Лабораторная работа №7

**Цель работы**: ознакомиться с методологией объектноориентированного программирования, изучить реализацию данной методологии в языке Python 3.

**Задание:** 19)Написать класс Publisher и несколько различных классов с постфиксом Subscriber. Реализовать между данными классами отношения по паттерну Observer.

**Выполнение задания:** Для выполнения этой лабораторной работы потребуется создать класс Паблишер, который будет оповещать своих подписчиков Сабскрайбер о различных событиях  
  
Листинг кода:

class Publisher:  
 def \_\_init\_\_(self):  
 self.subscribers = []  
  
 def subscribe(self, subscriber):  
 self.subscribers.append(subscriber)  
 print(f"{subscriber.\_\_class\_\_.\_\_name\_\_} subscribed.")  
  
 def unsubscribe(self, subscriber):  
 self.subscribers.remove(subscriber)  
 print(f"{subscriber.\_\_class\_\_.\_\_name\_\_} unsubscribed.")  
  
 def notify(self, message):  
 for subscriber in self.subscribers:  
 subscriber.update(message)  
  
  
class SubscriberA:  
 def update(self, message):  
 print(f"SubscriberA received message: {message}")  
  
  
class SubscriberB:  
 def update(self, message):  
 print(f"SubscriberB received message: {message}")  
  
  
class SubscriberC:  
 def update(self, message):  
 print(f"SubscriberC received message: {message}")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 publisher = Publisher()  
  
 subscriber\_a = SubscriberA()  
 subscriber\_b = SubscriberB()  
 subscriber\_c = SubscriberC()  
  
 publisher.subscribe(subscriber\_a)  
 publisher.subscribe(subscriber\_b)  
 publisher.subscribe(subscriber\_c)  
  
 publisher.notify("Hello, Subscribers!")  
  
 publisher.unsubscribe(subscriber\_b)  
  
 publisher.notify("Goodbye, Subscribers!")

Результаты работы кода:



Рис. 1 - Результаты работы кода