**Come scrivere un programma IJVM**

– Costanti

– Metodo MAIN

– Metodi

– Variabili locali

– Etichette

– Commenti

– I/O

**Costanti globali**

• Sono dichiarate in una sezione all’inizio del file

• Il formato può essere decimale (senza prefissi) , esadecimale (se preceduto da “0x”) oppure ottale (se preceduto da “o”).

• Le costanti possono poi essere usate facendo riferimento al loro nome

• I valori delle costanti sono caricati nella CONSTANT POOL

**Sintassi**

.constant

costante1 valore1

costante2 valore2

…..

.end-constant

**ESEMPIO**

.constant

MAX\_LENGTH 12

.end-constant

….

LDC\_W MAX\_LENGTH

…..

**Metodo Main**

• Ogni programma IJVM deve contenere uno speciale metodo chiamato *main*

• Il *main* deve essere dichiarato prima di ogni altro metodo del programma

**Sintassi**

.main

….. Dichiarazione delle variabili locali

….. Codice del main

.end-main

**Metodi**

• I parametri di un metodo sono dichiarati tra parentesi tonde e separati da virgole

• Un metodo può avere zero parametri

• La chiamata di un metodo da parte di codice IJVM deve avvenire seguendo un preciso protocollo:

**Metodi**

• Push di OBJREF. In JVM è il riferimento all’oggetto sul quale invocare il metodo. IJVM mantiene questa caratteristica per uniformità ma il valore di OBJREF è inessenziale. In ogni caso, un qualsiasi valore per OBJREF deve essere messo in

cima allo stack.

• Push dei parametri nell’ordine in cui i parametri di un metodo sono specificati nella dichiarazione.

• Chiamata del metodo con INVOKEVIRTUAL method\_name.

• Il valore di ritorno si trova in cima allo stack. L’istruzione IRETURN copia la cima dello stack del metodo sulla cima dello stack di chi ha invocato il metodo.

**Sintassi**

.method nome\_metodo(param1, param2,...)

….. Dichiarazione delle variabili locali

….. Codice del metodo

.end-method

**Variabili locali**

• Sono dichiarate all’interno di una dichiarazione di metodo e/o del main

• Si possono accedere tramite il riferimento al nome solo all’interno del metodo dove si trova la loro dichiarazione

• Non ci sono variabili globali

**Sintassi**

.var

var1

var2

…..

.end-var

**ESEMPIO**

….

ISTORE var1

…..

**Etichette**

• Servono ad identificare istruzioni in un programma che possono essere l’argomento di un’istruzione di salto

• Le etichette sono accessibili solo all’interno del metodo dove sono state dichiarate (I salti possono avvenire all’interno di un metodo e non “tra metodi”)

• Le etichette sono stringhe alfanumeriche terminate dal carattere “:”

**Sintassi**

ILOAD total

LDC\_W max

ISUB

IFLT lt\_max // If total < max, goto lt\_max

GOTO gte\_max // else total >= max, goto gte\_max

lt\_max: HALT

gte\_max: // Possono essere su linee separate

INVOKEVIRTUAL error

**Commenti**

• Servono ad inserire annotazioni al codice prodotto (FONDAMENTALI)

• Si indica l’inizio di un commento con due slash “//”

• Tutti i caratteri dal “//” alla fine della linea sono trattati come commenti

**Struttura generale di un programma IJVM**

.constant

costante1 valore1

costante2 valore2

…..

.end-constant

.main

….. Dichiarazione delle variabili locali del main

….. Codice del main

.end-main

.method nome\_metodo1(param1, param2,...)

….. Dichiarazione delle variabili locali del metodo1

….. Codice del metodo1

.end-method

.method nome\_metodo2(param1, param2,...)

….. Dichiarazione delle variabili locali del metodo2

….. Codice del metodo2

.end-method

….

….

**Problema numero 1**

• Scrivere un programma IJVM che scambia il contenuto di due variabili locali X e Y

.main

.var

XY

.end-var

start: BIPUSH 0x10

ISTORE X // inizializzo X=0x10

BIPUSH 0x20

ISTORE Y // inizializzo Y=0x20

// scambia il contenuto

ILOAD X

ILOAD Y

ISTORE X

ISTORE Y

HALT

.end-main

**Problema numero 2**

• Scrivere un programma IJVM che esegua X=min(X,Y) e Y=max(X,Y)

.constant

tre 0x03

quattro 0x04

.end-constant

.main

.var

X

Y

.end-var

start:

LDC\_W quattro

ISTORE X

LDC\_W tre

ISTORE Y

ILOAD X

ILOAD Y

ISUB

IFLT end // if X<Y vai a end

ILOAD X

ILOAD Y

ISTORE X

ISTORE Y

end: HALT

.end-main

**Problema numero 3**

• Scrivere un programma IJVM che sommi i primi 32 numeri interi (struttura ciclo **do-while**)

.main

.var

sum

cont

.end-var

START:

BIPUSH 0x0

ISTORE sum // inizializzo sum=0x00

BIPUSH 0x0

ISTORE cont // inizializzo cont=0x00

CICLO: IINC cont 1 // incrementa contatore

ILOAD cont // carica cont sullo stack

ILOAD sum // carica sum sullo stack

IADD // sum+cont

ISTORE sum // memorizza somma

ILOAD cont // carica cont sullo stack

BIPUSH 0x20 // carica la costante 32

IF\_ICMPEQ L2 // se cont==32 vai a L2

GOTO CICLO // salta all'inizio del ciclo

L2: HALT

.end-main