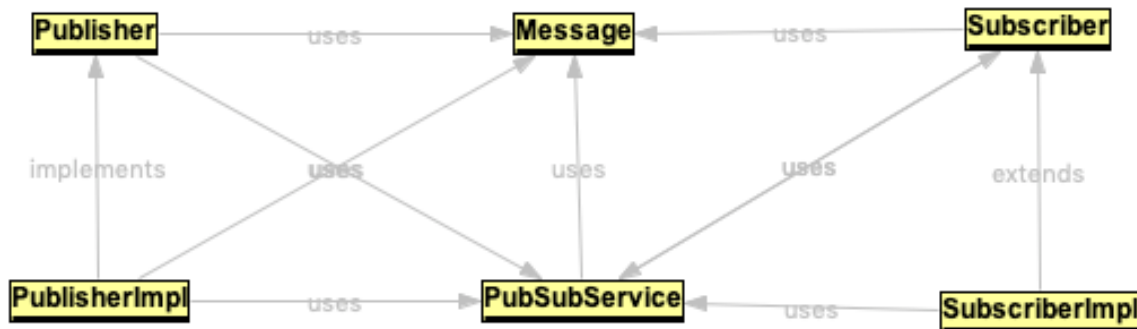


Publisher and Subscriber Pattern

1. Publisher and Subscriber 패턴

게시자(Publisher)가 다수의 구독자(Subscriber)에게 메시지를 보낼 때, 특정 구독자를 지정하여 보내는 것이 아니라, 구독자에 대한 특정 명시 없이 메시지를 일정 부류로 분류함으로써 메시지를 전달한다. 구독자는 하나 이상의 부류에 대한 관심을 표시하여 메시지를 구독한다. 구독자 또한 게시자를 특정하지 않는다.

2. Publisher and Subscriber 패턴 구조



3. Publisher and Subscriber 패턴 사용

Publisher and Subscriber를 사용하는 메인 함수는 다음과 같다¹.

```

public class ExampleRunner {
    public static void main(String[] args) {
        Publisher javaPublisher = new PublisherImpl();
        Subscriber javaSubscriber = new SubscriberImpl();
        PubSubService pubSubService = new PubSubService();
        //Declare Messages and Publish Messages to PubSubService
        Message javaMsg1 = new Message("Java", "Core Java Concepts");
        Message javaMsg2 = new Message("Java", "Spring MVC");
        javaPublisher.publish(javaMsg1, pubSubService);
        javaPublisher.publish(javaMsg2, pubSubService);

        //Declare Subscribers: Java subscriber only subscribes to Java topics
        javaSubscriber.addSubscriber("Java", pubSubService);

        //Broadcast message to all subscribers.
        //After broadcast, messageQueue will be empty in PubSubService
        pubSubService.broadcast();

        //Print messages of each subscriber to see which messages they got
        System.out.println("Messages of Java Subscriber are: ");
        javaSubscriber.printMessages();
    }
}
  
```

¹ <https://gist.github.com/roryokane/9606238>

4. Publisher and Subscriber 패턴 활용

- (1) 제공하는 소스는 자바 언어에 대해서만 게시자(Publisher)와 구독자(Subscriber)를 선언하여 활용하고 있다. 파이썬 언어에 대한 게시자(Publisher)와 구독자(Subscriber)를 선언하고 자바 언어에 대한 메시지를 출력하는 것과 같은 형태로 파이썬 관련 메시지를 3개 생성하여 게시자가 게시하면 구독자가 이를 받아 메시지를 출력하도록 하자.
- (2) 구독자는 여러 주제의 메시지를 구독할 수 있다. 두 개의 언어 모두를 구독하는 구독자(Subscriber)를 선언하고, 해당 구독자를 통해 발행한 두 개 언어에 대한 메시지를 모두 출력하도록 해보자.
- (3) 어떻게 게시자-구독자(Publisher and Subscriber) 구조로 상기와 같은 작업이 가능한지 코드를 분석하여 설명해 보자.