

La programmazione procedurale

In informatica la programmazione procedurale è un paradigma di programmazione che consiste nel creare dei blocchi di codice sorgente, identificati da un nome e racchiusi da dei delimitatori, che variano a seconda del linguaggio di programmazione; questi sono detti anche sottoprogrammi (in inglese subroutine) procedure o funzioni, a seconda del linguaggio e dei loro ruoli all'interno del linguaggio stesso.

Il nome deriva dal linguaggio COBOL, che è stato il primo ad utilizzare questo concetto.

I linguaggi imperativi (o procedurali) sono certamente i più diffusi, considerando anche quelli **Object Oriented**. Per esempio: C++, Java, Python, Objective C, e ovviamente i linguaggi come C, Fortran, Cobol, Pascal.

La programmazione è incentrata sui dati (variabili, strutture dati, oggetti) e sui modi per processarli (si parla di procedure, funzioni, metodi a seconda dei casi).

Si parte da un insieme di dati cui si applicano una o più **manipolazioni** da eseguire in un determinato ordine al fine di ottenere il risultato finale.

La programmazione procedurale può essere il primo paradigma di programmazione che un nuovo sviluppatore imparerà. Fondamentalmente, il codice procedurale è quello che istruisce direttamente un dispositivo su come completare un'attività in passaggi logici.

Questo paradigma utilizza un approccio lineare dall'alto verso il basso e tratta i dati e le procedure come due entità diverse. Sulla base del concetto di una chiamata di procedura, la Programmazione procedurale suddivide il programma in procedure, note anche come routine o funzioni, contenenti semplicemente una serie di passaggi da eseguire.

In poche parole, la Programmazione procedurale comporta la redazione di un elenco di istruzioni per dire al computer cosa dovrebbe fare passo dopo passo per completare l'attività a portata di mano.

Caratteristiche principali della programmazione procedurale

Le caratteristiche principali della programmazione procedurale sono riportate di seguito:

- **Funzioni predefinite:** una funzione predefinita è in genere un'istruzione identificata da un nome. Di solito, le funzioni predefinite sono integrate in linguaggi di programmazione di livello superiore, ma derivano dalla libreria o dal registro, piuttosto che dal programma. Un esempio di una funzione predefinita è `charAt()`, che cerca la posizione di un carattere in una stringa.
- **Variabile locale:** una variabile locale è una variabile dichiarata nella struttura principale di un metodo ed è limitata all'ambito locale che viene fornito. La variabile locale può essere utilizzata solo nel metodo in cui è definita e, se dovesse essere utilizzata al di fuori del metodo definito, il codice smetterà di funzionare.
- **Variabile globale:** una variabile globale è una variabile dichiarata al di fuori di ogni altra funzione definita nel codice. Per questo motivo, le variabili globali possono essere utilizzate in tutte le funzioni, diversamente da una variabile locale.
- **Modularità:** la modularità è quando due sistemi diversi hanno due compiti diversi a portata di mano ma sono raggruppati insieme per concludere prima un compito più ampio. Ogni gruppo di sistemi dovrebbe

quindi completare i propri compiti uno dopo l'altro fino al completamento di tutti i compiti.

- Passaggio parametri: il passaggio parametri è un meccanismo utilizzato per passare parametri a funzioni, subroutine o procedure. Il passaggio dei parametri può essere eseguito tramite "passa per valore", "passa per riferimento", "passa per risultato", "passa per risultato valore" e "passa per nome".

Vantaggi e svantaggi della programmazione procedurale

La programmazione procedurale include una serie di pro e contro, alcuni dei quali sono indicati di seguito.

vantaggi

- La programmazione procedurale è eccellente per la programmazione generale
- La semplicità insieme alla facilità di implementazione di compilatori e interpreti
- Una grande varietà di libri e materiale didattico online disponibile su algoritmi testati, che semplifica l'apprendimento lungo il percorso
- Il codice sorgente è portabile, pertanto può essere utilizzato anche come target per una CPU diversa
- Il codice può essere riutilizzato in diverse parti del programma, senza la necessità di copiarlo
- Attraverso la tecnica di programmazione procedurale, anche il requisito di memoria viene ridotto
- Il flusso del programma può essere monitorato facilmente

svantaggi

- Il codice del programma è più difficile da scrivere quando si utilizza la Programmazione procedurale
- Il codice procedurale spesso non è riutilizzabile, il che può comportare la necessità di ricreare il codice se necessario per l'uso in un'altra applicazione
- Difficile relazionarsi con oggetti del mondo reale
- L'importanza è data all'operazione piuttosto che ai dati, il che potrebbe comportare problemi in alcuni casi sensibili ai dati
- I dati sono esposti a tutto il programma, il che rende meno sicuro il programma

Esistono diversi tipi di paradigma di programmazione, che non sono altro che uno stile di programmazione. Il paradigma non si adatta a una lingua specifica, ma al modo in cui è scritto il programma.

[Paradigmi e Linguaggi](#)