# Java 内存溢出（java.lang.OutOfMemoryError）的常见情况和处理方式总结

java.lang.OutOfMemoryError这个错误我相信大部分开发人员都有遇到过，产生该错误的原因大都出于以下原因：JVM内存过小、程序不严密，产生了过多的垃圾。

导致OutOfMemoryError异常的常见原因有以下几种：

1. 内存中加载的数据量过于庞大，如一次从数据库取出过多数据；
2. 集合类中有对对象的引用，使用完后未清空，使得JVM不能回收；
3. 代码中存在死循环或循环产生过多重复的对象实体；
4. 使用的第三方软件中的BUG；
5. 启动参数内存值设定的过小；

此错误常见的错误提示：

1. tomcat:java.lang.OutOfMemoryError: PermGen space
2. tomcat:java.lang.OutOfMemoryError: Java heap space
3. weblogic:Root cause of ServletException java.lang.OutOfMemoryError
4. resin:java.lang.OutOfMemoryError
5. java:java.lang.OutOfMemoryError

解决java.lang.OutOfMemoryError的方法有如下几种：

一、增加jvm的内存大小。方法有： 1）在执行某个class文件时候，可以使用java -Xmx256M aa.class来设置运行aa.class时jvm所允许占用的最大内存为256M。 2）对tomcat容器，可以在启动时对jvm设置内存限度。对tomcat，可以在catalina.bat中添加：

set CATALINA\_OPTS=-Xms128M -Xmx256M

set JAVA\_OPTS=-Xms128M -Xmx256M

或者把%CATALINA\_OPTS%和%JAVA\_OPTS%代替为-Xms128M -Xmx256M

3）对resin容器，同样可以在启动时对jvm设置内存限度。在bin文件夹下创建一个startup.bat文件，内容如下：

@echo off

call "httpd.exe" "-Xms128M" "-Xmx256M"

:end

其中"-Xms128M"为最小内存，"-Xmx256M"为最大内存。

二、 优化程序，释放垃圾。

主要包括避免死循环，应该及时释放种资源：内存, 数据库的各种连接，防止一次载入太多的数据。导致java.lang.OutOfMemoryError的根本原因是程序不健壮。因此，从根本上解决Java内存溢出的唯一方法就是修改程序，及时地释放没用的对象，释放内存空间。 遇到该错误的时候要仔细检查程序，嘿嘿，遇多一次这种问题之后，以后写程序就会小心多了。

## **Java代码导致OutOfMemoryError错误的解决：**

需要重点排查以下几点：

1. 检查代码中是否有死循环或递归调用。
2. 检查是否有大循环重复产生新对象实体。
3. 检查对数据库查询中，是否有一次获得全部数据的查询。一般来说，如果一次取十万条记录到内存，就可能引起内存溢出。这个问题比较隐蔽，在上线前，数据库中数据较少，不容易出问题，上线后，数据库中数据多了，一次查询就有可能引起内存溢出。因此对于数据库查询尽量采用分页的方式查询。
4. 检查List、MAP等集合对象是否有使用完后，未清除的问题。List、MAP等集合对象会始终存有对对象的引用，使得这些对象不能被GC回收。

## **tomcat中java.lang.OutOfMemoryError: PermGen space异常处理**

****PermGen space****的全称是Permanent Generation space,是指内存的永久保存区域,这块内存主要是被JVM存放Class和Meta信息的,Class在被Loader时就会被放到PermGen space中, 它和存放类实例(Instance)的Heap区域不同,GC(Garbage Collection)不会在主程序运行期对PermGen space进行清理，所以如果你的应用中有很多CLASS的话,就很可能出现PermGen space错误, 这种错误常见在web服务器对JSP进行pre compile的时候。如果你的WEB APP下都用了大量的第三方jar, 其大小超过了jvm默认的大小(4M)那么就会产生此错误信息了。 解决方法： 手动设置MaxPermSize大小修改TOMCAT\_HOME/bin/catalina.sh在

echo "Using CATALINA\_BASE: $CATALINA\_BASE"

上面加入以下行：

JAVA\_OPTS="-server -XX:PermSize=64M -XX:MaxPermSize=128m

建议：将相同的第三方jar文件移置到tomcat/shared/lib目录下，这样可以达到减少jar 文档重复占用内存的目的。

## **weblogic中java.lang.OutOfMemoryError异常处理**

错误提示： Root cause of ervletException java.lang.OutOfMemoryError 解决办法：调整bea/weblogic/common中CommEnv中参数

:sun

　　if "%PRODUCTION\_MODE%" == "true" goto sun\_prod\_mode

　　set JAVA\_VM=-client

　　set MEM\_ARGS=-Xms256m -Xmx512m -XX:MaxPermSize=256m

　　set JAVA\_OPTIONS=%JAVA\_OPTIONS% -Xverify:none

　　goto continue

　　:sun\_prod\_mode

　　set JAVA\_VM=-server

　　set MEM\_ARGS=-Xms256m -Xmx512m -XX:MaxPermSize=256m

　　goto continue

## **Resin下java.lang.OutOfMemoryError异常处理**

产生内存溢出的原因：

出现这个错误，一般是因为JVM物理内存过小。默认的Java虚拟机最大内存仅为64兆，这在开发调试过程中可能没有问题，但在实际的应用环境中是远远不能满足需要的，除非你的应用非常小，也没什么访问量。否则你可能会发现程序运行一段时间后包java.lang.OutOfMemoryError的错误。因此我们需要提升resin可用的虚拟机内存的大小。

解决方法：

修改/usr/local/resin/bin/httpd.sh中的args选项 添加参数-Xms（初始内存）和-Xmx（最大能够使用内存大小）可以用来限制JVM的物理内存使用量。例如：

args="-Xms128m -Xmx256m"

设置后，JVM初始物理内存是128m，最大能使用物理内存为256m。

这两个值应该由系统管理员根据服务器的实际情况进行设置。