

## Exercise 10.1

---

- Surf the Internet and write a short paper to compare GC on Java platform with GC on Microsoft .Net platform.

.NET 垃圾回收过程：

.NET 框架包含一个托管堆，所有的.NET 语言在分配引用类型对象时都要使用它。像值类型这样的轻量级对象始终分配在栈中，但是所有的类实例和数组都被生成在一个内存池中，这个内存池就是托管堆。.NET 框架中的垃圾回收器被称为分代的垃圾回收器 (Generational Garbage Collector)，也就是说被分配的对象划分为 3 个类别，或称为“代”，分别为 0, 1, 2。最近被分配内存空间的对象被放置于第 0 代，因为第 0 代很小，小到足以放进处理器的二级 (L2) 缓存，所以第 0 代能够为我们提供对其中对象的快速存取；经过一轮垃圾回收后，仍然保留在第 0 代中的对象被移进第 1 代中；再经过一轮垃圾内存回收后，仍然保留在第 1 代中的对象则被移进第 2 代中。第 2 代包含了生存期较长的对象，这些对象至少经过了两轮回收。垃圾回收器会先检查所有的 0 代对象，如果还需要更多的空间，那么垃圾回收器会继续检查所有的 1 代对象，整理出足够的空间。这时，没有被回收的 1 代对象将成为 2 代对象。分代可以避免每次垃圾回收都遍历整个托管堆，这样可以提高垃圾回收的性能。

Java 垃圾回收过程：

JVM 使用分代收集器，它把堆分为三个主要的域：新域（Young Generation）、旧域（Tenured Generation）以及永久域；JVM 生成的所有新对象放在新域中；一旦对象经历了一定数量的垃圾收集循环后，便获得使用期并进入旧域；