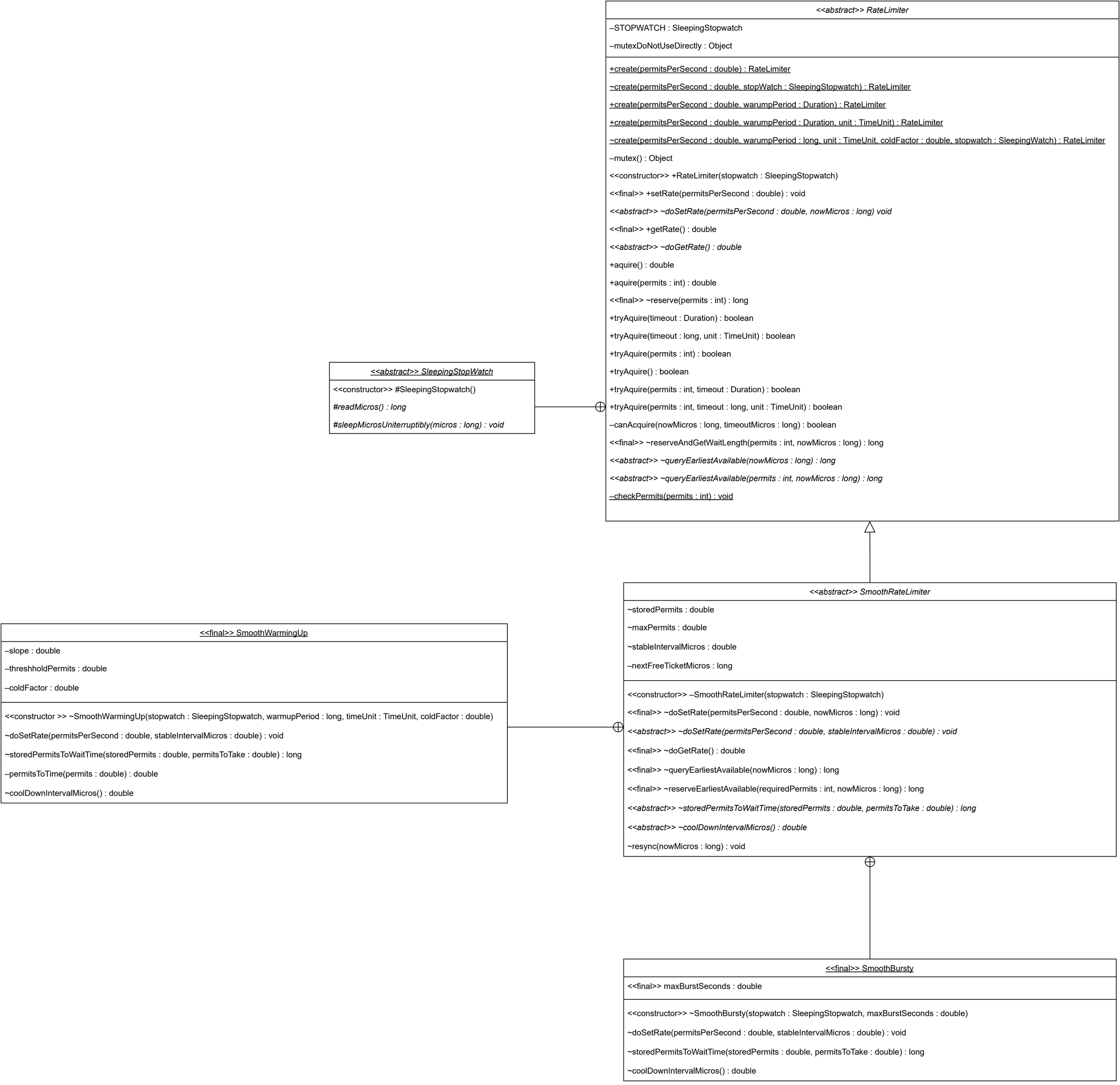
<<abstract>> RateLimiter
–STOPWATCH : SleepingStopwatch
–mutexDoNotUseDirectly : Object
+create(permitsPerSecond : double) : RateLimiter
~create(permitsPerSecond : double, stopWatch : SleepingStopwatch) : RateLimiter
+create(permitsPerSecond : double, warumpPeriod : Duration) : RateLimiter
+create(permitsPerSecond : double, warumpPeriod : Duration, unit : TimeUnit) : RateLimiter
~create(permitsPerSecond : double, warumpPeriod : long, unit : TimeUnit, coldFactor : double, stopwatch : SleepingWatch) : RateLimiter
–mutex() : Object
<<constructor>> +RateLimiter(stopwatch : SleepingStopwatch)
<<final>> +setRate(permitsPerSecond : double) : void
<<abstract>> ~doSetRate(permitsPerSecond : double, nowMicros : long) void
<<final>> +getRate() : double
<<abstract>> ~doGetRate() : double
+aquire() : double
+aquire(permits : int) : double
<<final>> ~reserve(permits : int) : long
+tryAquire(timeout : Duration) : boolean
+tryAquire(timeout : long, unit : TimeUnit) : boolean
+tryAquire(permits : int) : boolean
+tryAquire() : boolean
+tryAquire(permits : int, timeout : Duration) : boolean
+tryAquire(permits : int, timeout : long, unit : TimeUnit) : boolean
–canAcquire(nowMicros : long, timeoutMicros : long) : boolean
<<final>> ~reserveAndGetWaitLength(permits : int, nowMicros : long) : long
<<abstract>> ~queryEarliestAvailable(nowMicros : long) : long
<<abstract>> ~queryEarliestAvailable(permits : int, nowMicros : long) : long
–checkPermits(permits : int) : void
<<abstract>> SleepingStopWatch
<<constructor>> #SleepingStopwatch()
#readMicros() : long
#sleepMicrosUniterruptibly(micros : long) : void
<<abstract>> SmoothRateLimiter
~storedPermits : double
~maxPermits : double
~stableIntervalMicros : double
–nextFreeTicketMicros : long
<<constructor>> –SmoothRateLimiter(stopwatch : SleepingStopwatch)
<<final>> ~doSetRate(permitsPerSecond : double, nowMicros : long) : void
<<abstract>> ~doSetRate(permitsPerSecond : double, stableIntervalMicros : double) : void
<<final>> ~doGetRate() : double
<<final>> ~queryEarliestAvailable(nowMicros : long) : long
<<final>> ~reserveEarliestAvailable(requiredPermits : int, nowMicros : long) : long
<<abstract>> ~storedPermitsToWaitTime(storedPermits : double, permitsToTake : double) : long
<<abstract>> ~coolDownIntervalMicros() : double
~resync(nowMicros : long) : void
<<final>> SmoothWarmingUp
–slope : double
–threshholdPermits : double
–coldFactor : double
<<constructor >> ~SmoothWarmingUp(stopwatch : SleepingStopwatch, warmupPeriod : long, timeUnit : TimeUnit, coldFactor : double)
~doSetRate(permitsPerSecond : double, stableIntervalMicros : double) : void
~storedPermitsToWaitTime(storedPermits : double, permitsToTake : double) : long
–permitsToTime(permits : double) : double
~coolDownIntervalMicros() : double
<<final>> SmoothBursty
<<final>> maxBurstSeconds : double
<<constructor>> ~SmoothBursty(stopwatch : SleepingStopwatch, maxBurstSeconds : double)
~doSetRate(permitsPerSecond : double, stableIntervalMicros : double) : void
~storedPermitsToWaitTime(storedPermits : double, permitsToTake : double) : long
~coolDownIntervalMicros() : double