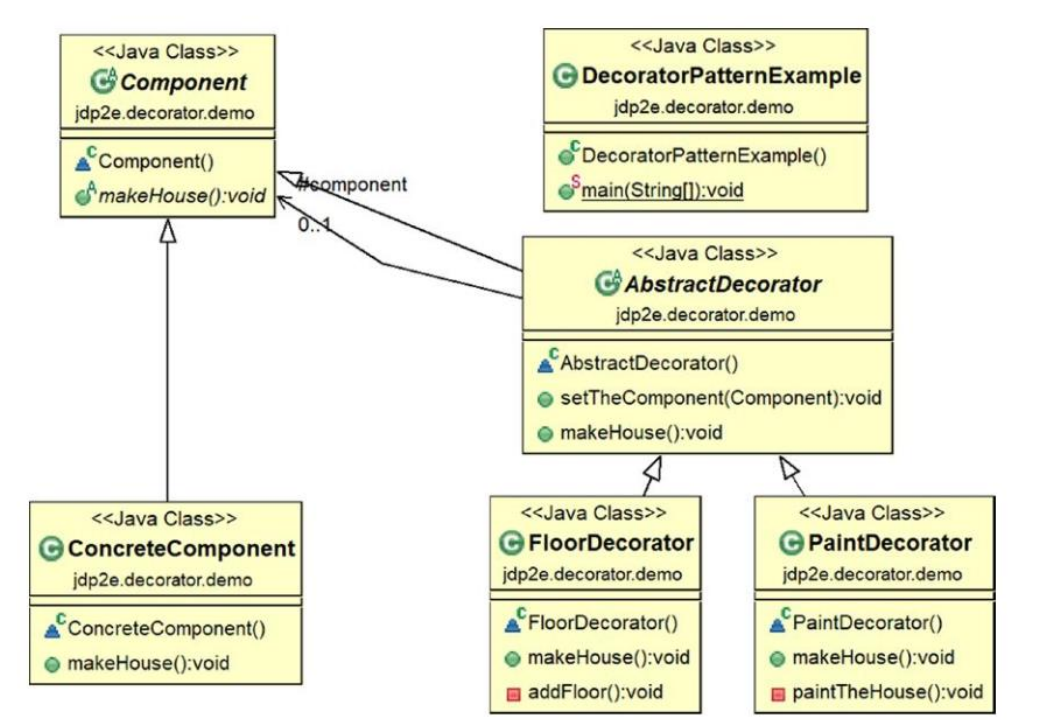
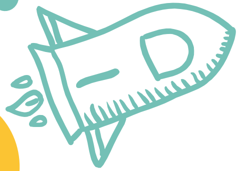


Figure 7-4. *Class diagram*



Behavioral Pattern

Chain of Responsibility

Chain of Responsibility Concept

GoF Definition

- **Avoid coupling** the sender of a request to its receiver by **giving more than one object** a chance to handle the request.
- Chain the **receiving** objects and **pass** the request along the chain until an object handles it.

Chain of Responsibility Concept

- In this pattern, you **form a chain of objects** where each object in the chain **handles a particular kind of request**.
- If **an object cannot handle** the request fully, **it passes the request to the next object** in the chain. The same process may follow until the end of a chain is reached.

