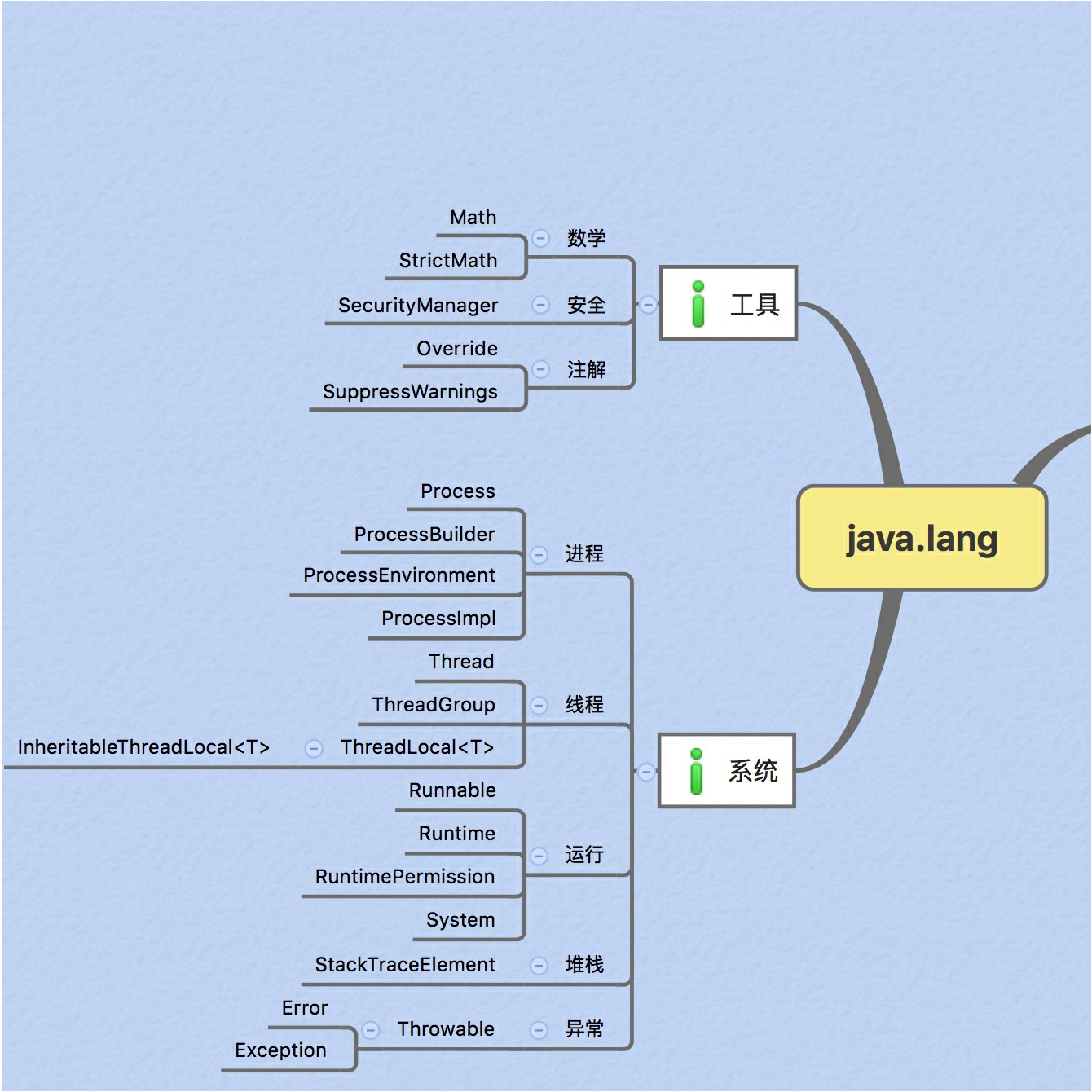


Java千百问_09基础类库（001）_lang包有什么功能

[点击进入_更多_Java千百问-基础类库](#)

