



# TOMCAT

COME IL TOMCAT CHE SI AVVENTURA INESPLORATO NEL BUIO DELLA NOTTE, COSÌ L'UOMO CERCA IL SIGNIFICATO DELLA VITA TRA LE OMBRE DELL'IGNOTO.

ALESSANDRO MASONE

Java .....	2
Introduzione.....	2
Cos'è la programmazione Java? .....	2
Perché usare Java?.....	2
Terminologia Java .....	3
Kit di sviluppo Java (JDK) .....	3
Ambiente run-time Java (JRE) .....	3
Macchina virtuale Java (JVM).....	4
Bytecode nel processo di sviluppo .....	4
Garbage Collector .....	4
Caratteristiche principali di Java .....	5
Piattaforma indipendente.....	5
Linguaggio di programmazione orientato agli oggetti .....	5
Applicazioni del linguaggio di programmazione Java.....	6
Database .....	7
Cos'è il DBMS? .....	7
Che cos'è un database?.....	7
Cosa sono le tabelle e i campi? .....	7
Cos'è una chiave esterna?.....	8
Che cos'è una join? .....	8
Che cos'è un indice?.....	8
Che cos'è una query? .....	8
Cos'è un vincolo? .....	9
Cos'è un'entità debole? .....	9

# Java

## Introduzione

Java è un semplice linguaggio di programmazione orientato agli oggetti basato su classi. È un linguaggio di programmazione generico e di alto livello che aiuta programmatori e sviluppatori a scrivere un codice una volta ed eseguirlo ovunque.

Java è considerato sia un linguaggio compilato che interpretato. È perché il codice sorgente Java viene prima compilato in bytecode che viene quindi interpretato da Java Virtual Machine. Java Virtual Machine interpreta il bytecode e lo converte in codice macchina specifico della piattaforma. Quindi, Java è anche chiamato un linguaggio di programmazione indipendente dalla piattaforma.

## Cos'è la programmazione Java?

Java è un linguaggio di programmazione che James Gosling sviluppò presso Sun Microsystems\_Inc nel 1995, che in seguito fu acquisito dalla Oracle Corporation nel 2009.

- È possibile definirlo un linguaggio di programmazione di alto livello (rende lo sviluppo di programmi facile e molto più user-friendly) che rende molto conveniente scrivere, compilare ed eseguire il debug di programmi Java.
- Java è un linguaggio di programmazione orientato agli oggetti basato su classi che implementa il principio del codice write once ovunque.
- Poiché le applicazioni Java sono compilate in byte-code, possono essere eseguite su qualsiasi macchina supportata da JVM.
- I codici Java sono molto simili a C/C++, il che li rende più facili da capire.

## Perché usare Java?

Java è un linguaggio di alto livello e va studiato se si vogliono conoscere le basi della programmazione.

1. Orientato agli oggetti

2. Portatile
3. Semplice
4. Sicuro
5. Robusto
6. Indipendente dalla piattaforma
7. Architettura neutrale

## Terminologia Java

### Kit di sviluppo Java (JDK)

Come afferma formalmente il nome, Java Development Kit è un kit a tempo pieno che include un compilatore, Java Runtime Environment (JRE), debugger e documenti Java. Per un'ulteriore esecuzione in java, bisogna avere JDK installato sui dispositivi per portare ulteriormente alla creazione, compilazione ed esecuzione del programma java.

Qui, si utilizza JDK, si necessita di un ambiente per eseguire i programmi. Utilizzando JRE Java Runtime Environment, che fornisce i requisiti minimi per eseguire il programma Java. Fornisce JVM, classi Core e file di supporto.

### Ambiente run-time Java (JRE)

JDK include JRE, che a sua volta, dopo l'installazione, consente l'esecuzione del programma Java. Ma non è possibile ancora compilarlo. Ha un browser, l'applet supporta e alcuni plugin inclusi in esso. Quindi, per eseguire un programma java sui rispettivi computer, è necessario JRE.