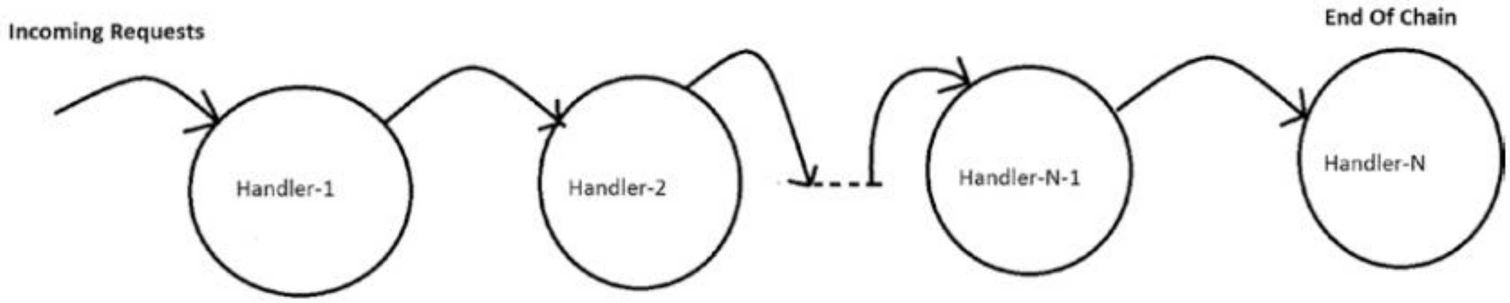


Figure 22-1. *The concept of a chain-of-responsibility pattern*

Chain of Responsibility Concept

Real World Example

- Each organization employs customer care executives who receive feedback or complaints directly from the customers. If the employees cannot answer the customers' issues properly, they forward these issues/escalations to the appropriate departments in the organization.
- A similar scenario occurs when a patient visits a hospital. Doctors from one department can refer a patient to a different department for further diagnosis.

Chain of Responsibility Concept

Computer World Example

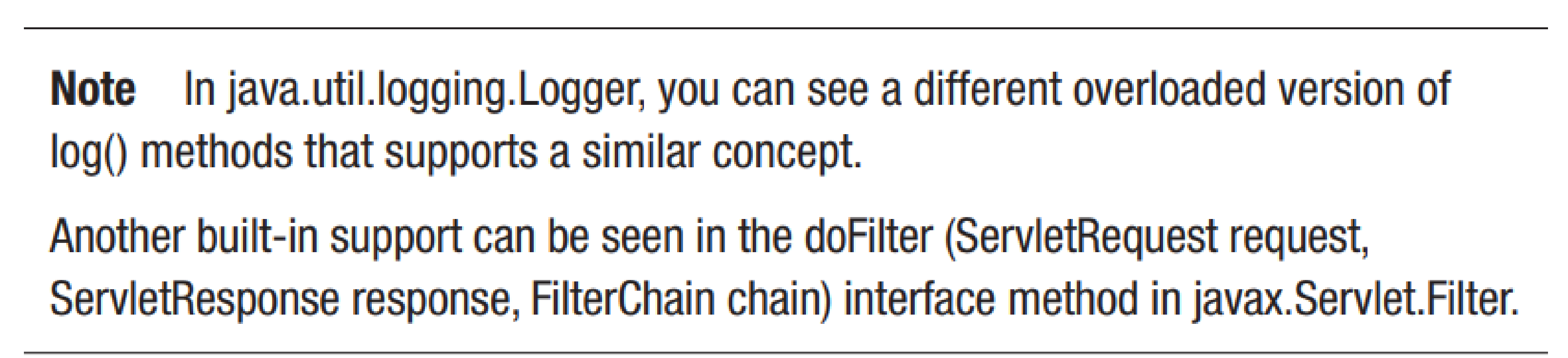
- Consider a software application (e.g., a printer) that can send emails and faxes. So, customers can report faxing issues or email issues. Let’s assume that these issues are handled by handlers. So, you introduce two different types of error handlers: **EmailErrorHandler** and **FaxErrorHandler**.
- Note that the chain ends with **EmailErrorHandler**. But if you need to handle a different type of issue—for example, authentication issues (which can occur due to security vulnerabilities), you can make a handler called **AuthenticationErrorHandler** and place it after **EmailErrorHandler**.

Chain of Responsibility Concept

Computer World Example

- Consider a software application (e.g., a printer) that can send emails and faxes. So, customers can report faxing issues or email issues. Let’s assume that these issues are handled by handlers. So, you introduce two different types of error handlers: **EmailErrorHandler** and **FaxErrorHandler**.
- Note that the chain ends with **EmailErrorHandler**. But if you need to handle a different type of issue—for example, authentication issues (which can occur due to security vulnerabilities), you can make a handler called **AuthenticationErrorHandler** and place it after **EmailErrorHandler**.