1、lang包有什么功能

java.lang包是java语言的**核心**，它提供了java中的基础类。包括基本Object类、Class类、String类、基本类型的包装类、基本的数学类等等最基本的类。我们介绍一下Java 8中的java.lang包。主要类如下图：



大体将java.lang包分为三类：**类型**、**工具**、**系统**，下面分别介绍其中比较常用的类：

类型

1. **对象基类**
Object，是java.lang的根类，也是所有类的超类。
2. **类**
Class，用来表示类和接口的类型。Class对象在类加载时由JVM调用类加载器中的defineClass方法自动构造。

ClassLoader, 负责加载类。

Compiler, 作为编译器的占位符, 它不做任何事情, 仅用来支持Java到本机代码的编译器及相关服务。

3. 基本类型

基本类型的**包装类**, 包括Boolean、Character、Byte、Short、Integer、Long、Float、Double, 其中数值类型均即成**Number类**。了解包装器看这里: [什么是基本类型包装器](#)
String, 字符串类。了解字符串看这里: [String是什么](#)

4. 字符序列

StringBuffer、**StringBuilder**, 可变的字符序列。具体区别看这里: [StringBuffer/StringBuilder有什么区别](#)

5. 枚举

Enum, 是所有枚举类型的公共基类。

6. 包

Package, 包含了有关Java包 (package) 的信息。

7. 无类型

Void, 标示关键字void的Class对象的引用, 不可被实例化。

8. 迭代器

Iterable, 可迭代接口, 实现接口可以使用迭代器进行对象遍历。

工具

1. 数学

Math、**StrictMath**, 提供了基本的数字操作, 如指数、对数、平方根和三角函数。一般情况下, Math调用StrictMath的方法来完成实现。java中还有一个java.math包, 这个包主要提供用于执行任意精度整数算法 (BigInteger) 和任意精度小数算法 (BigDecimal) 的类。

2. 安全

SecurityManager, 允许应用程序实现安全策略的类。

3. 注解

Override, 标记类中方法是实现/重写父类的方法。

SuppressWarnings, 取消对被标记的元素的警告。了解Java注解看这里: [注解是什么系统](#)

4. 进程

Process, 进程抽象类。

ProcessBuilder, 用于创建操作系统进程。

ProcessEnvironment, 进程的运行环境参数。

ProcessImpl, 进行接口的实现类。

5. 线程

Thread, 进程中的执行线程。

ThreadGroup, 线程组, 表示一个线程的集合。它构成一个树状结构, 可以包含其他线程组, 除了根节点的线程组, 每个线程组都具有父线程组。

ThreadLocal, 提供线程的变量。

6. 运行

Runnable, 可运行接口, 所有Thread都应实现它。

Runtime, 运行时类, 将应用程序与其运行的环境相关联。

RuntimePermission, 用于运行时权限。

System, 系统级的很多属性和控制方法都放置在该类的内部。

7. 堆栈

StackTraceElement, 堆栈跟踪中的元素, 它的每个实例都表示单独的一个栈帧 (表示一个方法调用)。

8. 异常

Throwable, 异常基类, Java中所有异常都继承于它。了解更多异常看这里: [什么是java中的异常](#)

