# 高并发中常见的限流方式

## 本文内容

1. 介绍常见的限流算法
2. 通过控制最大并发数来进行限流
3. 通过漏桶算法来进行限流
4. 通过令牌桶算法来进行限流
5. 限流工具类RateLimiter

## 常见的限流的场景

1. 秒杀活动，数量有限，访问量巨大，为了防止系统宕机，需要做限流处理
2. 国庆期间，一般的旅游景点人口太多，采用排队方式做限流处理
3. 医院看病通过发放排队号的方式来做限流处理。

## 常见的限流算法

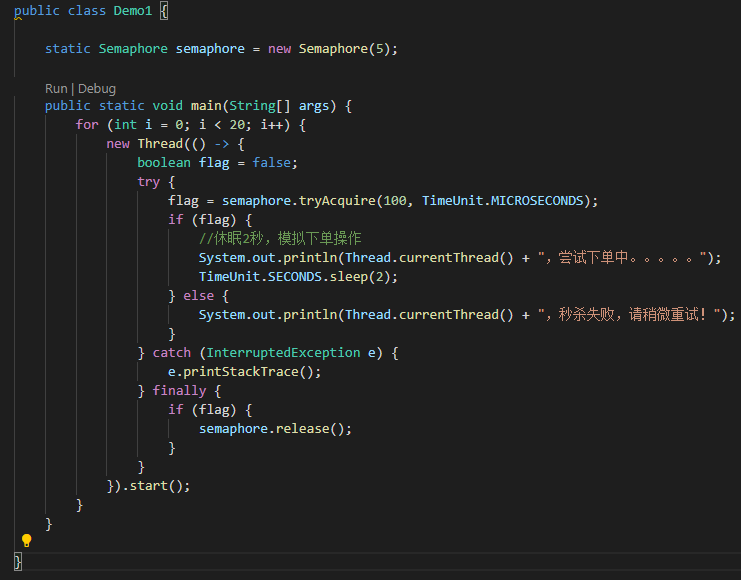
1. 通过控制最大并发数来进行限流
2. 使用漏桶算法来进行限流
3. 使用令牌桶算法来进行限流

## 通过控制最大并发数来进行限流

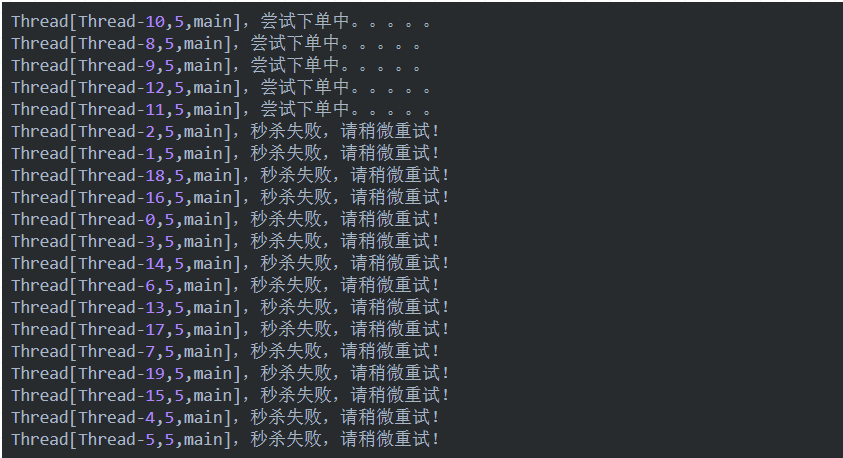
以秒杀业务为例，10个iphone，100万人抢购，100万人同时发起请求，最终能够抢到的人也就是前面几个人，后面的基本上都没有希望了，那么我们可以通过控制并发数来实现，比如并发数控制在10个，其他超过并发数的请求全部拒绝，提示：秒杀失败，请稍后重试。

并发控制的，通俗解释：一大波人去商场购物，必须经过一个门口，门口有个门卫，兜里面有指定数量的门禁卡，来的人先去门卫那边拿取门禁卡，拿到卡的人才可以刷卡进入商场，拿不到的可以继续等待。进去的人出来之后会把卡归还给门卫，门卫可以把归还来的卡继续发放给其他排队的顾客使用。