Chain of Responsibility Example

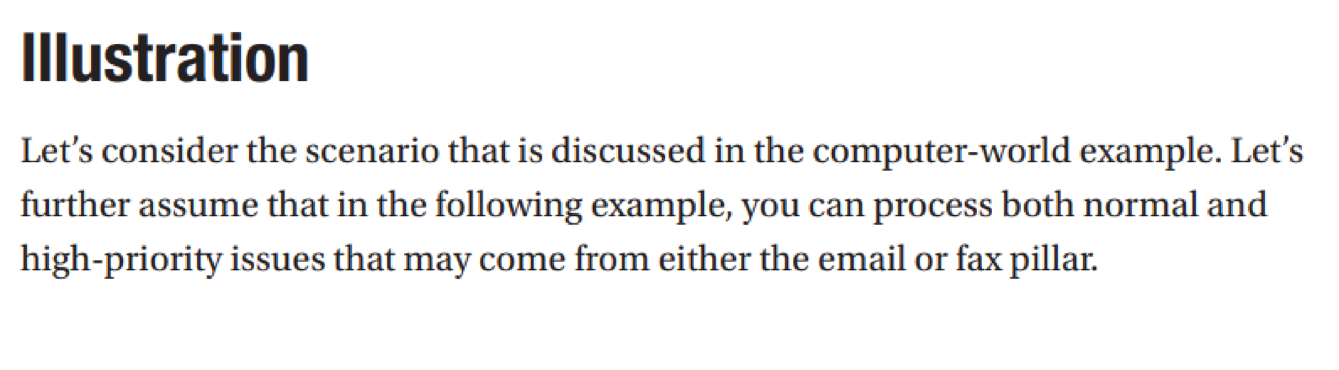


Note In java.util.logging.Logger, you can see a different overloaded version of log() methods that supports a similar concept.

Another built-in support can be seen in the doFilter (ServletRequest request, ServletResponse response, FilterChain chain) interface method in javax.Servlet.Filter.

Let's Practice!

