1. Jconsole
2. Jvisualvm
3. Jmap
4. 类加载
   1. Java代码中，类型的加载，连接与初始化过程都是在程序运行期间完成的
   2. 加载（字节码加载到jvm中），连接（类之间的调用关系，符号引用转换直接引用）
   3. 初始化
5. Java虚拟机结束生命周期

1), 执行了System.exit()

2), 程序正常结束

3), 程序执行过程中遇到异常或错误而异常终止

4),操作系统故障

6， 类的加载,连接与初始化

1),加载： 查找并加载类的二进制数据

2),连接

验证: 确保被加载的类的正确性

准备: 为类的静态变量分配内存，并将其初始化为默认值（整型 0, Boolean false …）

解析: 把类中的符号引用转换为直接引用

3),初始化：为类的静态变量赋予正确的初始值

7， 类的使用与卸载

8，类的加载，连接，初始化

Java 程序对类的使用方式可分为两种

1. 主动使用
2. 被动使用

所有的Java虚拟实现必须在每个类或接口被java程序首次主动使用时才初始化他们

如下是对类的主动使用，除此之外都是被动使用

1,创建类的实例

2,访问某个类或接口的静态变量，或者对该静态变量赋值

3,调用类的静态方法

4,反射

5,初始化一个类的子类

6,java 虚拟机被标记为启动类的类

7,jdk7 开始提供的动态语言的支持：

Java.lang.invoke.MethodHandle 实例的解析结果REF\_getStatic,REF\_putStatic.REF\_invokeStatic句柄对应的类没有初始化则初始化。

9，类的加载

1，类的加载指的是将类的.class 文件中的二进制数据读入到内存中，将其放在运行时数据区的方法区内，然后在内存中创建一个java.lang.Class对象用来封装类再方法区的数据结构

2，加载.class文件的方式

1》从本地系统直接加载

2》通过网络下载.class

3》从zip,jar等归档文件中加载.class文件

4》从专有数据库中提取.class

5》将java源文件动态编译为.class文件

助记符号：getstatic/putstatic/invokestatic

Ldc:将int,float,String 类型的常量值从常量池中推送至栈顶

Bipush:将 -128~127 常量值从常量池中推送至栈顶

Sipush:将短整形常量值推送至栈顶(-32767~32767)

Iconst\_1:1

Iconst\_2:2

Iconst\_3:3

Iconst\_4:4

Iconst\_5:5