JRE è composto da più elementi insieme e sono:

- Macchina virtuale Java (JVM)
- Librerie di classi Java
- Caricatore di classi Java

Quando il software tende a eseguire un particolare programma, richiede un certo ambiente per essere eseguito. Di solito si tratta di qualsiasi sistema

operativo, ad esempio Unix, Linux, Microsoft Windows o macOS. Qui JRE funge da traduttore e facilitatore tra il programma java e il sistema operativo.

## Macchina virtuale Java (JVM)

Questo generalmente è indicato come JVM e contiene tre fasi da seguire. È una parte molto importante non solo di JDK ma anche di JRE poiché è integrata in entrambi i luoghi. Quando si esegue un programma utilizzando JRE e JDK, passa anche alla JVM poiché è necessario per eseguire il programma java e interpreta il programma. Le fasi sono le seguenti:

Compilare il codice: il Java Development Kit (JDK) ci fornisce il compilatore JAVAC per superare questo passaggio. Fondamentalmente richiediamo al JDK di convertire il nostro codice sorgente in un formato specifico (codice compilato) che può essere facilmente interpretato da Java Runtime Environment (JRE).

Eseguire il codice: JVM esegue il bytecode fornito dal compilatore. Poiché Java è un linguaggio indipendente dalla piattaforma, il codice compilato prodotto dal compilatore JAVAC viene convertito in codice macchina utilizzando JVM specifiche della piattaforma. Piattaforme diverse hanno JVM diverse. Le JVM convertono il bytecode in codice macchina specifico della piattaforma.

## Bytecode nel processo di sviluppo

Il codice sorgente fornito al compilatore JAVAC presente in JDK viene compilato in un bytecode che la JVM può eseguire. Viene salvato come file .class dal compilatore stesso. È possibile anche visualizzare questo bytecode usando un disassemblatore come javap.

## Garbage Collector

All'interno di Java, i programmatori non possono eliminare gli oggetti. Quindi, per cancellare o richiamare un ricordo, JVM ha un Garbage Collector. Questi Garbage Collector possono richiamare quegli oggetti che non sono stati referenziati. L'uso di questo Java rende la vita di un programmatore molto comoda gestendo la memoria. Nonostante ciò, i programmatori dovrebbero essere consapevoli di ciò che stanno scrivendo nei loro codici o se stanno

utilizzando oggetti che sono già stati utilizzati per più tempo. Questo perché questo raccoglitore non può ripristinare la memoria degli oggetti a cui si fa riferimento. La raccolta dei rifiuti qui è un processo automatico.

Caratteristiche principali di Java

## Piattaforma indipendente

Innanzitutto, il compilatore converte il codice sorgente nel suo bytecode e, successivamente, JVM esegue il bytecode generato dal suo compilatore. Inoltre, questo bytecode può essere eseguito su qualsiasi altra piattaforma, ad esempio Windows, Linux o macOS. Ciò significa che possiamo compilare un programma su Windows, quindi possiamo eseguirlo anche su Linux utilizzando la JVM di Linux.

Ogni sistema operativo ha diversi set di JVM. Queste diverse JVM generano codice macchina specifico della piattaforma diverso dallo stesso bytecode. Quindi, l'output che apparirà su tutti i sistemi operativi sarà lo stesso dopo l'esecuzione del codice sorgente. E quindi lo chiamiamo un linguaggio indipendente dalla piattaforma.

## Linguaggio di programmazione orientato agli oggetti

La programmazione orientata agli oggetti organizza la progettazione e la logica del software attorno a classi, oggetti e dati piuttosto che a procedure o funzioni. La programmazione orientata agli oggetti utilizza concetti del mondo reale per implementare il codice dell'applicazione e mantenere la sicurezza e l'integrità dei dati al centro della progettazione.

I quattro principi di base della programmazione orientata agli oggetti:

- Ereditarietà: L' ereditarietà può essere definita come un oggetto che acquisisce tutti i comportamenti e le proprietà di un oggetto genitore. Rende il codice riutilizzabile. Il polimorfismo di runtime può essere ottenuto utilizzando l'ereditarietà.
- Polimorfismo: Quando si esegue una singola attività in più modi, può essere definita come polimorfismo. In Java, l'override del metodo del

linguaggio di programmazione e l'overload del metodo vengono utilizzati per ottenere il polimorfismo.