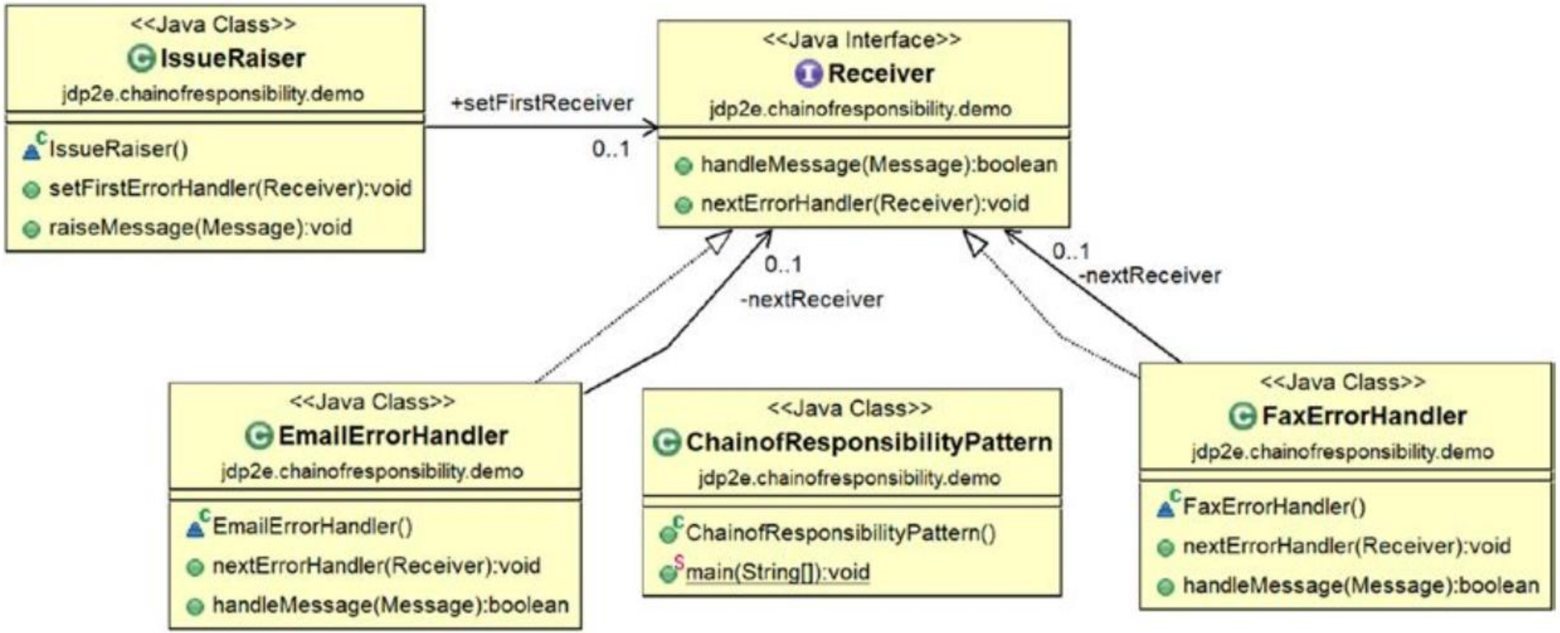
Chain of Responsibility Example



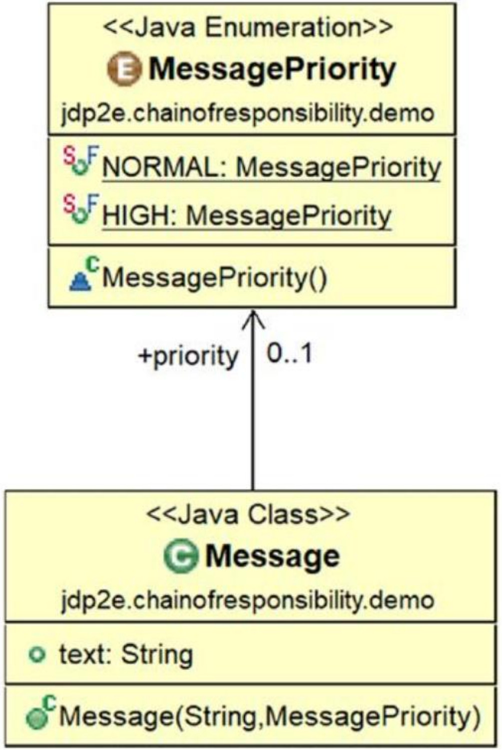
Illustration

Let's consider the scenario that is discussed in the computer-world example. Let's further assume that in the following example, you can process both normal and high-priority issues that may come from either the email or fax pillar.

