

Appunti per l'esame 1Z0-811 Java Foundations Associate

Capitolo 1

Programmi e linguaggi di programmazione

Un programmatore istruisce un computer con delle istruzioni (che è il nostro codice), sono istruzioni in un linguaggio che il computer possa capire, questo è il motivo per cui un programmatore fa uso di un linguaggio di programmazione. Il computer di base è in grado di comprendere soltanto il "linguaggio macchina" (sequenze di 1 e 0 binarie), i linguaggi che usiamo oggi per programmare vengono definiti ad "alto livello" e sono ad esempio Java, Python, C, PHP, Javascript... Vengono definiti di alto livello, perché le istruzioni che si impartiscono al computer vengono espresse in linguaggio naturale anziché macchina. Linguaggio naturale inglese. Questo linguaggio naturale viene poi tradotto da una sorta di "translator" (l'interprete) in codice macchina perché il computer lo capisca. Chi traduce il linguaggio di programmazione in codice macchina può essere l'interprete o il "compilatore", il flusso è il seguente:

Programmatore → istruzioni ad alto livello (es. java code) → translator → linguaggio macchina (101010101) → "Hello World" sullo schermo del computer

Le Applicazioni

Sono programmi che permettono di usare il computer. Bisogna installarle sul computer per poterle usare. Le applicazioni che girano sul pc si chiamano "applicazioni desktop", poi ci sono delle classi di applicazioni cosiddette "mobile" che funzionano soltanto sugli smartphone, e poi ci sono anche applicazioni (ad es. Gmail) che vengono definite "applicazioni web" che si possono usare attraverso un browser sul pc. Le applicazioni si scrivono con i linguaggi di programmazione.

Api (application program interface)

Sono programmi che vengono esposti da un servizio attraverso una interfaccia che può permettere di usare il servizio ad un altro programma. Questa interfaccia viene per l'appunto definita api.

Capitolo 2

Che cosa è Java

Java è un linguaggio di programmazione e come tutti i linguaggi di programmazione ha le sue "keywords", sintassi e regole.

JDK → java development kit, come sappiamo il codice scritto in un linguaggio di programmazione deve essere tradotto per il computer, java per fare ciò si serve di istruzioni quali il compilatore, il debugger e il disassembler che aiutano a scrivere codice, tutti insieme questi strumenti sono chiamati il Java Development Kit.

JVM → Java Virtual Machine, è possibile definirla un computer a sua volta. Se pensiamo che ogni dispositivo esegue un sistema operativo e che per ogni dispositivo se voglio eseguirci un programma devo adattarlo a quel sistema è proprio qui che entra in gioco la JVM. La JVM non è un computer fisico, ma è un software, che si esegue sopra un computer fisico. La JVM si prende cura di tradurre il nostro codice Java in un linguaggio macchina. La JVM è stata sviluppata per diverse architetture, però ce ne sono alcune che non sono supportate, soprattutto quelle che non hanno nulla in comune con i computer desktop, ecco perché non possiamo eseguire applicazioni desktop su dispositivo iOS o Android.

Bytecode → è il linguaggio macchina che capisce la JVM, se vogliamo eseguire programmi sulla JVM per prima cosa li dobbiamo “compilare” in istruzioni scritte in java bytecode, alla fine poi la JVM usa le funzionalità del sistema operativo per eseguire il bytecode.

Java Class Library → libreria che comprende un sacco di funzionalità si chiama “Java Api’s” tutto questo codice è organizzato in package.

JRE → Java Runtime Environment → si tratta della JVM e della collezione di classi di libreria precompilate, molti computer le hanno installate, ma se devo installarle le posso scaricare. Basata sul principio WORA (write once run everywhere).

N.B.

Le applicazioni java sono leggermente differenti a seconda del sistema operativo che abbiamo al di sotto, perché la JVM non ha dipendenze sul linguaggio java, si possono scrivere programmi in altri linguaggi e compilare in bytecode, ogni parte di java lavora in maniera indipendente questa è una delle ragioni di successo. La JVM è come un programma qualunque su computer e usa il sistema operativo sottostante.

Java Platform → è la combinazione tra JVM, JRE e Dev Tools (compilatore ecc..), come base di partenza c’è la Java Standard Edition (Java SE) che contiene la JVM, La Standard Class Library e i DevTools, tutto questo è sufficiente per creare applicazioni stand alone. Per la parte Enterprise abbiamo Java Enterprise Edition (J2EE) che contiene la JVM, standard and enterprise library e i dev tools, c’è poi anche Java Micro Edition (JME) per il mobile e i dispositivi elettronici.

La piattaforma Java è standardizzata per tutti i device. Esempio: se un device supporta Java SE allora supporta tutti gli elementi di Java SE, quindi se sviluppi un programma Java SE funzionerà su tutti i dispositivi che lo supportano.

N.B. J2EE ad oggi è Jakarta EE

Versioni di Java → Java 1.0 è del 1996, Java 8 risale al 2014.

Java Community Process → JCP (1998) → organizzazione indipendente di utilizzatori Java. Controlla che java non venga alterato.

Caratteristiche di Java

- Indipendenza dalla piattaforma
- Performance Elevate
- Sicurezza
- Familiarità
- Semplicità
- Molteplici modalità di deploy
- Ecosistema

- Backward compatibility
- Compilato → compilato in bytecode per mezzo del compilatore, il bytecode viene poi tradotto dalla JVM e questo fa sì che Java sia un linguaggio compilato. I linguaggi compilati come Java hanno performance migliori a runtime di quelli interpretati come JavaScript o PHP.
- Varietà di soluzioni (jsp, jpa, ejb)
- Multithreading
- Distribuito
- Garbage collection
- Orientato agli oggetti puro
- Strutturato
- Fortemente tipizzato
- Gestione automatica della memoria
- Meccanismo delle eccezioni per il controllo degli errori