

## 栈和局部变量操作

### 将常量压入栈的指令

aconst\_null 将null对象引用压入栈

iconst\_m1 将int类型常量-1压入栈

iconst\_0 将int类型常量0压入栈

iconst\_1 将int类型常量1压入操作数栈

iconst\_2 将int类型常量2压入栈

iconst\_3 将int类型常量3压入栈

iconst\_4 将int类型常量4压入栈

iconst\_5 将int类型常量5压入栈

lconst\_0 将long类型常量0压入栈

lconst\_1 将long类型常量1压入栈

fconst\_0 将float类型常量0压入栈

fconst\_1 将float类型常量1压入栈

dconst\_0 将double类型常量0压入栈

dconst\_1 将double类型常量1压入栈

bipush 将一个8位带符号整数压入栈

sipush 将16位带符号整数压入栈

ldc 把常量池中的项压入栈

ldc\_w 把常量池中的项压入栈（使用宽索引）

ldc2\_w 把常量池中long类型或者double类型的项压入栈（使用宽索引）

### 从栈中的局部变量中装载值的指令

iload 从局部变量中装载int类型值

lload 从局部变量中装载long类型值

float 从局部变量中装载float类型值

dload 从局部变量中装载double类型值

aload 从局部变量中装载引用类型值（reference）

iload\_0 从局部变量0中装载int类型值

iload\_1 从局部变量1中装载int类型值

iload\_2 从局部变量2中装载int类型值

iload\_3 从局部变量3中装载int类型值

lload\_0 从局部变量0中装载long类型值

lload\_1 从局部变量1中装载long类型值

lload\_2 从局部变量2中装载long类型值

lload\_3 从局部变量3中装载long类型值

float\_0 从局部变量0中装载float类型值

float\_1 从局部变量1中装载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

iiinc 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\_icmpcq 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\_l,const\_f,const\_d,const\_

加: iadd,ladd,fadd,dadd

减: isub,lsup,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,f2i,f2l,d2i,d2l,d2f(缩窄数值转换)

创建类实例: new

创建新数组: newarray,newarray,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

调度对象的实例方法: 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`