

java 深入之 JVM 指令手册

作者: [wgl530](#)

原文链接: <https://ld246.com/article/1577763341221>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p>载 float 类型值

fload_2 从局部变量 2 中装载 float 类型值

fload_3 从局部变量 3 中装载 float 类型值

dload_0 从局部变量 0 中装载 double 类型值

dload_1 从局部变量 1 中装载 double 类型值

dload_2 从局部变量 2 中装载 double 类型值

dload_3 从局部变量 3 中装载 double 类型值

aload_0 从局部变量 0 中装载引用类型值

aload_1 从局部变量 1 中装载引用类型值

aload_2 从局部变量 2 中装载引用类型值

aload_3 从局部变量 3 中装载引用类型值

iaload 从数组中装载 int 类型值

laload 从数组中装载 long 类型值

faload 从数组中装载 float 类型值

daload 从数组中装载 double 类型值

aaload 从数组中装载引用类型值

baload 从数组中装载 byte 类型或 boolean 类型值

caload 从数组中装载 char 类型值

saload 从数组中装载 short 类型值

将栈中的值存入局部变量的指令

istore 将 int 类型值存入局部变量

lstore 将 long 类型值存入局部变量

fstore 将 float 类型值存入局部变量

dstore 将 double 类型值存入局部变量

astore 将引用类型或 returnAddress 类型值存入局部变量

istore_0 将 int 类型值存入局部变量 0

istore_1 将 int 类型值存入局部变量 1

istore_2 将 int 类型值存入局部变量 2

istore_3 将 int 类型值存入局部变量 3

lstore_0 将 long 类型值存入局部变量 0

lstore_1 将 long 类型值存入局部变量 1

lstore_2 将 long 类型值存入局部变量 2

lstore_3 将 long 类型值存入局部变量 3

fstore_0 将 float 类型值存入局部变量 0

fstore_1 将 float 类型值存入局部变量 1

fstore_2 将 float 类型值存入局部变量 2

fstore_3 将 float 类型值存入局部变量 3

dstore_0 将 double 类型值存入局部变量 0

dstore_1 将 double 类型值存入局部变量 1

dstore_2 将 double 类型值存入局部变量 2

dstore_3 将 double 类型值存入局部变量 3

astore_0 将引用类型或 returnAddress 类型值存入局部变量 0

astore_1 将引用类型或 returnAddress 类型值存入局部变量 1

astore_2 将引用类型或 returnAddress 类型值存入局部变量 2

astore_3 将引用类型或 returnAddress 类型值存入局部变量 3

iastore 将 int 类型值存入数组中

lastore 将 long 类型值存入数组中

fastore 将 float 类型值存入数组中

dastore 将 double 类型值存入数组中

aastore 将引用类型值存入数组中

bastore 将 byte 类型或者 boolean 类型值存入数组中

castore 将 char 类型值存入数组中

sastore 将 short 类型值存入数组中

wide 指令

wide 使用附加字节扩展局部变量索引

通用(无类型) 栈操作

nop 不做任何操作

pop 弹出栈顶端一个字长的内容

pop2 弹出栈顶端两个字长的内容

dup 复制栈顶部一个字长内容

dup_x1 复制栈顶部一个字长的内容, 然后将复制内容及原来弹出的两个字长的内容压入栈

dup_x2 复制栈顶部一个字长的内容, 然后将复制内容及原来弹出的三个字长的内容压入栈

dup2 复制栈顶部两个字长内容

dup2_x1 复制栈顶部两个字长的内容, 然后将复制内容及原来弹出的三个字长的内容压入栈

dup2_x2 复制栈顶部两个字长的内容, 然后将复制内容及原来弹出的四个字长的内容压入栈

swap 交换栈顶部两个字长内容

类型转换

i2l 把 int 类型的数据转化为 long 类型

i2f 把 int 类型的数据转化为 float 类型

i2d 把 int 类型的数据转化为 double 类型

l2i 把 long 类型的数据转化为 int 类型

