

Appunti di Paradigmi

4 ottobre 2022

Rosso Carlo

Contents

1	Introduzione	2
1.1	Java	2
2	Concorrenza	2

1 Introduzione

Def. 1.1 (Paradigma) *Dal greco paradeigma: modello, esempio. Modello di riferimento, termine di paragone. In informatica, un paradigma l'insieme delle tecniche e dei metodi considerati adeguati ad affrontare una classe determinata, anche se ampia, di problemi.*

Esempi:

- concorrenza: più linee di esecuzione contemporanee, asincrone, che condividono l'uso di un insieme di risorse in modo (possibilmente) coordinato;
- parallelismo: più linee di esecuzione contemporanee, sincrone, che eseguono in modo coordinato lo stesso calcolo su di una partizione dei dati di ingresso;
- in rete: collaborazione con altri sistemi attraverso la comunicazione asincrona su di una interfaccia di rete;
- distribuzione: un sistema è costituito da nodi indipendenti che, attraverso una rete non affidabile, devono coordinare l'esecuzione dello stesso lavoro, condividendo il consenso sullo stato complessivo del sistema;
- reattività: un sistema distribuito costruito sulle basi della comunicazione asincrona tramite messaggi da cui ottiene caratteristiche di flessibilità, resilienza, scalabilità e reattività;

Utilizziamo Java per implementare i paradigmi di programmazione. Java è un linguaggio OO, a memoria gestita (GC), con controllo statico dei tipi, basato su classi ad ereditarietà singola, con una sintassi simile a C e C++. Java è compilato in un linguaggio intermedio (bytecode), il cui modello di esecuzione è descritto nella specifica della JVM. Il bytecode può essere eseguito in molti modi: interpretato, compilato durante l'esecuzione (aka JIT), compilato prima dell'esecuzione (GraalVM).

1.1 Java

Ogni file deve chiamarsi come l'oggetto pubblico che contiene: NomeClasse.java.

Il percorso delle directory deve corrispondere al package in cui l'oggetto si trova.

Nei sorgenti, o nel CLASSPATH, devono esserci tutti i tipi nominati dai sorgenti.

L'ordine di compilazione non è importante.

Il comando "java" avvia la JVM ed esegue il bytecode contenuto nel CLASSPATH.

Il formato .jar è il più comune formato di distribuzione del codice nella piattaforma Java. Si tratta di un archivio compresso zip contenente alcuni file specifici che descrivono il suo contenuto e come usarlo.

L'unità principale di organizzazione del codice è la Classe. Ogni oggetto fa necessariamente riferimento alla definizione di un aClasse, che determina la struttura del suo stato ed il codice che opera su tale stato.

Il codice eseguibile deve necessariamente essere incluso in una Classe: come metodo dotato di un nome e richiamabile da altri oggetti, o come blocco anonimo eseguito alla creazione di ciascuna istanza. Le classi formano l'insieme dei Tipi che il compilatore riconosce in un programma Java. Per convenzione i package sono denominati con nomi di dominio, in ordine inverso.

Una classe non può usare un'altra classe qualsiasi: deve averne visibilità e dichiarare l'intenzione di usarla.

Una classe può usare una qualsiasi altra classe all'interno dello stesso package senza indicazioni particolari.

Un file sorgente può contenere al più una classe pubblica, e deve chiamarsi come la classe contenuta. "import", per importare una classe. Una classe può contenere variabili, metodi, altre classi, blocchi di codice anonimi.

Interfacce: una interfaccia dichiara le caratteristiche di un tipo senza fornire una sua implementazione. Le classi possono dichiarare di implementare una interfaccia fornendo l'implementazione richiesta. Una interfaccia può essere estesa solo da un'altra interfaccia. Una interfaccia può avere visibilità pubblica o di package.

2 Concorrenza

Def. 2.1 (programma concorrente) *Teoria e tecniche per la gestione di più processi sulla stessa macchina che operano contemporaneamente condividendo le risorse disponibili.*

Questo richiede sempre maggiore gestione della concorrenza e dell'accesso contemporaneo alle stesse risorse. Inoltre, alcune di queste tecniche si sono rivelate problematiche dal punto di vista della sicurezza. Per gestire più linee di esecuzione all'interno dello stesso processo è stato ideato il concetto di thread. I thread condividono le risorse di uno stesso processo, rendendo più economico il costo di passaggio da un ramo di esecuzione all'altro. La programmazione distribuita implica la comunicazione fra entità che non possono avere stato condiviso. La programmazione funzionale tratta preferibilmente dati immutabili, con qualche concessione alla mutabilità per lo stretto necessario. Lo stato può essere distinto o condiviso. La programmazione concorrente si pone nel quadrante più difficile, dove lo stato è mutabile e condiviso, e quindi l'accesso e l'intervento su di esso va coordinato e gestito.

Problemi della concorrenza:

- non determinismo;
- starvation;
- race condition;
- deadlock;

Tipologie di concorrenza:

- collaborativa: co-routine;
- pre-emptive: processi, threads;
- real-time: processi, threads;
- event driven (o async): future, events, streams;

collaborativa I programmi devono esplicitamente cedere il controllo ad intervalli regolari. Casi d'uso: embedded, very high performance.

pre-emptive Il sistema operativo è in grado di interrompere l'esecuzione di un programma e sottrargli il controllo delle risorse per affidarle al programma seguente.

real-time Il sistema operativo garantisce prestazioni precise e prefissate nella suddivisione delle risorse fra i programmi.

event driven/async I programmi dichiarano esplicitamente le operazioni che vanno eseguite e lasciano all'ambiente di esecuzione la decisione di quando eseguirle e come assegnare le risorse. Si tratta di un metodo poco comune a livello di sistema operativo, ma molto popolare nell'organizzazione delle applicazioni.