- **Overloading:** quando più metodi nella stessa classe hanno gli stessi nomi ma parametri diversi.
- **Overriding:** quando il nome del metodo e i parametri sono gli stessi nella classe figlia e nella superclasse, lo chiamiamo overriding.
- Astrazione: L'astrazione è il concetto di nascondere i dettagli dell'implementazione interna ed esporre solo le funzionalità richieste agli utenti finali.

## Applicazioni del linguaggio di programmazione Java

Man mano che Java continuava a migliorare e ad avanzare, divenne piuttosto popolare, costruendo diverse configurazioni in grado di adattarsi a più piattaforme.

Java è famoso per avere la proprietà scrivi una volta, esegui ovunque. Ha anche nuove versioni J2, rinominate rispettivamente Java SE, Java ME e Java EE.

### *Multithread*

Utilizzando questa funzione, diventa facile per noi scrivere programmi in grado di eseguire più attività contemporaneamente. Questo è utile per gli sviluppatori quando realizzano applicazioni interattive che possono funzionare senza problemi.

### *Alte prestazioni*

L'architettura Java è progettata in modo tale da ridurre il sovraccarico quando si esegue un'applicazione e, a volte, utilizza il compilatore JIT (Java In Time) in cui il compilatore compila facilmente le basi del codice su richiesta in cui compila solo i metodi che chiamiamo che rendono le applicazioni più veloci.

### *Distribuito*

Quando usiamo il linguaggio di programmazione Java, diventiamo abbastanza capaci di creare applicazioni distribuite. RMI (Remote Method Invocation) ed

EJB (Enterprise Java Beans) ci aiutano a creare applicazioni distribuite. Capiamo in parole semplici: La lingua può essere facilmente distribuita su più di un sistema che è connesso tra loro attraverso una connessione internet. Gli oggetti su una JVM possono essere facilmente utilizzati per eseguire le procedure su una JVM remota.

## Database

Cos'è il DBMS?

Un Database Management System (DBMS) è un programma che controlla la creazione, la manutenzione e l'uso di un database. DBMS può essere definito come File Manager che gestisce i dati in un database anziché salvarli nei file system.

Che cos'è un database?

Il database non è altro che una forma organizzata di dati per un facile accesso, archiviazione, recupero e gestione dei dati. Questo è anche noto come forma strutturata di dati a cui è possibile accedere in molti modi.

Esempio: Database di gestione della scuola, Database di gestione della banca.

Cosa sono le tabelle e i campi?

Una tabella è un insieme di dati organizzati in un modello con colonne e righe.

Le colonne possono essere classificate come verticali e le righe sono orizzontali. Una tabella ha un numero specificato di colonne chiamate campi ma può avere un numero qualsiasi di righe che viene chiamato record.

Esempio:

Tabella: Impiegato

Campo: ID, Nome, Data di nascita.

Dati: 201456, Luca, 15/11/1960.

Cos'è una chiave primaria?



Una chiave primaria è una combinazione di campi che specificano in modo univoco una riga. Questo è un tipo speciale di chiave univoca e ha un vincolo NOT NULL implicito. Significa che i valori della chiave primaria non possono essere NULL.

Cos'è una chiave esterna?

Una chiave esterna è una tabella che può essere correlata alla chiave primaria di un'altra tabella. La relazione deve essere creata tra due tabelle facendo riferimento alla chiave esterna con la chiave primaria di un'altra tabella.

Che cos'è una join?

Questa è una parola chiave utilizzata per interrogare i dati da più tabelle in base alla relazione tra i campi delle tabelle. Le chiavi svolgono un ruolo importante quando vengono utilizzati le join.

Che cos'è un indice?