l2f 把 long 类型的数据转化为 float 类型

l2d 把 long 类型的数据转化为 double 类型

f2i 把 float 类型的数据转化为 int 类型

f2l 把 float 类型的数据转化为 long 类型

f2d 把 float 类型的数据转化为 double 类型

d2i 把 double 类型的数据转化为 int 类型

d2l 把 double 类型的数据转化为 long 类型

d2f 把 double 类型的数据转化为 float 类型

i2b 把 int 类型的数据转化为 byte 类型

i2c 把 int 类型的数据转化为 char 类型

i2s 把 int 类型的数据转化为 short 类型

整数运算

iadd 执行 int 类型的加法

ladd 执行 long 类型的加法

isub 执行 int 类型的减法

lsub 执行 long 类型的减法

imul 执行 int 类型的乘法

lmul 执行 long 类型的乘法

idiv 执行 int 类型的除法

ldiv 执行 long 类型的除法

irem 计算 int 类型除法的余数

lrem 计算 long 类型除法的余数

ineg 对一个 int 类型值进行取反操作

lneg 对一个 long 类型值进行取反操作

iinc 把一个常量值加到一个 int 类型的局部变量上

逻辑运算

移位操作

ishl 执行 int 类型的向左移位操作

lshl 执行 long 类型的向左移位操作

ishr 执行 int 类型的向右移位操作

lshr 执行 long 类型的向右移位操作

iushr 执行 int 类型的向右逻辑移位操作

lushr 执行 long 类型的向右逻辑移位操作

按位布尔运算

iand 对 int 类型值进行“逻辑与”操作

land 对 long 类型值进行“逻辑与”操作

ior 对 int 类型值进行“逻辑或”操作

lor 对 long 类型值进行“逻辑或”操作

ixor 对 int 类型值进行“逻辑异或”操作

lxor 对 long 类型值进行“逻辑异或”操作

浮点运算

fadd 执行 float 类型的加法

dadd 执行 double 类型的加法

fsub 执行 float 类型的减法

dsub 执行 double 类型的减法

fmul 执行 float 类型的乘法

dmul 执行 double 类型的乘法

fdiv 执行 float 类型的除法

ddiv 执行 double 类型的除法

frem 计算 float 类型除法的余数

drem 计算 double 类型除法的余数

fneg 将一个 float 类型的数值取反

dneg 将一个 double 类型的数值取反

对象和数组

对象操作指令

new 创建一个新对象

checkcast 确定对象为所给定的类型

getfield 从对象中获取字段

putfield 设置对象中字段的值

getstatic 从类中获取静态字段

putstatic 设置类中静态字段的值

instanceof 判断对象是否为给定的类型

数组操作指令

newarray 分配数据成员类型为基本上数据类型的新数组

anewarray 分配数据成员类型为引用类型的新数组

arraylength 获取数组长度

multianewarray 分配新的多维数组

控制流

条件分支指令

ifeq 如果等于 0, 则跳转

ifne 如果不等于 0, 则跳转

iflt 如果小于 0, 则跳转

ifge 如果大于等于 0, 则跳转

ifgt 如果大于 0, 则跳转

ifle 如果小于等于 0, 则跳转

if_icmpeq 如果两个 int 值相等, 则跳转

if_icmpne 如果两个 int 类型值不相等, 则跳转

if_icmplt 如果一个 int 类型值小于另外一个 int 类型值, 则跳转

if_icmpge 如果一个 int 类型值大于或者等于另外一个 int 类型值, 则跳转

if_icmpgt 如果一个 int 类型值大于另外一个 int 类型值, 则跳转

if_icmple 如果一个 int 类型值小于或者等于另外一个 int 类型值, 则跳转

ifnull 如果等于 null, 则跳转

ifnonnull 如果不等于 null, 则跳转

if_acmpeq 如果两个对象引用相等, 则跳转

if_acmpnc 如果两个对象引用不相等, 则跳转

比较指令

lcmp 比较 long 类型值

fcmpl 比较 float 类型值 (当遇到 NaN 时, 返回-1)

fcmpg 比较 float 类型值 (当遇到 NaN 时, 返回 1)

dcmpl 比较 double 类型值 (当遇到 NaN 时, 返回-1)

dcmpg 比较 double 类型值 (当遇到 NaN 时, 返回 1)

