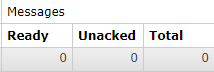
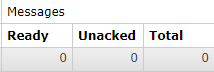
Szymon Zieliński   
nr indexu 296722

**Zad 1a (1 punkt). Zaobserwuj działanie mechanizmu niezawodności w różnych scenariuszach potwierdzeń wiadomości.**

* Po otrzymaniu wiadomości:
  + Wiadomość zostaje normalnie odebrana w obydwu przypadkach

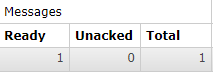


* Po przetworzeniu wiadomości:
  + Konsument zostaje zrestartowany po zakończeniu przetwarzania wiadomości:



Wiadomość zostaje normalnie odebrana

* + Konsument zostaje zrestartowany w trakcie przetwarzania wiadomości:

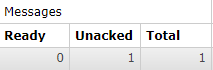


Wiadomość przechodzi po chwili ze stanu unacked do stanu ready, można ją ponownie odebrać.

* Który sposób potwierdzeń zapewnia większą niezawodność?

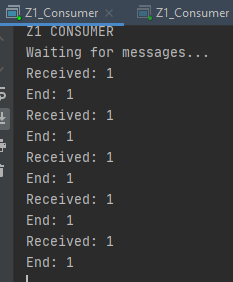
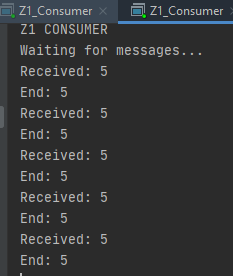
Potwierdzenie dopiero po przetworzeniu wiadomości.

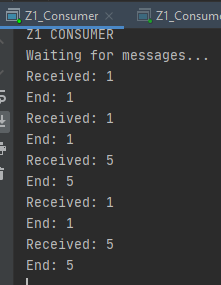
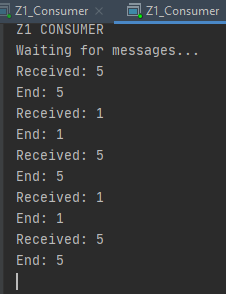
* Co się stanie, jeśli nie będziemy potwierdzać wiadomości ani po otrzymaniu, ani po przetworzeniu?



Pozostaje w stanie unacked do momentu akceptacji lub odrzucenia(np. wyłączenia consumera, wtedy przechodzi ponownie do stanu ready)

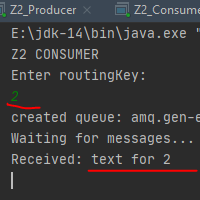
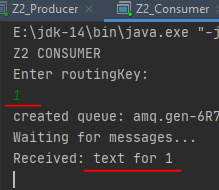
**Zad 1b (1 punkt). Zaobserwuj działanie mechanizmu load-balancing**

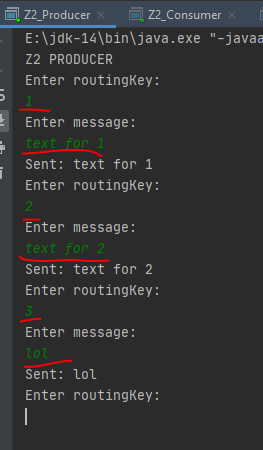
* Round-robin:  
   
* Load-balancing:

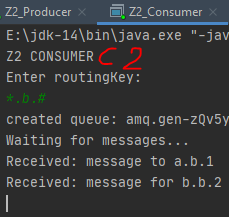
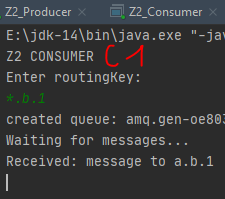
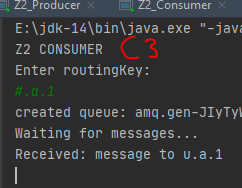
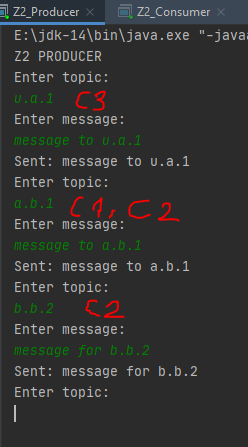
**Zad 2 (2 punkty). Zaimplementuj oraz pokaż w działaniu routing Direct oraz Topic**

* + routingu Direct:



* + routingu Topic:

Topic ma taką przewagę nad direct, że jeśli chcemy wysyłać wiadomości do dwóch consumerów przez direct to musimy im nadać ten sam klucz, co oznacza że nie będziemy w stanie wysłać wiadomości tylko do jednego z nich. (Topic pozwala tworzyć grupy odbiorców)