UML Class Diagram - Observer Design Pattern
For a Newsletter Subscription Service
Oluwafemi Oluwafunbi Tedunjaiye | April 16, 2023

<<Subject>>

+ setNewletter(String newsletter): void
+ registerObserver(Observer subscriber): void
+ removeObserver(Observer subscriber): void
+ notifyObservers( ): void

<<Observer>>

+ update(String newsletter): void

0...1   1

1

1   *

1

NewsletterSubscriptionService

- observers: List<Observer>
- observers: List<Observer>
- observers: List<Observer>
- newNewsletter: String

+ setNewsletter(String newsletter): void
+ registerObserver(Observer newObserver): void
- registerObserver(Observer newObserver, String subscriptionDuration): void
+ removeObserver(Observer newObserver): void
+ notifyObservers(): void
- notifyObservers(List<Observer> newObserver): void

NewsletterSubscriber

- newsletter: String
- subscriptionDuration: String

+update(String newsletter): void

Main

+ main(String[] args): void