JUC中提供了这样的工具类：Semaphore，示例代码：



输出：



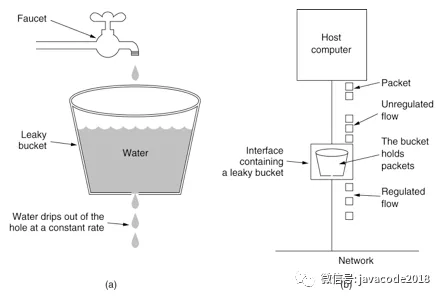
关于Semaphore的使用，可以移步：[JUC中的Semaphore（信号量）](https://mp.weixin.qq.com/s?__biz=MzA5MTkxMDQ4MQ==&mid=2648933130&idx=1&sn=cecc6bd906e79a86510c1fbb0e66cd21&chksm=88621b34bf159222042da8ed4b633e94ca04a614d290d54a952a668459a339ebec0c754d562d&token=702505185&lang=zh_CN&scene=21#wechat_redirect)

## 使用漏桶算法来进行限流

国庆期间比较火爆的景点，人流量巨大，一般入口处会有限流的弯道，让游客进去进行排队，排在前面的人，每隔一段时间会放一拨进入景区。排队人数超过了指定的限制，后面再来的人会被告知今天已经游客量已经达到峰值，会被拒绝排队，让其明天或者以后再来，这种玩法采用漏桶限流的方式。

漏桶算法思路很简单，水（请求）先进入到漏桶里，漏桶以一定的速度出水，当水流入速度过大会直接溢出，可以看出**漏桶算法能强行限制数据的传输速率。**

漏桶算法示意图：



简陋版的实现，代码如下：







代码中BucketLimit.build(10, 60, TimeUnit.MINUTES);创建了一个容量为10，流水为60/分钟的漏桶。

代码中用到的技术有：

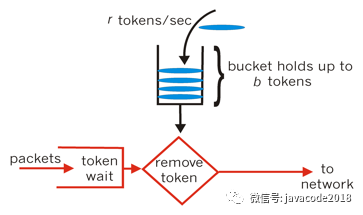
1、[BlockingQueue阻塞队列](https://mp.weixin.qq.com/s?__biz=MzA5MTkxMDQ4MQ==&mid=2648933190&idx=1&sn=916f539cb1e695948169a358549227d3&chksm=88621b78bf15926e0a94e50a43651dab0ceb14a1fb6b1d8b9b75e38c6d8ac908e31dd2131ded&token=1963100670&lang=zh_CN&scene=21#wechat_redirect)

2、[JUC中的LockSupport工具类，必备技能](https://mp.weixin.qq.com/s?__biz=MzA5MTkxMDQ4MQ==&mid=2648933125&idx=1&sn=382528aeb341727bafb02bb784ff3d4f&chksm=88621b3bbf15922d93bfba11d700724f1e59ef8a74f44adb7e131a4c3d1465f0dc539297f7f3&token=1338873010&lang=zh_CN&scene=21#wechat_redirect)

## 使用令牌桶算法来进行限流

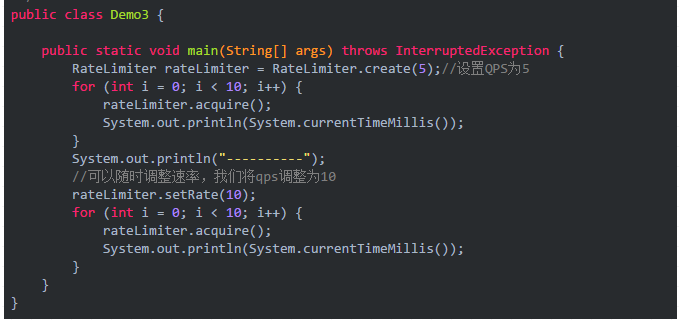
令牌桶算法的原理是系统以恒定的速率产生令牌，然后把令牌放到令牌桶中，令牌桶有一个容量，当令牌桶满了的时候，再向其中放令牌，那么多余的令牌会被丢弃；当想要处理一个请求的时候，需要从令牌桶中取出一个令牌，如果此时令牌桶中没有令牌，那么则拒绝该请求。从原理上看，令牌桶算法和漏桶算法是相反的，一个“进水”，一个是“漏水”。这种算法可以应对突发程度的请求，因此比漏桶算法好。

令牌桶算法示意图：



## 限流工具类RateLimiter

Google开源工具包Guava提供了限流工具类RateLimiter，可以非常方便的控制系统每秒吞吐量，示例代码如下：



输出：



代码中RateLimiter.create(5)创建QPS为5的限流对象，后面又调用rateLimiter.setRate(10);将速率设为10，输出中分2段，第一段每次输出相隔200毫秒，第二段每次输出相隔100毫秒，可以非常精准的控制系统的QPS。