Un indice è un metodo di ottimizzazione delle prestazioni per consentire un recupero più rapido dei record dalla tabella. Un indice crea una voce per ogni valore e sarà più veloce recuperare i dati.

Che cos'è una relazione e cosa sono?

Database relazionale è definita come la connessione tra le tabelle in un database. Esistono varie relazioni di base dei dati e sono le seguenti:

- Relazione uno a uno.
- Relazione uno a molti.
- Relazione molti a uno. 2022-11-01

ALESSANDRO MASONE 1

- Relazione autoreferenziale.

Che cos'è una query?

Una query DB è un codice scritto per recuperare le informazioni dal database. La query può essere progettata in modo tale da corrispondere alle nostre aspettative sul set di risultati. Semplicemente, una domanda al Database.

Cos'è un vincolo?

Il vincolo può essere utilizzato per specificare il limite del tipo di dati della tabella.

Il vincolo può essere specificato durante la creazione o la modifica dell'istruzione della tabella. Esempio di vincolo sono.

- NOT NULL.
- CHECK.
- DEFAULT.
- UNIQUE.
- PRIMARY KEY.
- FOREIGN KEY.

Cos'è un'entità debole?

Un'entità è definita debole se non possiede un attributo chiave. Per esistere, un'entità debole deve appoggiarsi a un'altra entità tramite un'associazione identificante. È importante che l'entità a cui si appoggi sia un'entità forte, cioè in grado di esistere senza il supporto di altre entità.

## Tomcat

Cos'è Tomcat?

È un contenitore servlet Java open source che implementa molte specifiche Java Enterprise come l'API dei siti Web, le pagine del server Java e, ultimo ma non meno importante, il servlet Java. Il nome completo di Tomcat è "Apache Tomcat" è stato sviluppato in un ambiente aperto e partecipativo e rilasciato nel 1998 per la prima volta. È iniziato come implementazione di riferimento per le prime Java-Server Pages e l' API Java Servlet. È ancora uno dei java-server più utilizzati grazie a diverse funzionalità come la buona estensibilità, il comprovato motore di base e ben collaudato e durevole. È un tipo di software che consente al server web di gestire il contenuto dinamico (basato su java) utilizzando i protocolli Http.

cos'è esattamente Apache Tomcat?

Se qualcuno vuole fornire qualsiasi servizio web, come ad esempio fornire un semplice contenuto statico utilizzando HTML (o Hypertext Markup Language), o magari solo inviare dati da un server all'indirizzo, allora ha bisogno necessariamente di un server e quel server è l'HTTP (protocollo di trasferimento Hypertext). Come tutti sappiamo, se qualcuno vuole creare un sito web semplice e statico, ha bisogno di un server HTTP, ma se vuole rendere il sito web dinamico, deve utilizzare le servlet. L'HTTP server viene utilizzato se si desidera inviare dati semplici. Se invece si vogliono inviare dati dinamici o rendere il sito web dinamico, allora si devono utilizzare le servlet. Quindi, si necessita di un server HTTP e, inoltre, bisogna avere un contenitore in cui eseguire le servlet. Quando combiniamo l'HTTP server e le servlet (o il contenitore delle servlet), entrambi diventano un singolo server, come ad esempio il server Tomcat.

In parole semplici, possiamo dire che Apache Tomcat è sia un server che un contenitore servlet.

Che tipo di server è Tomcat?

L'ecosistema Java supporta un'ampia varietà di server delle applicazioni, quindi discutiamo un po' su ciascuno di essi e vediamo dove si inserisce Tomcat:

Un contenitore servlet è fondamentalmente un'implementazione della specifica servlet Java, che viene utilizzata principalmente allo scopo di ospitare servlet Java.

Il server delle applicazioni Java è un'implementazione della specifica Java.

Un server Web è un tipo di server progettato per fornire file utilizzando un sistema locale come Apache.