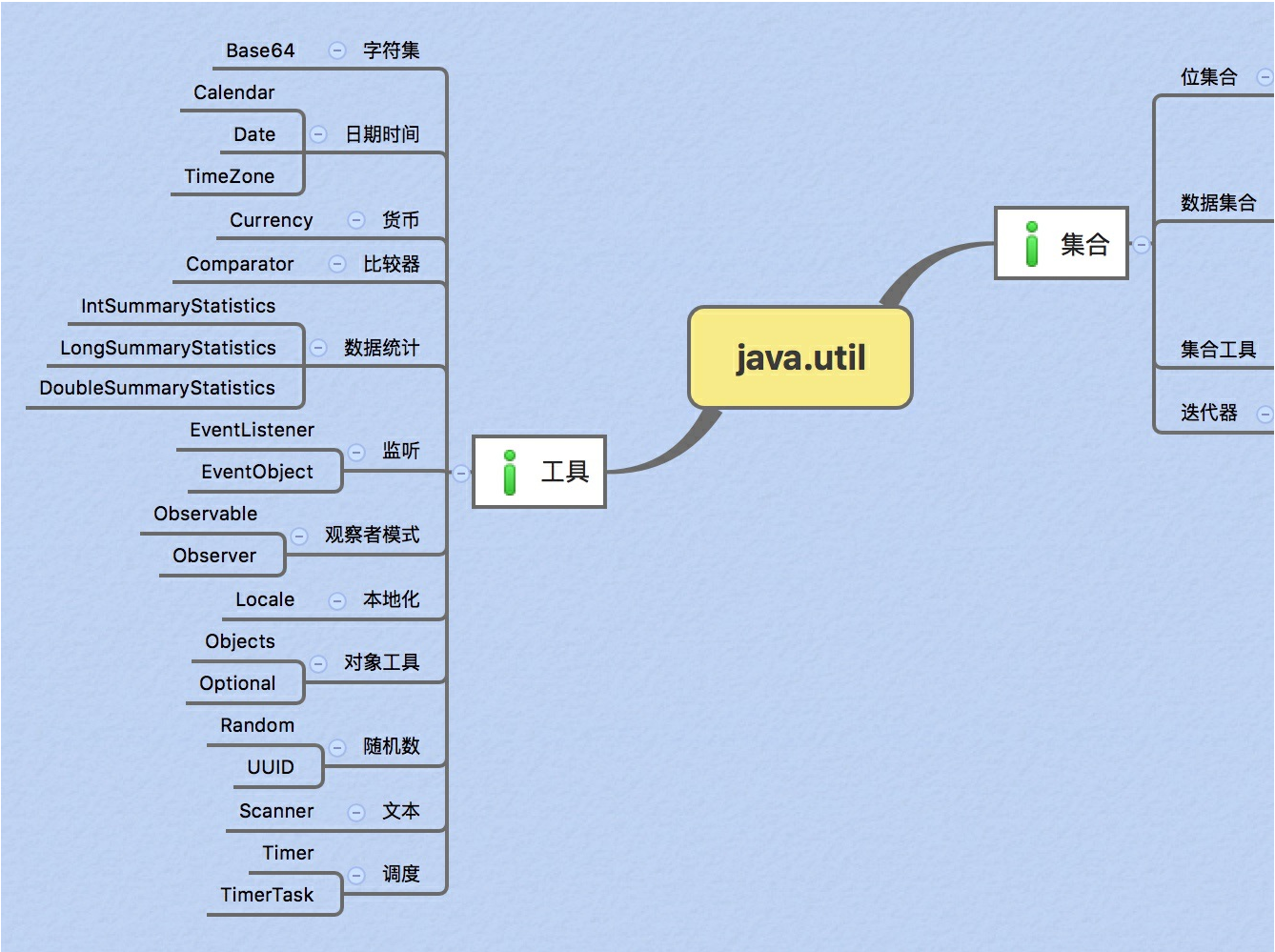
]

Java千百问_09基础类库（002）_util包有什么功能

[点击进入_更多_Java千百问-基础类库](#)

1、util包有什么功能

java.util包提供了java中的基础工具类。包括基本的集合框架等基础类以及通用的工具类。主要类如下图：



大体将java.util包分为两类：集合、工具，下面分别介绍其中比较常用的类：

集合

1. 位集合

BitSet，只存储0、1的集合。

2. 数据集合

Collection，常用集合容器，包括列表List、数据集合Set、队列Queue。

Map，键值对容器。

Dictionary，字典容器，和Map非常类似。

Java提供的若干数据集合容器是在开发过程中经常使用的，不同容器具有自己的特性，应用于不同的场景。了解集合容器看这里：[什么是集合容器][2]

