

# Контест 9

11 янв 2022, 18:20:51

старт: 23 фев 2021, 12:30:00

финиш: 1 мар 2021, 23:59:59

длительность: 6д. 11ч.

начало: 23 фев 2021, 12:30:00

конец: 1 мар 2021, 23:59:59

## I. Kafka

Ограничение времени	10 секунд
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Kafka - это сервис для работы с сообщениями. На сервис поступают сообщения, а пользователи сервиса могут их считывать отдельными наборами (Батчами). В этой задаче вам предлагается реализовать данный сервис.

Необходимо реализовать следующие классы и члены классов:

- Класс `KafkaException` - наследник класса `Exception`
- Класс `User` - пользователь сервиса
  1. Конструктор с одним параметром - именем пользователя
  2. метод `IncreaseIndex(int count)` - сдвинуть указатель пользователя в очереди сообщений на *count* сообщений вправо
- Класс `Message` - сообщение, обрабатываемое пользователем
  1. Конструктор с двумя параметрами - текстом сообщения и временем создания
  2. Предопределенный метод `ToString()`, возвращающий строку в формате дата и время текст сообщения
- Класс `MessageQueue` - очередь сообщений в сервисе
  1. Конструктор с одним параметром - размером очереди
  2. метод `Push(Message message)` - добавить в очередь сообщение. Если в очередь оказалась переполнена, необходимо выбросить исключение `KafkaException` с сообщением "Queue is out of storage"
  3. Свойство `Size` - количество элементов в очереди
  4. Индексатор с одним целочисленным параметром *index* - сообщение под порядковым номером *index*
  5. Индексатор с двумя целочисленными параметрами *from*, *to* - список сообщений между номерами *from* и *to*.
- Класс `Kafka` - основной класс сервиса
  1. Конструктор с одним целочисленным параметром - размером очереди сообщений
  2. метод `Subscribe(User user)` - подписать пользователя на сервис. Если пользователь уже был подписан, выбросить исключение `KafkaException` с сообщением "User is already subscribed"
  3. метод `Unsubscribe(User user)` - отписать пользователя от сервиса. Если пользователь не подписан, выбросить исключение `KafkaException` с сообщением "User is already unsubscribed"
  4. метод `Push(Message message)` - добавить сообщение в очередь. Если сервис не активен, выбросить исключение `KafkaException` с сообщением "Kafka is not active"
  5. метод `PopMessages(User user, int count)` - передать пользователю *user* *count* сообщений. Если пользователь не подписан на сервис, выкинуть `KafkaException` с сообщением "User is not subscribed", если в очереди недостаточно сообщений, выкинуть `KafkaException` с сообщением "Not enough messages". Если обе ошибки произошли одновременно, необходимо выводить вторую.
  6. методы `Activate` и `Deactivate` - активировать и отключить сервис соответственно
  7. Примечание: При совершении любой операции кроме активации сервиса, если сервис еще не активирован, необходимо выбросить `KafkaException` с сообщением "Kafka is not active".

## Формат ввода

Сначала на вход подается целое положительное число *n* - число команд.

Далее на вход подается целое положительное число *m* - размер очереди сервиса.

Далее на вход подается *n* команд формата <команда> <аргументы>

Возможные команды в задаче:

- activate - активировать сервис
- deactivate - деактивировать сервис
- new\_user - создать пользователя (аргументы: имя пользователя)
- subscribe - подписать пользователя (аргументы: имя пользователя, гарантируется, что подписываемый пользователь уже создан)
- unsubscribe - отписать пользователя (аргументы: имя пользователя, гарантируется, что отписываемый пользователь уже создан)
- push - добавить сообщение в очередь (аргументы: текст сообщения и время создания)
- pop - достать сообщения из очереди (аргументы: имя пользователя и количество сообщений, гарантируется, что пользователь уже создан)

## Формат вывода

Программа должна выводить отчет о каждой команде.

- activate - вывести сообщение "Kafka is activated"
- deactivate - вывести сообщение "Kafka is deactivated"
- new\_user - вывести сообщение "User <name> is created"
- subscribe - вывести сообщение "User <name> subscribed"
- unsubscribe - вывести сообщение "User <name> unsubscribed"
- push - вывести сообщение "Message from <datetime> <text> pushed"
- pop - вывести сообщение "User <name> popped <count> messages" Далее с новой строки для каждого сообщения необходимо вывести время сообщения и текст сообщения

В случае, если выбрасывается исключение типа KafkaException, необходимо выводить сообщение исключения.

## Пример

Ввод

Вывод

12	Kafka is activated
3	User user1 is created
activate	User user2 is created
new_user user1	User user1 subscribed
new_user user2	User is already subscribed
subscribe user1	Message from 04-03-2000T15:45 message1 pushed
subscribe user1	Message from 04-03-2000T15:46 message2 pushed
push message1 04-03-2000T15:45	Message from 04-03-2000T15:47 message3 pushed
push message2 04-03-2000T15:46	User user1 popped 3 messages
push message3 04-03-2000T15:47	[04-03-2000T15:45] message1
pop user1 3	[04-03-2000T15:46] message2
pop user2 2	[04-03-2000T15:47] message3
pop user1 2	User is not subscribed
subscribe user2	Not enough messages
	User user2 subscribed

## Примечания

В систему необходимо сдавать .zip архив, содержащий **только** файла **"Kafka.cs"**, **"KafkaException.cs"**, **"Message.cs"**, **"MessageQueue.cs"**, **"User.cs"**

Набрать здесь

Отправить файл

```
1 PK...uZR...Kafka.csUJ1}. F...VATT...ctw...I...ws...P...&dI...E...近I...|
2 5...A...mc...BR...:t...f...6...yA.v... &...}...w...^)...%G+...L...v@...d...X... ^...99...B...0...a...Z...=...%E...
3 v%...cJe...0...<...<...{I n{u...Y...D...o,...YY...0...R...
4 J6..._
5 2...i9...
6 ...k...4...@...-...2...-.[...q...$mB=... An...~<...F...g*k...z...0x...xP...|4...e...g?
7 y...k...4...1...*...:...>>...PK...vZR]n...Y...r...KafkaException.cs]1...{r...},...w$...X...r
8 %...I+...PK...wR...4...5...
9 ...Message.csmM
10 0...9...P\T.../P<...v'.j;...@...L...rw...Z:...0...o4r...M...]&I ...jk"...Q\·0·JN%#·+_a|q...n/d...s... yopJ...E
11 M:·Q...C&...t...q...&e...3...Z...H ...#4>E...J...&[...d...6X·0...y...f ...kG...{q... 'a... 8<...T...d...f... (B...Fe
12 A ...~ ...M...Mx...o...Z^...t... 8...V...[K...J...U...r...~...0...H5...3>...m...c/...T...v... PK...sZR...K...t...
13 0...D...>...)^ ...]...uD*H...g%... [...b...3... _j & z 53<...Z...*{...w...Pj...Z8
14 !Y...z... [...K...R...&...%...: +...Gjc...P3...DU...fFo...+...y...
15 ?PK...?...uZR...}...$...Kafka.cs
16 ...4...2 ...4...y... PK...?...vZR]n...Y...r...$...KafkaExcepti
17 ...5...9{...5... PK...?...wR...4...5...
18 $...Message.cs
19 ...]/...
20 2 ...4... PK...?...vZRs3...fH... $...5...MessageQueue.cs
21 ...&...5... PK...?...sZR...K...t... $...User.cs
22 b...2...U...4... PK...?..._...
```