无条件转移指令

goto 无条件跳转

goto_w 无条件跳转 (宽索引)

表跳转指令

tableswitch 通过索引访问跳转表, 并跳转

lookupswitch 通过键值匹配访问跳转表, 并执行跳转操作

异常

athrow 抛出异常或错误

finally 子句

jsr 跳转到子例程

jsr_w 跳转到子例程 (宽索引)

rct 从子例程返回

方法调用与返回

方法调用指令

invokcvirtual 运行时按照对象的类来调用实例方法

invokespecial 根据编译时类型来调用实例方法

invokestatic 调用类 (静态) 方法

invokcinterface 调用接口方法

方法返回指令

ireturn 从方法中返回 int 类型的数据

lreturn 从方法中返回 long 类型的数据

freturn 从方法中返回 float 类型的数据

dreturn 从方法中返回 double 类型的数据

areturn 从方法中返回引用类型的数据

return 从方法中返回, 返回值为 void

线程同步

monitorenter 进入并获取对象监视器

monitorexit 释放并退出对象监视器

JVM 指令助记符

变量到操作数栈: iload,iload_,lload,lload_,fload,fload_,dload,dload_,aload,aload_

操作数栈到变量: istore,istore_,lstore,lstore_,fstore,fstore_,dstore,dstore_,astore,astore_

常数到操作数栈: bipush,sipush,ldc,ldc_w,ldc2_w,acnst_null,iconst_ml,iconst_,lconst_,fconst_,dconst_

加: iadd,ladd,fadd,dadd

减: isub,lsub,fsub,dsub

乘: imul,lmul,fmul,dmul

除: idiv,ldiv,fdiv,ddiv

余数: irem,lrem,frem,drem

取负: ineg,lneg,fneg,dneg

移位: ishl,lshr,iushr,lshl,lshr,lushr

按位或: ior,lor

按位与: iand,land

按位异或: ixor,lxor

类型转换: i2l,i2f,i2d,l2f,l2d,f2d(放宽数值转换)

i2b,i2c,i2s,l2i,f2l,f2l,d2i,d2l,d2f(缩窄数值转换)

创建类实例: new

创建新数组: newarray,anewarray,multianwarray

访问类的域和类实例域: getfield,putfield,getstatic,putstatic

把数据装载到操作数栈: baload,caload,saload,iaload,laload,faload,daload,aaload

从操作数栈存储到数组: bastore,castore,sastore,iastore,lastore,fastore,dastore,aastore

获取数组长度: arraylength

检相类实例或数组属性: instanceof,checkcast

操作数栈管理: pop,pop2,dup,dup2,dup_xl,dup2_xl,dup_x2,dup2_x2,swap

有条件转移: ifeq,iflt,ifle,ifne,ifgt,ifge,ifnull,ifnonnull,if_icmpeq,if_icmpene,

if_icmplt,if_icmpgt,if_icmple,if_icmpge,if_acmpeq,if_acmpne,lcmp,fcmpl

fcmpg,dcmpl,dcmpg

复合条件转移: tableswitch,lookupswitch

无条件转移: goto,goto_w,jsr,jsr_w,ret

调度对象的实便方法: invokevirtual

调用由接口实现的方法: invokeinterface

调用需要特殊处理的实例方法: invokespecial

调用命名类中的静态方法: invokestatic

方法返回: ireturn,lreturn,freturn,dreturn,areturn,return

异常: athrow

finally 关键字的实现使用: jsr,jsr_w,ret</p>