3. 集合工具

Collections，为集合容器提供常用的工具。

Arrays，为数组提供常用的工具。

4. 迭代器

Iterator，迭代器，使用迭代器可以对对象进行遍历。

工具

1. **字符集**
Base64，提供对Base64的编解码工具。
2. **日期时间**
Date，提供对日期、时间的操作和处理，以及获取当前系统的时间。
Calendar，比Date更加强大的日期处理工具，可以实现国际化，官方建议使用的日期工具。
TimeZone，描述时区，为Calendar提供国际化基础。
3. **货币**
Currency，描述各国的货币。
4. **比较器**
Comparator，可以通过实现Comparator接口，对集合对象或数组对象进行自定义排序。
5. **数据统计**
IntSummaryStatistics、**LongSummaryStatistics**、**DoubleSummaryStatistics**，Java 8中新特性，主要用于统计数值型数组中元素的常见统计值，例如：最大值，最小值，平均值，总和等。常与Stream一起使用。
6. **监听**
EventListener，事件监听器，想使用事件监听需要实现它。
EventObject，事件对象，包含事件对应的数据，需要自定义类继承它。
7. **观察者模式**
Observable，被观察对象接口，需要被观察的对象实现该接口。
Observer，观察者需要实现的接口，当Observable对象发生变化时，Observer得到通知，就可以进行相应的工作。
8. **本地化**
Locale，描述了特定的地理、政治和文化地区，通过它来实现Java国际化。
9. **对象工具**
Objects，为Java对象提供常用工具方法，例如是否相等、比较大小等。
Optional，可以承载为null的容器对象，通过返回该对象可以避免null值带来的问题。
10. **随机数**
Random，用来产生随机数，原理是通过种子数来生成伪随机数。
UUID，可以生成128位长、且不重复的数字。它结合机器的网卡、当地时间、随机数来生成GUID。从理论上讲，如果一台机器每秒产生10000000个GUID，则可以保证（概率意义上）3240年不重复。
11. **文本**
Scanner，主要功能是简化文本扫描，其他功能例如检索文本、对字符串和基本类型进行分析。
12. **调度**
Timer，定时器工具，在主线程之外启动一个单独的线程，来执行指定的计划任务，可以指定执行一次或者反复执行多次。
TimerTask，一个可以被Timer执行的任务，实现了Runnable接口。

]

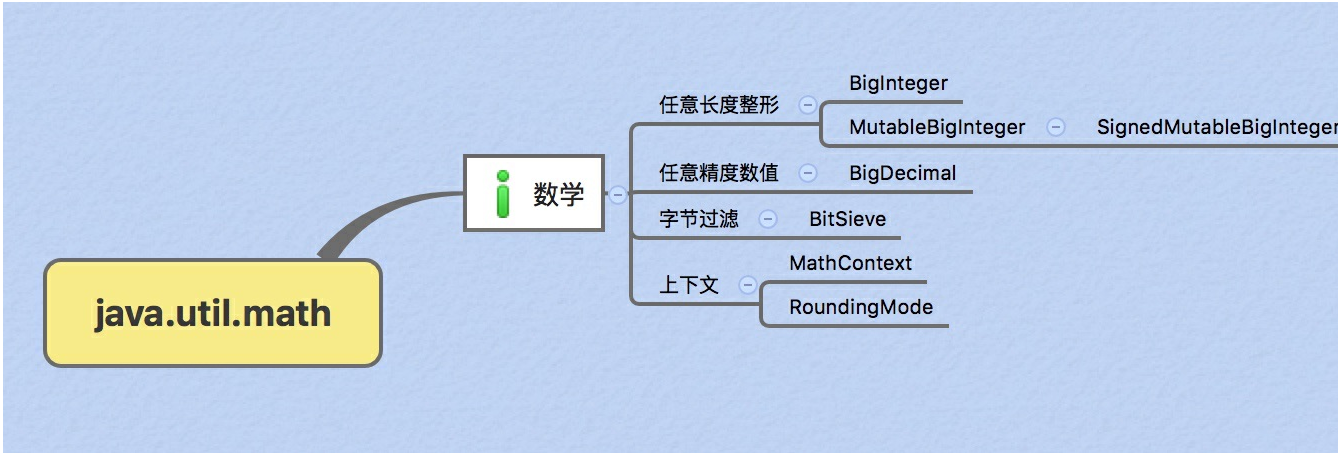
[

Java千百问_09基础类库（003）_java.math包有什么功能

,
[点击进入_更多_Java千百问-基础类库](#)

