OutOfMemoryError异常实战

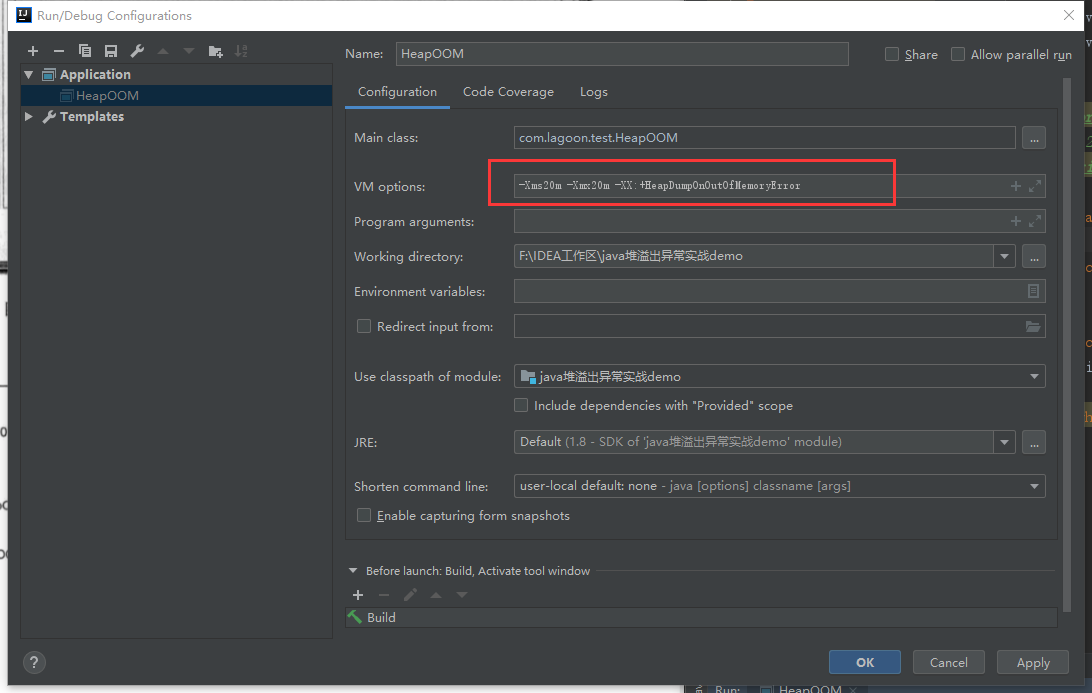
在Java虚拟机规范的描述中，除了程序计数器以外，虚拟机内存的其他几个运行时数据区域都有可能发生OutOfMemoryError异常（下文称OOM异常）的可能

1. **java堆溢出**

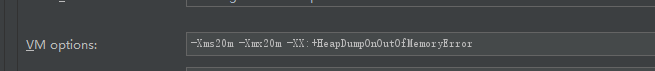
java堆用来存储对象实例，只要不断地创建对象，并且保证GC Roots到对象之间有可达路径来避免垃圾回收机制清除这些对象，就会在对象数量到达最大堆的容量限制后产生内存溢出异常。

代码限制java堆的大小为20MB，不可扩展（将堆的最小值-Xms参数与最大值-Xmx参数设置为一样即可避免堆自动扩展），通过参数-XX:+HeapDumpOnOutOfMemoryError 可以让虚拟机在出现内存溢出异常时Dump出当前的内存堆转储快照以便事后进行分析

在idea设置vm参数



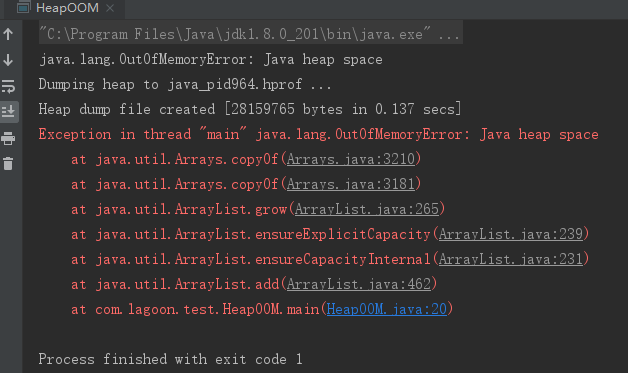
大图：



测试demo：

package com.lagoon.test;  
  
import java.util.ArrayList;  
import java.util.List;  
  
*/\*\*  
 \** ***@Author*** *WinkiLee  
 \** ***@Date*** *2019/5/4 17:01  
 \** ***@Description*** *java堆溢出异常实战demo  
 \*/*public class HeapOOM {  
  
 static class OOMObject{  
 }  
  
 public static void main(String[] args) {  
 List<OOMObject> list=new ArrayList<OOMObject>();  
  
 while (true){  
 list.add(new OOMObject());  
 }  
 }  
}

运行结果：

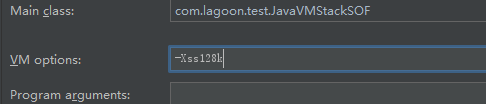


1. **虚拟机栈和本地方法栈溢出**

将试验范围限制于单线程中的操作，两种测试：

* 使用-Xss参数减少栈内存容量，结果：抛出StackOverflowError异常，异常出现是输出的栈的深度相应缩小
* 定义了大量的本地变量，增加此方法帧中本地变量表的长度，结果：抛出StackOverflowError异常时，输出的栈深度相应缩小

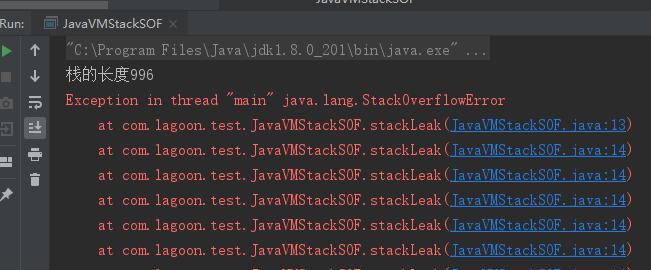
设置vm参数



第一测试demo：

package com.lagoon.test;  
  
*/\*\*  
 \** ***@Author*** *WinkiLee  
 \** ***@Date*** *2019/5/4 17:17  
 \** ***@Description*** *虚拟机溢出异常实战demo  
 \*/*public class JavaVMStackSOF {  
  
 private int stackLength=1;  
  
 public void stackLeak(){  
 stackLength++;  
 stackLeak();  
 }  
  
 public static void main(String[] args) {  
 JavaVMStackSOF oom=new JavaVMStackSOF();  
 try {  
 oom.stackLeak();  
 }catch (Throwable e){  
 System.*out*.println("栈的长度"+oom.stackLength);  
 throw e;  
 }  
 }  
}

运行结果：



实验结果表明：在单个线程下，无论是由于栈帧太大，还是虚拟机栈容量太小，当内存无法分配的时候，虚拟机抛出的都是StackOverflowError异常