**Ресурсы:**

* Ресурсов фиксированное число, у каждого ресурса есть шанс поломки и время, которое он будет чиниться. Они генерируются случайно.
* При поломке запрос переходит в очередь на этот ресурс
* Замены ресурсов при поломке не происходит
* У ресурса три состояния: свободен (FREE), занят (BUSY), сломан (BROKEN)
* У каждого ресурса есть свой таймер
* Если состояние ресурса FREE или BUSY, то таймер срабатывает каждую секунду. Тогда генерируется случайное число и проверяется, сломается ли ресурс.
* Если ресурс сломался, то ресурс переходит в состояние BROKEN, а таймер сработает через время починки этого ресурса. Тогда ресурс перейдёт в состояние FREE
* При переходе ресурса в состояние FREE проверяется очередь в этот ресурс. Если в ней есть запросы, то первый из них занимает ресурс, он переходит в состояние BUSY
* Очереди на ресурсы реализованы по принципу FIFO
* Запрос, занимающий ресурс в настоящий момент, хранится как самый первый в очереди на этот ресурс
* Запрос, занимающий ресурс, занимает его до того момента, пока пользователь его не отзовёт

**Запросы:**

* У каждого запроса есть параметры: уникальный id; ресурс, на который он претендует; максимальное время, которое запрос может ждать в очереди
* Реализована общая очередь для всех запросов ExpirationQueue. В ней хранится, сколько времени запрос ещё может ждать
* При истечении времени ожидания (время ожидания = максимальному времени, которое запрос может ждать) уведомляется пользователь
* После истечения времени ожидания пользователь или отзывает запрос, или продлевает его время ожидания
* Запросы в очередях на ресурсы и в ExpirationQueue соотносятся друг с другом с помощью id запроса (и ресурса, на который он претендует, чтобы его можно было легче найти)

**Хранение модели в файле**

Структура файла:

Первая строка – ExpirationQueue. Первым идёт число – количество запросов в очереди. Через пробел записываются данные о самих запросах в порядке: сколько времени запрос ещё может ждать; id запроса; ресурс, на который запрос претендует; максимальное время, которое запрос может ждать.

Далее каждая строка хранит информацию об одном ресурсе. Формат (разделитель – пробел):

состояние ресурса (FREE, BUSY, BROKEN); вероятность поломки; время починки ресурса; число запросов в очереди на ресурс; информация о запросах в очереди в порядке их нахождения в очереди: id запроса, id ресурса, на который он претендует, максимальное время, которое запрос может ждать.

**Проблемы многопоточности**

* Перед любым запросом данных от пользователя (продление истёкшего запроса, запрос номера ресурса) ставится спинлок на все прочие вызовы ввода данных от пользователя. (Спинлок реализован как булевый флаг остановки внутри цикла)
* Во время любого запроса ввода от пользователя время «замирает», то есть все таймеры ставятся на паузу.