1、java.math包有什么功能

java.math包提供了java中的数学类。包括基本的浮点库、复杂运算以及任意精度的数据运算。主要类如下图：



下面介绍math中的类：

- 任意长度整形**
BigInteger，任意长度不可变整形。由于基本类型的整形长度有限，所以java提供了任意长度任意精度的整形。提供了种运算方法。继承于Number。
MutableBigInteger，可变的BigInteger。
SignedMutableBigInteger，有符号的可变的BigInteger。
- 任意精度数值**
BigDecimal，不可变的、任意精度的有符号数值。由任意精度的整数非标度值和32位的整数标度(scale)组成。提供了种运算方法。继承于Number。
- 字节过滤**
BitSieve，一个简单用于寻找质数的筛选器。
- 上下文**
MathContext，数学操作上下文，描述数字运算符的某些规则。
RoundingMode，舍入算法的枚举。

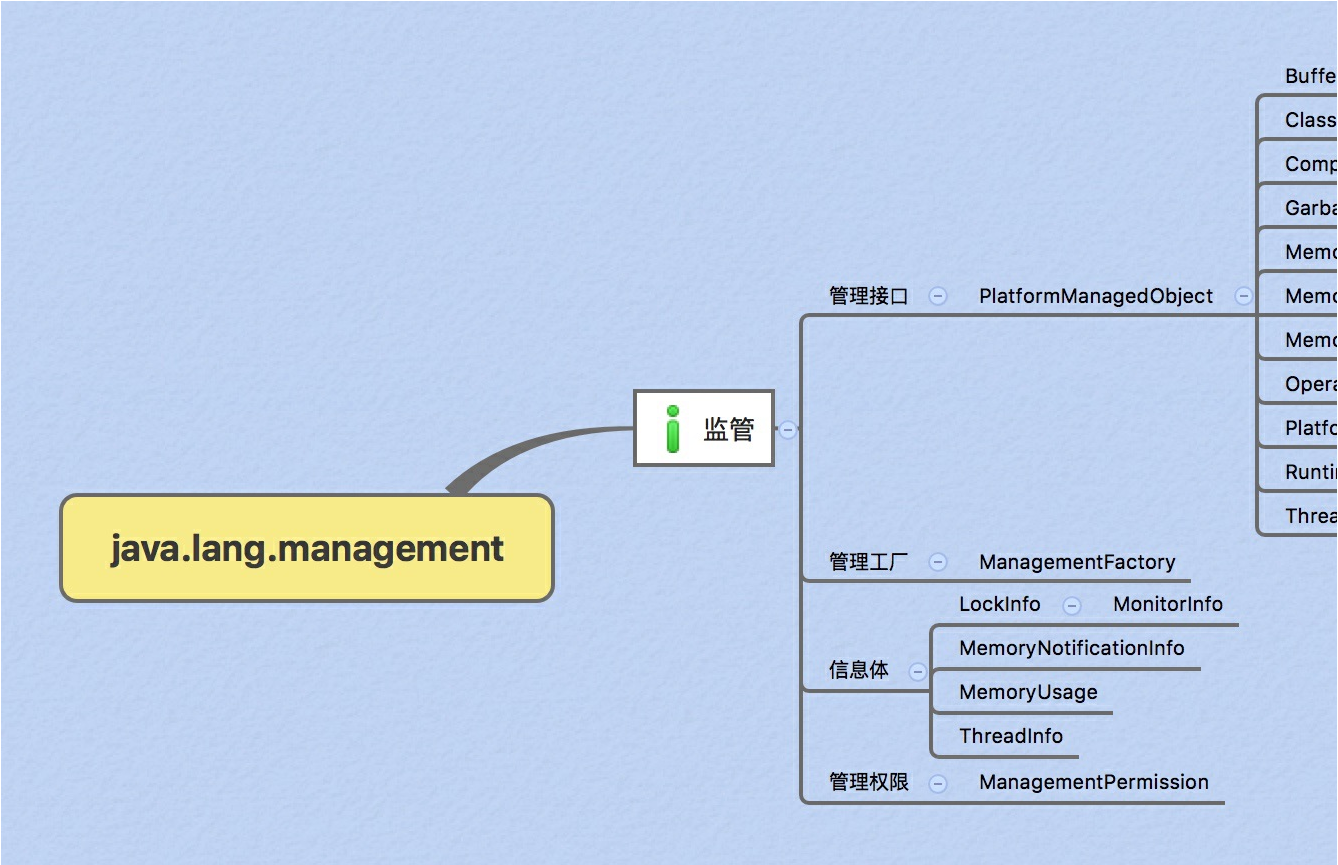
]

