```
package main.java.Test1;

public class GC3 {
    public static void main(String[] args) {
        int size = 1024 * 1024;
        byte[] myAlloc1 = new byte[2 * size];
        byte[] myAlloc2 = new byte[2 * size];
        byte[] myAlloc3 = new byte[2 * size];
        byte[] myAlloc4 = new byte[2 * size];

        System.out.println("hello world");
    }
}
```

```
-verbose:gc  //报告每个垃圾收集事件，输出虚拟机中垃圾回收的详细日志
-Xms20M      //初始化堆空间大小
-Xmx20M      //最大堆空间大小，-Xms和-Xmx设置成一样可以避免垃圾回收造成的抖动问
题
-Xmn10M      //新生代内存大小
-XX:SurvivorRatio=8
-XX:+PrintGCDetails //打印GC回收详细日志
-XX:PretenureSizeThreshold=4194304 //新创建的对象大小大于这个阈值直接在老年代创
建
-XX:+UseSerialGC
-XX:+PrintCommandLineFlags  打印JVM参数
-XX:MaxTenuringThreshold=5  设置到老年代中的最大GC年龄,可自动调节 默认值是15，
CMS中是6,G1中为15(在JVM中,该数值是由4个bit来表示的,所以最大为15)
-XX:+PrintTenuringDistribution  打印对象的年龄
```

```
-XX:InitialHeapSize=20971520 -XX:InitialTenuringThreshold=5 -
XX:MaxHeapSize=20971520 -XX:MaxNewSize=10485760 -
XX:MaxTenuringThreshold=5 -XX:NewSize=10485760 -
XX:PretenureSizeThreshold=4194304 -XX:+PrintCommandLineFlags -XX:+PrintGC -
XX:+PrintGCDetails -XX:+PrintTenuringDistribution -XX:SurvivorRatio=8 -
XX:+UseCompressedClassPointers -XX:+UseCompressedOops -XX:-
UseLargePagesIndividualAllocation -XX:+UseSerialGC
[GC (Allocation Failure) [DefNew
Desired survivor size 524288 bytes, new threshold 1 (max 5)
- age   1:     699304 bytes,     699304 total
: 6443K->682K(9216K), 0.0040592 secs] 6443K->4778K(19456K), 0.0041203 secs]
[Times: user=0.00 sys=0.01, real=0.00 secs]
hello world
Heap
 def new generation   total 9216K, used 5016K [0x00000000fec00000,
0x00000000ff600000, 0x00000000ff600000)
```

```
  eden space 8192K,  52% used [0x00000000fec00000, 0x00000000ff03b710,
0x00000000ff400000)
  from space 1024K,  66% used [0x00000000ff500000, 0x00000000ff5aaba8,
0x00000000ff600000)
  to   space 1024K,   0% used [0x00000000ff400000, 0x00000000ff400000,
0x00000000ff500000)
 tenured generation   total 10240K, used 4096K [0x00000000ff600000,
0x0000000100000000, 0x0000000100000000)
   the space 10240K,  40% used [0x00000000ff600000, 0x00000000ffa00020,
0x00000000ffa00200, 0x0000000100000000)
 Metaspace       used 3500K, capacity 4498K, committed 4864K, reserved 1056768K
  class space    used 387K, capacity 390K, committed 512K, reserved 1048576K
```