Possiamo dire che, al centro, il Tomcat è JSP (Java Server Pages) e Servlet. JSP è una delle tecnologie di programmazione lato server che consente agli sviluppatori di creare contenuti dinamici indipendenti dalla piattaforma ed è anche nota come tecnologia di rendering della vista lato server. Una servlet è un componente software basato su Java che aiuta a estendere le capacità di un server. Tuttavia, può anche rispondere a diversi tipi di richieste e in genere

implementare contenitori di server Web per ospitare le applicazioni Web sui server Web. Dal punto di vista dello sviluppatore, bisogna solo scrivere le pagine del server java (o JSP) o delle servlet e non bisogna preoccuparsi del routing, Tomcat gestirà il routing.

Tomcat è costituito anche dal server web noto come motore Coyote grazie al quale è possibile estendere la capacità di Tomcat per includere diverse specifiche Java Enterprise e includere Java Persistence API (JPA) . Tomcat ha anche una versione estesa nota come "TomEE" che contiene più funzionalità.

#### Vantaggi di Tomcat

- **È open-source**

Significa che chiunque da qualsiasi luogo può scaricarlo, installarlo e usarlo gratuitamente, il che lo rende la prima scelta tra i nuovi sviluppatori e i nuovi utenti.

- **Incredibilmente leggero**

In realtà è un'applicazione molto leggera, anche con la certificazione JavaEE. Tuttavia, fornisce tutte le funzionalità necessarie e standard richieste per far funzionare un server, il che significa che offre un caricamento e una ridistribuzione molto rapidi rispetto alle sue varie alternative.

- **Altamente flessibile**

Grazie alle sue opzioni di personalizzazione integrate, alla natura ampia e leggera, offre un'elevata flessibilità, un utente può eseguirlo in qualsiasi modo desideri e funzionerà comunque senza problemi. Poiché è open source, chiunque abbia conoscenza può modificarlo in base alle proprie esigenze.

- **Stabilità**

È una delle piattaforme più stabili oggi disponibili su cui basarsi e utilizzarla per eseguire le nostre applicazioni. È incredibilmente stabile perché funziona indipendentemente dalla nostra installazione di Apache. Nel caso in cui si verificasse un grosso errore in Tomcat a causa del quale smettesse di funzionare, il resto del nostro server funzionerebbe perfettamente.

- **Fornisce un ulteriore livello di sicurezza**

Poiché le diverse organizzazioni di solito preferiscono posizionare l'installazione di Tomcat dietro la protezione di un firewall aggiuntivo che può essere accessibile solo dall'installazione di Apache.

- **È ben documentato**

Dispone di un'eccellente documentazione disponibile, inclusa una vasta gamma di tutorial online disponibili gratuitamente che possono essere scaricati o visualizzati direttamente online dall'utente, il che lo rende una delle scelte migliori per soddisfare i requisiti di un server applicativo principalmente ogni applicazione web java. Se un utente sta cercando le istruzioni di installazione, le impostazioni di avvio, le note di configurazione del server, tutti i tipi di informazioni su Tomcat sono già disponibili su Internet.

- **È uno dei server applicativi più utilizzati**

Secondo una stima, detiene quasi il 60 per cento della quota di mercato di quasi tutte le distribuzioni di server applicativi Java, il che lo rende uno dei server applicativi più popolari utilizzati per le applicazioni basate su Web Java. Tomcat funge da "webserver" o "servlet container".

- **È maturo**

Si scoprirà che esiste da quasi 20 anni, che è un tempo abbastanza significativo, in cui matura con il passare del tempo. Poiché Tomcat è un software open source, viene aggiornato e le nuove versioni escono quasi regolarmente e la comunità open source lo mantiene. La maturità lo rende uno dei server applicativi più estremamente stabili per lo sviluppo di software, applicazioni e la distribuzione di applicazioni Java. Da ora, è un'opzione estremamente stabile che diventa più potente con un eccellente supporto della comunità.

#### Svantaggi di Tomcat

- Ha un'interfaccia utente primaria, che non è di alta qualità.
- In termini di analisi statica, Tomcat non è veloce come Apache .
- Gli utenti di solito incontrano problemi nell'installazione di SSL con Tomcat.
- Ha alcuni problemi come una perdita di memoria