Bytecode

Java bytecode is the instruction set of the Java virtual machine.

Each bytecode is composed of one, or two bytes that represent the instruction (opcode)

with zero or more bytes for passing parameters.

There is no necessary connection between the Java programming language and Java bytecode.

A program written in Java can be compiled directly into the machine language of a real computer

Programs written in other languages than Java can be compiled into Java bytecode.

Java bytecode is intended to be platform-independent and secure.

The JVM instructions groups

- Load and store (e.g. aload_0, istore)
- Arithmetic and logic (e.g. ladd, fcmpl)
- Type conversion (e.g. i2b, d2i)
- Object creation and manipulation (new, putfield)
- Operand stack management (e.g. swap, dup2)
- Control transfer (e.g. ifeq, goto)
- Method invocation and return (e.g. invokespecial, areturn)
- Throwing exceptions
- Monitor-based concurrency

Prefixes and suffixes

Many instructions have prefixes and/or suffixes referring to the types of operands they operate on.

Prefix-suffix	Operand type
`i`	integer
`l`	long
`s`	short
`b`	byte
`c`	character
`f`	float
`d`	double
`z`	boolean
`a`	reference

Examples:

iadd – sum of integers

s2i – short to integer

.jar and .class files

jar file = zip archive

jar xf name.jar

unzip -d jarcontent name.jar

Contains

META-INF
classes

Java Class File Disassembler

javap -c name.class

javap -v name.class

- Meta
- Constant pool
- Bytecode
- SourceFile
- InnerClasses
- BootstrapMethods