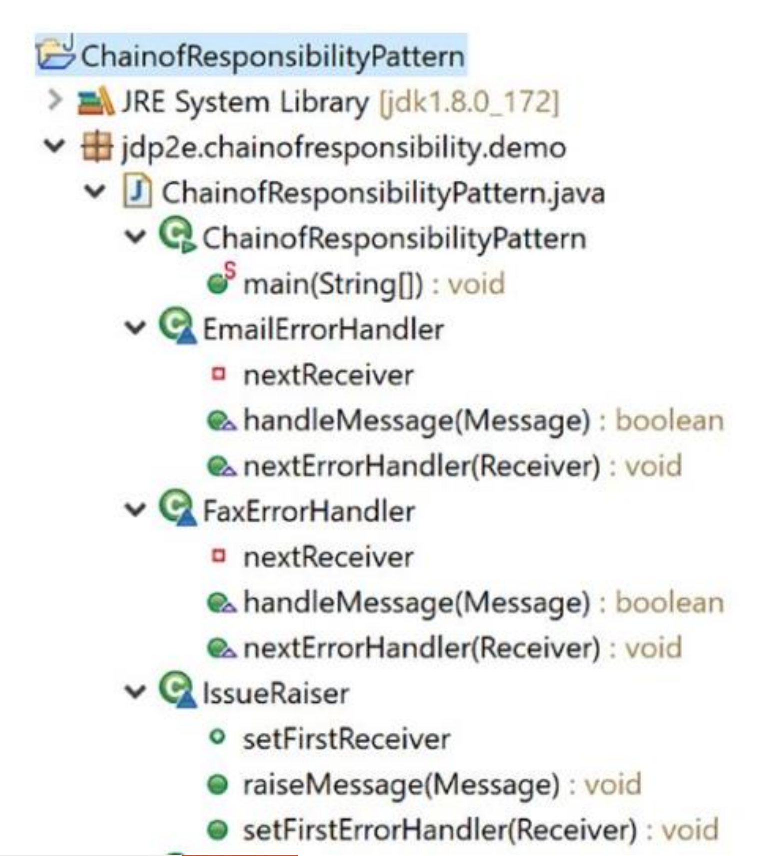
Class Diagram


<<Java Class>>
IssueRaiser
jdp2e.chainofresponsibility.demo
IssueRaiser()
setFirstErrorHandler(Receiver):void
raiseMessage(Message):void
+setFirstReceiver
0..1
<<Java Interface>>
Receiver
jdp2e.chainofresponsibility.demo
handleMessage(Message):boolean
nextErrorHandler(Receiver):void
0..1
-nextReceiver
0..1
-nextReceiver
<<Java Class>>
EmailErrorHandler
jdp2e.chainofresponsibility.demo
EmailErrorHandler()
nextErrorHandler(Receiver):void
handleMessage(Message):boolean
<<Java Class>>
ChainofResponsibilityPattern
jdp2e.chainofresponsibility.demo
ChainofResponsibilityPattern()
main(String[]):void
<<Java Class>>
FaxErrorHandler
jdp2e.chainofresponsibility.demo
FaxErrorHandler()
nextErrorHandler(Receiver):void
handleMessage(Message):boolean

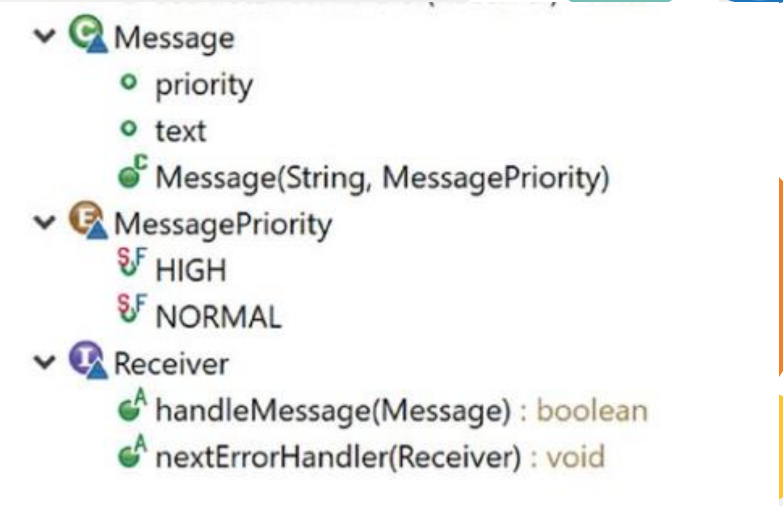
Class Diagram



Decorator Example (Package Explorer)



- ChainofResponsibilityPattern
  - JRE System Library [jdk1.8.0_172]
  - jdp2e.chainofresponsibility.demo
    - ChainofResponsibilityPattern.java
      - ChainofResponsibilityPattern
        - main(String[]) : void
      - EmailErrorHandler
        - nextReceiver
        - handleMessage(Message) : boolean
        - nextErrorHandler(Receiver) : void
      - FaxErrorHandler
        - nextReceiver
        - handleMessage(Message) : boolean
        - nextErrorHandler(Receiver) : void
      - IssueRaiser
        - setFirstReceiver
        - raiseMessage(Message) : void
        - setFirstErrorHandler(Receiver) : void



- Message
  - priority
  - text
  - Message(String, MessagePriority)
- MessagePriority
  - HIGH
  - NORMAL
- Receiver
  - handleMessage(Message) : boolean
  - nextErrorHandler(Receiver) : void

Silahkan Kerjakan
Tugas Praktikkum..

CHARACTER BUILDING CAMPUS
STT - NF
NURUL FIKRI