

Instruktion	Bedeutung	Stack vorh.	Stack nachh.
iload_x	lege int-Wert von lok. Var. mit Nr. x ($0 \leq x \leq 3$) auf den Stack	(...)	(...) (Wert v. x)
iload x	lege int-Wert von lok. Var. mit Nr. x auf den Stack	(...)	(...) (Wert v. x)
aload_x	lege Referenz in lok. Var. mit Nr. x ($0 \leq x \leq 3$) auf den Stack	(...)	(...) (Ref. in x)
aload x	lege Referenz in lok. Var. mit Nr. x auf den Stack	(...)	(...) (Ref. in x)
iaload	lese int-Wert aus Array a an Position i	(...)(a)(i)	(...) ($a[i]$)
istore_x	verschiebe obersten int-Wert auf Stack in lok. Var. mit Nr. x ($0 \leq x \leq 3$)	(...)(a)	(...)
istore x	verschiebe obersten int-Wert auf Stack in lok. Var. mit Nr. x	(...)(a)	(...)
iconst_v	lege Konstante v ($-1 \leq v \leq 5$) als int auf den Stack	(...)	(...) (v als int)
bipush v	lege Byte v als int auf den Stack	(...)	(...) (v als int)
sipush v	lege Short v als int auf den Stack	(...)	(...) (v als int)
ldc i	lege i -te Konstante aus Konstantenpool als int auf den Stack	(...)	(...) (i -te Konst. als int)
iadd	addiere die beiden obersten int-Werte auf dem Stack	(...)(a)(b)	(...) ($a+b$)
isub	subtrahiere die beiden obersten int-Werte auf dem Stack	(...)(a)(b)	(...) ($a-b$)
imul	multipliziere die beiden obersten int-Werte auf dem Stack	(...)(a)(b)	(...) ($a*b$)
goto t	springe an Instruktion mit Offset t	(...)	(...)
if_icmple t	springe an Instruktion mit Offset t , falls $a \leq b$ (ints)	(...)(a)(b)	(...)
if_eq t	springe an Instruktion mit Offset t , falls $a = 0$ (ints)	(...)(a)	(...)
if_ne t	springe an Instruktion mit Offset t , falls $a \neq 0$ (ints)	(...)(a)	(...)