Java千百问_09基础类库（004）_java.lang.management包有什么功能

[点击进入_更多_Java千百问-基础类库](#)

1、java.lang.management包有什么功能

java.lang.management包提供了全面的**监控和管理工具**，包括JVM的监管API、监管API日志、jconsole和其他监控工具、out-of-the-box的监管工具（开箱即用）、Java管理扩展平台（JMX）等等。sun在sun.management中提供了对应的实现类。主要类如下图：



下面介绍java.lang.management中的类：

1. 管理接口

- PlatformManagedObject**，平台管理对象的接口，是JMX MXBean监视管理平台的核心组件。每个平台管理对象都要实现这个接口。
- BufferPoolMXBean**，缓冲池管理接口，可以管理和监控缓冲池，direct或mapped两类缓冲池。
- ClassLoadingMXBean**，类加载管理接口，可以管理和监控Java虚拟机的类加载系统。
- CompilationMXBean**，编译管理接口，可以管理和监控Java虚拟机的编译系统。
- GarbageCollectorMXBean**，垃圾回收管理接口，可以管理和监控用于Java虚拟机的垃圾回收。
- MemoryManagerMXBean**，内存/内存池基本信息管理接口，可以管理和监控用于Java虚拟机内存/内存池的基本信息。
- MemoryMXBean**，内存信息管理接口，可以管理和监控Java虚拟机的内存信息。
- MemoryPoolMXBean**，内存池管理接口，可以管理和监控Java虚拟机内存池的使用。
- OperatingSystemMXBean**，操作系统管理接口，可以管理和监控Java虚拟机所运行操作系统的信息。
- PlatformLoggingMXBean**，日志管理接口，可以管理和监控Java虚拟机的日志。
- RuntimeMXBean**，运行时系统管理接口，可以管理和监控Java虚拟机运行的情况。
- ThreadMXBean**，线程管理接口，可以管理和监控Java虚拟机的线程。

2. 管理工厂

- ManagementFactory**，MXBean的工厂类，MXBean是通过工厂模式进行管理的，通过该工厂获取对应的MXBean。

3. 信息体

- LockInfo**、**MonitorInfo**，锁的信息类，LockInfo可以是任何的Java锁（包括简单Java锁和java.util.concurrent包中所使用的锁：AbstractOwnableSynchronizer和Condition的实现类/子类），而MonitorInfo是简单的Java对象所代表的锁。

`MemoryNotificationInfo`，内存通知的信息类。

`MemoryUsage`，内存快照的信息类，通过`MemoryMXBean`、`MemoryPoolMXBean`内存管理接口可以获取到内存快照的信息。

`ThreadInfo`，线程的信息类，通过`ThreadMXBean`线程管理接口可以获取到线程的信息。

4. 管理权限

`ManagementPermission`，权限管理类，一般不会手动创建，而是由安全策略文件创建。

了解`java.lang.management`包使用看这里：[\[java.lang.management包如何使用\]\[2\]](#)

[2]:

]