1. a Czy konsument zakończy obsługę wiadomości?

Potwierdzenie	Restart po przetworzeniu wiadomości	Restart w trakcie przetwarzania wiadomości
Po otrzymaniu wiadomości	TAK	TAK
Po przetworzeniu wiadomości	TAK	NIE

Który sposób potwierdzeń zapewnia większą niezawodność?

Większą niezawodność zapewnia oczywiście wysyłanie potwierdzeń dopiero po przetworzeniu wiadomości.

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

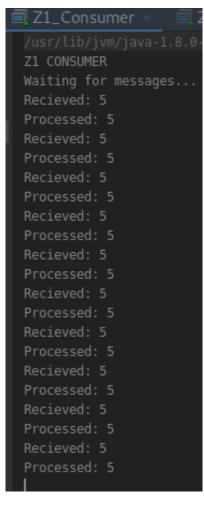
Będziemy przy każym ponownym uruchomieniu konsumenta dostawać wszystkie wiadomości, jakie zostały wysłane przez Producenta i nie zostały potwierdzone.

1. b

Output widoczny na screenshotach pojawił się po dwukrotnym uruchomieniu producenta (i przed poprawą literówki w słowie *Received*)

Przed włączeniem obsługi QoS:





Po włączeniu obsługi QoS:

```
Z1_Consumer × Z1_0

/usr/lib/jvm/java-1.8.0-ope
Z1 CONSUMER

Waiting for messages...

Recieved: 5

Processed: 5

Recieved: 5

Processed: 5

Recieved: 1

Processed: 1

Recieved: 5

Processed: 5

Recieved: 5

Processed: 5

Recieved: 1

Processed: 1

Recieved: 1

Processed: 1

Recieved: 1

Processed: 5

Recieved: 1

Processed: 5

Recieved: 5

Processed: 5

Recieved: 5

Processed: 5

Recieved: 5

Processed: 5

Recieved: 5

Processed: 5

Recieved: 1

Processed: 5
```

```
/usr/lib/jvm/java-1.8.0-or
Z1 CONSUMER
Waiting for messages...
Recieved: 1
Processed: 1
Recieved: 5
Processed: 5
Recieved: 1
Processed: 5
Recieved: 1
Processed: 5
Recieved: 1
Processed: 5
Recieved: 1
Processed: 5
Recieved: 5
Processed: 5
```

Rozłożenie wiadomości jest bardziej równomierne w tym sensie, że przez większość czasu obydwaj konsumenci pracują, a nie tylko jeden. I dostają wiadomości w różnej kolejności (każdy dostaje nową wiadomość dopiero gdy jest wolny), a nie w takiej: i-ty worker dostaje i-tą wiadomość.

2. Przykład 1: W przypadku routingu Direct wiadomość dochodzi tylko do konkretnego odbiorcy:

```
/usr/lib/jvm/java-1.8.0-openjdk/bin/

Z2 PRODUCER
Enter [routingKey, message]:
black, heeeeeeeeej
Sent: heeeeeeeeej
Enter [routingKey, message]:
Waiting for messages...
Waiting for messages...
Received: heeeeeeeej
```

Dziedzic Monika 296551

A w przypadku routingu Topic do tego, do którego pasuje jakikolwiek regex z **nazwą klucza**:

```
/usr/lib/jvm/java-1.8.0-openji
Z2 PRODUCER
Enter [routingKey, message]:
black.orange, witam
Sent: witam
Enter [routingKey, message]:
Waiting for messages...
Received: witam

Received: witam

7usr/lib/jvm/java-1.8.0-openji
Z2 CONSUMER
Enter routingKey:
Enter routingKey:
**.orange*
created queue: amq.gen-Do5k*
Waiting for messages...
Received: witam
Received: witam
```

Przykład 2: W routingu Direct jeśli chce się wysłać wiadomość do paru konsumentów, trzeba ich powiązać z tą samą nazwą kolejki:

```
Z2 PRODUCER

Enter [routingKey, message]:

black wiadomosc

Sent: wiadomosc

Enter [routingKey, message]:

Waiting for messages...

Received: wiadomosc

Received: wiadomosc

Z2 CONSUMER

Enter routingKey:

black

created queue: amq.gen-SE

Waiting for messages...

Received: wiadomosc
```

A w routingu Topic wystarczy wpasować się w regexa po stronie producenta:

```
Z2 CONSUMER
Enter [routingKey, message]:
black.orange, witam
Sent: witam
Enter [routingKey, message]:
Received: witam

Z2 CONSUMER
Enter routingKey:
Black.#

created queue: amq.gen-Do5k
Waiting for messages...
Received: witam

Received: witam
```